

Características de Qualidade para Componentes de Software

Régis Patrick Silva Simão
regis.simao@serpro.gov.br

Arnaldo Dias Belchior (Orientador)
belchior@unifor.br

Mestrado em Informática Aplicada, Universidade de Fortaleza (UNIFOR)
Av. Washington Soares, 1321, CEP: 60.811-341 – Fortaleza-CE, Brasil

Resumo

A qualidade de componentes de software é importante para a qualidade de aplicações baseadas em componentes e para a obtenção dos principais objetivos do Desenvolvimento Baseado em Componentes (DBC), baixo custo e alta produtividade e qualidade. As características de qualidade, segundo a norma ISO/IEC 9126, podem ser utilizadas como metas a serem atingidas no desenvolvimento, na seleção e na aquisição de componentes e, também, como fatores na predição de propriedades de aplicações finais. Este trabalho identifica e organiza as características de qualidade de maior relevância para componentes de software e define dimensões de qualidade, que influenciam no grau de importância das mesmas. O levantamento dessas características foi baseado em literatura especializada. Realizou-se, ainda, uma pesquisa de campo, com especialistas em DBC, visando a hierarquização dessas características de qualidade, cujos resultados foram obtidos através de um modelo de avaliação da qualidade de software.

Palavras-chave: Qualidade de Software, Componentes de Software, Avaliação de Componentes, Características de Qualidade.

Abstract

The quality of software components is important for the quality of component-based applications and for attaining the main objectives of Component-based Development (CBD), low cost and high productivity and quality. The quality characteristics, according to ISO/IEC 9126, can be used as goals to be reached in the development, in the selection and in the acquisition of components and, also, as factors in the prediction of properties of final applications. This work identifies and organizes the main quality characteristics for software components and, also, defines quality dimensions that influence in the degree of importance of those characteristics. The survey of the characteristics was based on the specialized literature. Research was undertaken, with CBD practitioners, to rank the quality characteristics and the results were obtained from a software quality evaluation model.

Keywords: Software Quality, Software Components, Component Evaluation, Quality Characteristics.

1. Introdução

O Desenvolvimento Baseado em Componentes (DBC) aparece como uma técnica promissora para a resolução dos problemas, tais como: prazos e orçamentos excedidos, e criação de aplicações com baixo nível de qualidade e altos índices de manutenção. Em geral,

essa técnica consiste na divisão de blocos monolíticos em módulos interoperáveis, que são os componentes. Os desenvolvedores passam a construir módulos independentes e combiná-los nas aplicações finais. A reutilização de módulos pré-fabricados pode substituir ou, pelo menos, diminuir a atividade de construção desses módulos, reduzindo o tempo e o custo do desenvolvimento, e aumentando a produtividade e a qualidade [10].

O termo componente tem sido utilizado vastamente com inúmeras definições; cada uma delas apresenta as propriedades mais relevantes para o contexto no qual está inserido. Neste trabalho, o conceito de componentes utiliza-se o conceito de Villela [10], onde “componentes reutilizáveis são artefatos autocontidos, claramente identificáveis, que descrevem ou realizam uma função específica e têm interfaces claras em conformidade com um dado modelo de arquitetura de software, documentação apropriada e um grau de reutilização definido”. A escolha dessa definição não possui a pretensão de fechar as discussões sobre as propriedades para componentes, mas de ser um ponto de partida para uma pesquisa mais abrangente.

1.1. Objetivos do Trabalho

Este trabalho identificou, organizou, e hierarquizou um conjunto de características e subcaracterísticas de qualidade para componentes de software, a partir da norma ISO/IEC 9126, definindo dimensões de qualidade, que influenciam no grau de importância das mesmas [7, 8, 9]. Apresenta-se também uma pesquisa de campo realizada para validar o conjunto de características e subcaracterísticas [7, 8, 9]. Com isto, objetiva-se poder auxiliar na melhoria da qualidade de sistemas de software baseados em componentes, aumentando a qualidade de suas partes constituintes, os componentes.

Este trabalho está organizado da seguinte forma: na seção 2, apresentam-se os fatores que influenciam na qualidade de um software baseado em componentes, e um conjunto de atributos de qualidade para componentes de software; na seção 3, uma pesquisa de campo realizada com especialistas em DBC e seus resultados são comentados; finalmente, as principais conclusões da dissertação encontram-se na seção 4.

2. Qualidade de Software Baseado em Componentes

Da mesma forma que no desenvolvimento tradicional de software, vários fatores influenciam na qualidade de uma aplicação baseada em componentes [11]: qualidade do processo de desenvolvimento; fatores humanos, sociais e organizacionais; arquitetura do software, estilos arquiteturais, modelo e *framework* de componentes; e qualidade dos componentes, documentação e artefatos em geral.

Apesar de avanços no uso de componentes de software e arquiteturas de software, ainda existem muitos riscos funcionais e de qualidade envolvidos na utilização desses componentes [4]. Na tentativa de alcançar os principais benefícios prometidos pelo DBC, observou-se que há uma grande necessidade por componentes de software de qualidade [4, 6].

Alguns trabalhos apresentam um conjunto de atributos de qualidade para componentes de software [2, 6]. Entretanto, o fazem de forma simplificada e não seguem um modelo padrão [6] ou, de uma maneira muito específica, focam somente propriedades de componentes COTS [2]. A seguir, apresenta-se um conjunto bastante abrangente de atributos

de qualidade para componentes de software em geral, organizado segundo a norma ISO/IEC 9126 [3].

2.1. Características de Qualidade de Componentes de Software

A identificação das características de qualidade para componentes de software consistiu de uma significativa pesquisa bibliográfica na área de qualidade de produtos de software e, principalmente, na área de DBC. As características de qualidade foram organizadas segundo a norma ISO/IEC 9126, parte 1 [3], por ser um padrão estabelecido internacionalmente, embora, tratando somente o modelo de qualidade para características externas e internas.

É importante ressaltar que, na pesquisa bibliográfica realizada, boa parte das características de qualidade não estavam explicitamente definidas. Através do entendimento de contexto, os atributos de qualidade para componentes de software foram identificados e transformados em subcaracterísticas, em suas respectivas características de qualidade, segundo o modelo utilizado. A organização dessas subcaracterísticas de qualidade passou por uma análise criteriosa, através de um processo de depuração árduo, baseando-se, principalmente, na pesquisa bibliográfica efetuada e em experiências de desenvolvimento de componentes.

Foram levantados 124 atributos de qualidade para componentes de software, sendo organizados segundo a norma ISO/IEC 9126 [3]. Este conjunto de atributos consiste das 6 características da norma ISO e 118 subcaracterísticas de qualidade, considerando-se, também, todas as 27 subcaracterísticas de qualidade da ISO/IEC 9126 [3]. Assim sendo, foram selecionadas 91 novas subcaracterísticas de qualidade para componentes de software. Devido à especificidade dos componentes de software, houve a necessidade de se definir novas subcaracterísticas, além das já definidas para produtos de software em geral. Entretanto, não houve a necessidade de se acrescentar novas características de qualidade ao padrão ISO, observando-se a aplicabilidade de todas características e subcaracterísticas já definidas nessa norma.

No *Anexo A*, podem ser vistas as características e subcaracterísticas de qualidade para componentes de software. Algumas observações se fazem necessárias para a utilização do modelo proposto:

- As localizações das subcaracterísticas na hierarquia não pretendem ser absolutas, isto é, uma determinada subcaracterística poderia ser especificada em vários outros ramos da árvore hierárquica de atributos de qualidade proposta. Entretanto, por uma questão de organização, uma determinada subcaracterística é definida somente uma única vez e localizada em um ramo da árvore, onde se observou que a sua maior influência é exercida.
- Devido à especificidade dos componentes de software, alguns conceitos utilizados tomaram um caráter mais abrangente do que os propostos pela norma ISO/IEC 9126 [3] e por algumas bibliografias especializadas. Por exemplo, a usabilidade, deve abranger não somente a interação com o usuário final, mas também a manipulação do componente, pelos desenvolvedores, durante as diversas atividades do ciclo de vida de desenvolvimento de uma aplicação baseada em componentes [2]. A característica portabilidade, que visa o transporte do produto de software entre ambientes. Sua interpretação foi estendida para capturar os aspectos técnicos necessários à reutilização de componente em diversas aplicações.

- O conjunto de características de qualidade de componentes de software é relativamente extenso, podendo ser usado total ou parcialmente para a definição de guias de qualidade e para a avaliação da qualidade de componentes de software. Novas subcaracterísticas também podem ser acrescentadas, como, por exemplo, para mapear atributos de qualidade de contextos particulares, como objetos distribuídos ou móveis, que possam vir a ser avaliados.
- Os usuários, que estejam interessados nas características de qualidade para componentes, sejam eles arquitetos de software, projetistas, desenvolvedores de componentes e de aplicação baseada em componentes ou pesquisadores de DBC, podem utilizar as guias de qualidade deste modelo, para a definição de um novo modelo de qualidade ou para a avaliação e a certificação de componentes.

Durante o levantamento bibliográfico, percebeu-se que vários fatores podem influenciar no grau de importância atribuído às subcaracterísticas de qualidade para componentes de software, e devem ser considerados na elaboração de guias e na realização de avaliações desses componentes.

Assim sendo, adotaram-se três dimensões distintas para a obtenção do grau de importância de uma característica ou subcaracterística de qualidade para componentes de software:

- *o domínio da aplicação*: corresponde ao conjunto de características peculiares do domínio de aplicação, onde o componente é utilizado, definindo um conjunto de requisitos da qualidade e seus processos de avaliação.
- *a função arquitetural*: distribuída pelas camadas de interface, negócio, dados, e infra-estrutura, envolvendo seus padrões de interação.
- *o nível de abstração*: o componente pode ser reutilizado em suas formas de especificação, projeto e código (macro-etapas do ciclo de vida do software).

Vários trabalhos realizados para o levantamento da qualidade para produtos de software, em diversos domínios de aplicação, mostraram a influência da dimensão *Domínio da Aplicação* no grau de importância das propriedades de qualidade. Outros trabalhos evidenciaram que a importância dada aos atributos de qualidade também pode variar conforme a forma ou o *nível de abstração* em que o produto está sendo avaliado. Este trabalho aborda a dimensão *função arquitetural*, através dos resultados obtidos em uma pesquisa de campo realizada, demonstrando a influência da mesma nas características de qualidade e na validação do conjunto de atributos de qualidade para componentes de software.

3. Avaliação das Características de Qualidade para Componentes de Software

A avaliação das características de qualidade para componentes de *interface*, de *negócio*, de *dados*, de *infra-estrutura* e para componentes em geral (*consolidado*) é apresentada como resultado de uma pesquisa de campo realizada com especialistas, que atuam com DBC. Essa pesquisa objetivou capturar o grau de importância das características de qualidade de componentes de software de acordo com a dimensão *função arquitetural*, (subseção 2.1), resultando em cinco guias de qualidade para componentes de: (i) interface, (ii) negócio, (iii) dados, (iv) infra-estrutura, e (v) consolidado (para componentes de software em geral).

A metodologia adotada para a pesquisa de campo e avaliação das características de qualidade de componentes de software utilizou o Modelo *Fuzzy* para Avaliação de Qualidade de Software (MFAQS) [1]. Já foram avaliados vários produtos de software com esse modelo,

e os resultados obtidos foram considerados expressivos e satisfatórios. Na representação de modelos de raciocínio imprecisos, tem-se utilizado a teoria dos conjuntos *fuzzy*, que vem sendo aplicada em várias áreas do conhecimento humano, sendo o elo de ligação entre modelos imprecisos (subjetivos) do mundo real e sua representação matemática [12].

Um conjunto *fuzzy* é caracterizado por uma função de pertinência, que mapeia os elementos de um domínio ou universo de discurso X para um número real no intervalo $[0, 1]$. Formalmente, $\tilde{A} : X \rightarrow [0, 1]$. Desta forma, um conjunto *fuzzy* \tilde{A} , por exemplo, apresenta-se como um conjunto de pares ordenados, em que o primeiro elemento é $x \in X$, o segundo, $\mu_{\tilde{A}}(x)$, é o grau de pertinência ou a função de pertinência de x em \tilde{A} , que mapeia x no intervalo $[0, 1]$, ou seja, $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) \mid x \in X\}$. A pertinência de um elemento a um determinado conjunto passa a ser uma questão de gradação. Em casos extremos, o grau de pertinência é 0, indicando que o elemento não pertence ao conjunto, ou o grau de pertinência é 1, se o elemento pertence 100% ao conjunto [12].

O MFAQS é constituído das cinco etapas seguintes, que foram utilizadas para avaliação das características de qualidade para componentes de software [1]:

i. Identificação do objeto a ser avaliado e do conjunto de atributos de qualidade de software a ser considerado (características e subcaracterísticas).

O objeto de avaliação foi componente de software, isto é, avaliar a importância das características de qualidade para componentes de software, segundo a dimensão *função arquitetural*, e o conjunto de itens a ser avaliado foi o conjunto de 6 características e 118 subcaracterísticas de qualidade para componentes de software. É importante ressaltar que o MFAQS foi adaptado para permitir a utilização de modelos de qualidade organizados, segundo a norma ISO/IEC 9126 [5].

ii. Seleção de especialistas e identificação de seu perfil.

Os especialistas envolvidos foram: desenvolvedores de componentes e de aplicações com componentes, arquitetos de software, gerentes de projeto (que utilizam DBC), e professores universitários de engenharia de software (com conhecimento de DBC).

O perfil de cada especialista, que participou da pesquisa de campo, foi obtido através do Questionário de Identificação do Perfil do Especialista (QIPE), com o objetivo de capturar sua experiência, no desenvolvimento de componentes e de aplicações com componentes, gerando-se um peso por especialista, que influenciou no resultado final da avaliação, segundo o que determina o modelo *fuzzy*.

Foram enviados questionários a um grupo de 30 (trinta) especialistas no Brasil. Destes questionários, somente um não foi considerado por que o especialista não estava envolvido em atividades de DBC, mostrado em seu QIPE. Os perfis dos outros especialistas podem ser resumidos como se segue:

- *Experiência em DBC*: 56,25% dos especialistas eram desenvolvedores de componentes e tinham participado de construção de aplicações baseadas em componentes; 50 % já tinham trabalhado como arquitetos de software, e 31,25% já foram gerentes de projetos de DBC (estes percentuais não são mutuamente exclusivos).
- *Complexidade de DBC*: 37,50% já participaram de sistemas de DBC de alta complexidade; 87,50% e 68,75% têm desenvolvido aplicações de média e baixa complexidade, respectivamente.

- *Nível de graduação*: todos os especialistas são, no mínimo, graduados em ciências exatas; 31,25% possuem doutorado em ciência da computação; 37,50% são mestres; 12,50% têm especialização em nível de pós-graduação, e 18,75% restantes são somente graduados.
- *Publicações e palestras sobre DBC*: 31,25% têm publicações sobre DBC em anais de conferências internacionais; 18,75% têm ministrado palestras em congressos e simpósios.

iii. Obtenção do grau de importância das subcaracterísticas mensuráveis de qualidade, identificadas na primeira etapa, dos especialistas selecionados na segunda etapa.

As subcaracterísticas mensuráveis são aquelas no nível mais inferior dos diversos ramos da árvore hierárquica (nós folhas), e foram avaliadas através de um questionário específico para os especialistas. Nesta avaliação, os especialistas forneceram notas de 0 a 4 (0 – sem importância; 1 – pouca importância; 2 – desejável; 3 – muito importante; e 4 – imprescindível) para cada opção da dimensão *função arquitetural* de cada subcaracterística mensurável, que passou pelo processo de “fuzificação”, isto é, a transformação dessas notas (valores nítidos) em números *fuzzy* normais triangulares.

Um número *fuzzy* normal triangular pode ser representado por $\tilde{N}(a, m, b)$, onde:

- a característica ou subcaracterística será considerada de qualidade se for valorada no intervalo $[a, b]$;
- se a característica ou subcaracterística possuir uma valoração à esquerda de a não será considerada de qualidade; se possuir uma valoração à direita de b , o atributo excede à qualidade esperada, isto é, pode ter sido despendido um esforço maior do que o necessário, para se obter a qualidade da referida característica ou subcaracterística de qualidade;
- o valor de m corresponde ao valor desejável de qualidade do produto para aquela característica ou subcaracterística, por ter o grau de pertinência igual a 1. Nesta situação, é onde a relação custo/benefício para se obter a qualidade é a mais vantajosa.

iv. Tratamento *fuzzy* dos dados coletados dos especialistas, obtidos na terceira etapa.

Nesta etapa, foi realizado o tratamento *fuzzy* dos dados coletados dos especialistas na avaliação das subcaracterísticas de qualidade mensuráveis, levando-se em consideração o peso de cada especialista, obtido na segunda etapa. É importante ressaltar que o tratamento *fuzzy* foi realizado separadamente para cada um dos quatro itens considerados da *função arquitetural*, como também de forma consolidada, isto é, o resultado de todos esses itens agregados.

v. Agregação *fuzzy* dos atributos de qualidade de software, em cada nível hierárquico do modelo de qualidade, a partir do tratamento realizado na quarta etapa.

Nesta etapa, os resultados obtidos pelas subcaracterísticas mensuráveis foram agregados, para calcular os resultados das subcaracterísticas de níveis superiores e, conseqüentemente, das características. Esses resultados obtidos geraram os guias de qualidade. Novamente, foram realizadas agregações para cada um dos itens da *função arquitetural* em separado e também para a obtenção de um resultado consolidado.

Para o tratamento *fuzzy* dos dados nas etapas 4 e 5, foi utilizada a ferramenta AdeQuaS, que automatizou o MFAQS [5].

Como resultado desta pesquisa de campo, foram obtidos cinco guias de qualidade: um

para cada opção da dimensão *função arquitetural* e um guia a partir da consolidação para componentes de software em geral. Esse último guia foi apresentado em Simão e Belchior [7].

A seguir, são apresentados os resultados alcançados para componentes de interface, negócio, dados, infra-estrutura e consolidado.

As interpretações dos resultados obtidos pela pesquisa de campo realizada são feitas a partir dos termos lingüísticos dos números *fuzzy* triangulares obtidos, e consideraram apenas o valor *m* do número *fuzzy* triangular do tipo $\tilde{N}(a, m, b)$, por ser o valor desejável de qualidade.

Por uma questão de limitação de espaço, somente serão apresentados os resultados obtidos para as características de qualidade da ISO/IEC 9126 (guias de qualidade), para os quatro tipos de componentes (interface, negócio, dados, infra-estrutura), além do resultado geral (consolidado), sendo comentados apenas os resultados das três mais importantes de cada tipo.

3.1 Avaliação da Qualidade para Componentes de Interface

Como componentes de interface, entende-se que são componentes utilizados na interação entre os usuários e o sistema e nas interações entre sistemas. A *Tabela 1* apresenta os resultados obtidos para as características de qualidade para componentes de interface.

A característica *Portabilidade* foi a melhor avaliada para componentes de interface. Compreende subcaracterísticas como *Capacidade para ser instalado*, *Substituibilidade* e *Adaptabilidade*, que são muito úteis em técnicas de montagem rápida de componentes maiores, como montagem e validação de protótipos de telas no ambiente do usuário. Neste caso, os componentes podem ser substituídos ou rejeitados, mudando-se as configurações dos mesmos com facilidade. Boa parte do sucesso de componentes de interface em ambientes RAD (*Rapid Application Development*), por exemplo, ambientes com linguagens Delphi, C++ e Java, particularmente pacotes AWT (*Abstract Window Toolkit*) e *Swing*, pode ser creditada a essas subcaracterísticas. Essa característica obteve o grau de importância de 3,06, sendo 5,95% de *imprescindível* e 94,05% de *muito importante*.

Tabela 1: Características ISO em ordem decrescente de importância para Componentes de Interface

| Característica | Número Fuzzy | Interpretação |
|------------------|--------------------|--|
| Portabilidade | (2,06; 3,06; 3,65) | 5,95% Imprescindível e 94,05% Muito importante |
| Funcionalidade | (1,99; 2,99; 3,63) | 98,70% Muito importante e 1,30% Desejável |
| Usabilidade | (1,96; 2,96; 3,62) | 95,71% Muito importante e 4,29% Desejável |
| Manutenibilidade | (1,79; 2,78; 3,58) | 78,40% Muito importante e 21,60% Desejável |
| Confiabilidade | (1,78; 2,76; 3,54) | 75,99% Muito importante e 24,01% Desejável |
| Eficiência | (1,36; 2,31; 3,22) | 31,13% Muito importante e 68,87% Desejável |

Componentes de interface são responsáveis pela interação entre os usuários e a aplicação e entre aplicações, sendo a parte mais externamente visível de uma aplicação. É através deles que os usuários inicialmente validam os sistemas. Logo, é fundamental que os componentes escolhidos estejam em conformidade com os requisitos, sejam corretos, bem expressivos (interativos) e seguros. Assim, a segunda característica melhor avaliada foi a *Funcionalidade*, que obteve grau de importância 2,99, significando 98,70% de *muito importante* e 1,30% de *desejável*.

A característica *Usabilidade* foi a terceira melhor avaliada. A maioria das

organizações possui padrões visuais e de diálogo que devem ser seguidos. Assim sendo, a conformidade dos componentes de interface com os padrões deve ser seguida. É importante também que esses componentes sejam fáceis de aprender e de manusear, para que dêem agilidade como, por exemplo, à atividade de prototipagem rápida, e ao uso do sistema. A Usabilidade obteve o grau de 2,96, sendo 95,71% de *muito importante* e 4,29% de *desejável*.

3.2 Avaliação da Qualidade para Componentes de Negócio

Componentes de negócio são responsáveis pelo controle e execução das regras de negócio. Na *Tabela 2*, os resultados obtidos para as características de qualidade para componentes de negócio são apresentados.

Os componentes de negócio são responsáveis pela lógica da aplicação, isto é, regras de negócio do sistema. Questões como adequação dos serviços, corretitude, acurácia e segurança de acesso são importantes para esses componentes. Assim, a característica *Funcionalidade*, que é responsável por essas propriedades de qualidade, obteve o grau mais elevado de importância com 3,32, sendo 32,02% de *imprescindível* e 67,98% de *muito importante*.

A grande complexidade das aplicações está nos componentes de negócio, que são responsáveis pelo controle e pela execução da lógica do negócio. Esses componentes estão sujeitos a apresentar boa parte dos erros de uma aplicação. Assim, propriedades como *Maturidade*, *Tolerância a falhas* e *Recuperabilidade* são importantes para os componentes. Logo, a característica *Confiabilidade* responsável por estes atributos de qualidade recebeu o segundo maior grau de importância, 3,26, significando 26,10% de *imprescindível* e 73,90% de *muito importante*.

Tabela 2: Características ISO em ordem decrescente de importância para Componentes de Negócio

| Característica | Número <i>Fuzzy</i> | Interpretação |
|------------------|---------------------|---|
| Funcionalidade | (2,32; 3,32; 3,75) | 32,02% Imprescindível e 67,98% Muito importante |
| Confiabilidade | (2,27; 3,26; 3,76) | 26,10% Imprescindível e 73,90% Muito importante |
| Portabilidade | (2,23; 3,23; 3,80) | 22,89% Imprescindível e 77,11% Muito importante |
| Manutenibilidade | (2,13; 3,13; 3,74) | 12,76% Imprescindível e 87,24% Muito importante |
| Eficiência | (1,97; 2,97; 3,69) | 96,60% Muito importante e 03,40% Desejável |
| Usabilidade | (1,82; 2,76; 3,45) | 75,68% Muito importante e 24,32% Desejável |

Componentes de negócio são difíceis e caros de serem construídos. Poder reutilizá-los é muito vantajoso para as empresas. A característica *Portabilidade* possui atributos técnicos, que favorecem a reutilização como, por exemplo: *Substituibilidade*, *Capacidade para ser instalado*, e *Adaptabilidade*. Diante disso, os especialistas mostraram suas preocupações com o reaproveitamento desses componentes, fazendo da *Portabilidade* a terceira característica melhor avaliada. Ela obteve o grau 3,23, sendo interpretado por 22,89% de *imprescindível* e 77,11% de *muito importante*.

3.3 Avaliação da Qualidade para Componentes de Dados

Os resultados obtidos para as características de qualidade para componentes de dados são apresentados na *Tabela 3*. Esses componentes modelam a informação do domínio de

aplicação, possuem a característica de serem persistentes ou de fornecer mecanismos de suporte a persistência.

Tabela 3: Características ISO em ordem decrescente de importância para Componentes de Dados

| Característica | Número <i>Fuzzy</i> | Interpretação |
|-----------------------|----------------------------|---|
| Eficiência | (2,22; 3,21; 3,84) | 20,85% Imprescindível e 79,15% Muito importante |
| Confiabilidade | (2,12; 3,11; 3,72) | 10,99% Imprescindível e 89,01% Muito importante |
| Funcionalidade | (2,07; 3,06; 3,63) | 6,28% Imprescindível e 93,72% Muito importante |
| Portabilidade | (2,01; 3,01; 3,76) | 1,19% Imprescindível e 98,81% Muito importante |
| Manutenibilidade | (1,81; 2,81; 3,58) | 80,66% Muito importante e 19,34% Desejável |
| Usabilidade | (1,49; 2,41; 3,32) | 41,41% Muito importante e 58,59% Desejável |

A característica de qualidade *Eficiência* abrange aspectos relacionados à manutenção da informação com o suporte à transação e à persistência, como também questões, que envolvem a manipulação de grandes quantidades de informação, como tempo de resposta e acessos a periféricos. Foi considerada a mais importante característica para componentes de dados, com o grau 3,21, sendo 20,85% de *imprescindível* e 79,15% de *muito importante*.

A segunda característica melhor avaliada foi a *Confiabilidade*, que abrange aspectos que, juntamente com o suporte à transação, procuram garantir a integridade das informações, como, por exemplo, a tolerância a falhas e a recuperação em caso de problemas. A *Confiabilidade* recebeu o grau 3,11, interpretando-se como 10,99% de *imprescindível* e 89,01% de *muito importante*.

A *Funcionalidade* como a terceira mais importante característica de qualidade mostra a preocupação dos especialistas com a escolha de componentes adequados para os dados a serem manipulados, e a arquitetura que rege toda forma de acesso. A segurança também é um aspecto importante no acesso a dados e que também faz parte de *Funcionalidade*. Essa característica obteve o grau 3,06, com 06,28% de *imprescindível* e 93,72% de *muito importante*.

3.4 Avaliação da Qualidade para Componentes de Infra-estrutura

Os resultados obtidos para as características de qualidade para componentes de infra-estrutura, que são componentes de suporte aos outros tipos de componentes, são apresentados na *Tabela 4*.

Tabela 4: Características ISO em ordem decrescente de importância para Componentes de Infra-estrutura

| Característica | Número <i>Fuzzy</i> | Interpretação |
|-----------------------|----------------------------|---|
| Portabilidade | (2,29; 3,29; 3,83) | 29,25% Imprescindível e 70,75% Muito importante |
| Eficiência | (2,13; 3,13; 3,80) | 13,10% Imprescindível e 86,90% Muito importante |
| Funcionalidade | (1,97; 2,97; 3,61) | 96,71% Muito importante e 03,29% Desejável |
| Manutenibilidade | (1,91; 2,91; 3,67) | 91,32% Muito importante e 08,68% Desejável |
| Confiabilidade | (1,82; 2,80; 3,49) | 79,61% Muito importante e 20,39% Desejável |
| Usabilidade | (1,72; 2,69; 3,46) | 68,95% Muito importante e 31,05% Desejável |

Os componentes de infra-estrutura fornecem serviços independentes do negócio da aplicação, sendo suporte aos outros tipos de componentes. Seus serviços podem ser utilizados por vários tipos de componentes e em diferentes aplicações. Os especialistas apontaram a *Portabilidade* como a característica mais importante para esse tipo de componente, com o grau de importância 3,29, com 29,25% de *imprescindível* e 70,75% de *muito importante*.

Os componentes de intra-estrutura não estão diretamente relacionados com a lógica do negócio, podendo interagir diretamente com periféricos. Como requisito desejável, devem passar despercebidos pelos usuários que utilizam a aplicação final. Assim, a característica *Eficiência* foi apontada como a segunda mais importante, com 3,13, sendo 13,10% de *imprescindível* e 86,90% de *muito importante*.

A característica *Funcionalidade* foi a terceira melhor avaliada. Ela diz respeito a aspectos de interatividade com outros componentes, de adequação arquitetural e tecnológica e de realização dos serviços de forma completa e independente. Essa característica recebeu grau 2,97, sendo interpretado como 96,71% de *muito importante* e 3,29% de *desejável*.

3.5 Avaliação da Qualidade para Componentes de Software

Com a consolidação dos dados obtidos para os quatro tipos de componentes acima apresentados, obteve-se um guia de qualidade para componentes de software em geral, apresentados na *Tabela 5*.

A idéia de utilização associada a componentes de software está fortemente relacionada a propriedades como adaptação sem necessidade de modificação interna, compatibilidade, que permita a substituição por novas versões ou outros componentes, e facilidade de instalação em ambientes diferentes daqueles onde, inicialmente, foram construídos. Os especialistas mostraram que os aspectos compreendidos pela característica *Portabilidade* devem ter uma atenção maior no trato de componentes de software. Essa característica obteve o grau de importância 3,16, sendo interpretada como 15,99% de *imprescindível* e 84,01% de *muito importante*.

O DBC emprega componentes de software como uma solução para aumentar a produtividade e a qualidade das aplicações finais, através da reutilização de componentes pré-desenvolvidos e testados. Os especialistas consideraram os aspectos relativos à maturidade, tolerância a falhas e recuperabilidade também como muito importantes. Os componentes devem possuir mecanismos, que permitam tratar situações de falha ou, pelo menos, sinalizar outros componentes que tenham esta função. Assim, a característica *Confiabilidade* obteve, no consolidado, o segundo maior grau de importância para componentes, com valor 3,08, sendo interpretado como 7,80% de *imprescindível* e 92,20% de *muito importante*.

Tabela 5: Características ISO em ordem decrescente de importância para Componentes de Software

| Característica | Número Fuzzy | Interpretação |
|-----------------------|---------------------|---|
| Portabilidade | (2,16; 3,16; 3,77) | 15,99% Imprescindível e 84,01% Muito importante |
| Confiabilidade | (2,08; 3,08; 3,68) | 7,80% Imprescindível e 92,20% Muito importante |
| Funcionalidade | (2,04; 3,04; 3,63) | 4,17% Imprescindível e 95,83% Muito importante |
| Eficiência | (2,02; 3,01; 3,72) | 1,04% Imprescindível e 98,96% Muito importante |
| Manutenibilidade | (1,91; 2,91; 3,64) | 90,54% Muito importante e 09,46% Desejável |
| Usabilidade | (1,67; 2,64; 3,43) | 64,11% Muito importante e 35,89% Desejável |

A utilização de componentes de software, principalmente desenvolvidos por terceiros, depende inicialmente dos serviços oferecidos por esses componentes atenderem às necessidades dos usuários, totalmente ou parcialmente, com o nível de precisão exigido e desses serviços estarem em conformidade com a arquitetura e a tecnologia utilizadas pela aplicação. Assim, a característica *Funcionalidade* obteve o grau 3,04, com 4,17% de *imprescindível* e 95,83% de *muito importante*.

4. Conclusão

Este trabalho propôs um conjunto de características e subcaracterísticas de qualidade para componentes de software, baseado na norma ISO/IEC 9126, contendo 6 características e 118 subcaracterísticas de qualidade. Foram apresentadas três dimensões de qualidade, que podem influenciar na importância dessas características e subcaracterísticas de qualidade, considerando-se o contexto de componentes de software: o *domínio de aplicação*, a *função arquitetural*, e o *nível de abstração*.

Tendo-se como foco apenas a dimensão *função arquitetural* (que envolve componentes de interface, de negócio, de dados, e de infra-estrutura), foi realizada uma pesquisa de campo, com especialistas em DBC, para validá-las. Os resultados dessa pesquisa geraram cinco guias de qualidade: um para cada item da dimensão *função arquitetural*, e um guia consolidando toda essa dimensão. Os resultados foram obtidos através da utilização do Modelo *Fuzzy* para Avaliação da Qualidade de Software.

Os resultados obtidos demonstraram que os especialistas consideram *Portabilidade* e *Funcionalidade* como sendo as mais importantes características de qualidade para componentes de interface. Isto pode ser atribuído à facilidade e à agilidade com que se deseja construir telas das aplicações com o uso de técnicas de prototipagem rápida.

Para componentes de negócio, as características de qualidade *Funcionalidade* e *Confiabilidade* foram as melhores avaliadas, refletindo a atenção que se deve ter com as regras de negócio, realmente atendendo aos requisitos de forma correta.

As características *Eficiência* e *Confiabilidade* foram as melhores avaliadas para componentes de dados. Esses componentes normalmente interagem com recursos como discos e redes de comunicação, que são, reconhecidamente, mais lentos e estão mais sujeitos a falhas, além de poderem manipular grandes quantidades de dados.

Em relação aos componentes de infra-estrutura, observou-se que as características *Portabilidade* e *Eficiência* foram as melhores avaliadas. Esses componentes de infra-estrutura são componentes de suporte, genéricos e independentes do negócio dos usuários. Assim, há o desejo de serem reutilizados em várias aplicações. Eles também podem atender a um grande volume de solicitações, sendo pontos críticos de “gargalhos”.

Os dados consolidados para a dimensão *função arquitetural* apontaram a *Portabilidade* como a melhor avaliada, enfatizando realmente a necessidade de reutilização de componentes.

Referências

- [1] Belchior, A. D. Um modelo fuzzy para avaliação da qualidade de software. Tese de Doutorado, COOPE/UFRJ, Rio de Janeiro, Brasil. 1997.
- [2] Bertoa, M. e Vallecillo, A. Atributos de Calidad para Componentes COTS. Anais do 5º Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software. 2002.

- [3] ISO/IEC 9126. Software engineering – Product quality – Part 1: Quality model. 2001.
- [4] Kallio, P. e Ihme, T. Evolution of the Use Risks of Commercial Software Components, Proceedings of the 28th Euromicro Conference, Dortmund, Germany, September. 2002.
- [5] Oliveira, K. R. AdeQuaS: Ferramenta Fuzzy para Avaliação da Qualidade de Software. Dissertação de Mestrado, Universidade de Fortaleza (UNIFOR), Fortaleza. 2002.
- [6] Preiss, O., Wegmann, A. e Wong, J. On Quality Attribute Based Software Engineering. Proceedings of the 27th Euromicro Conference. Warsaw, Poland, September. 2001.
- [7] Simão, R. P. S. e Belchior, A. Quality Characteristics for Software Components: Hierarchy and Quality Guides. Component-Based Software Quality: Methods and Techniques. Springer-Verlag Heidelberg, “Lecture Notes in Computer Science, Junho, Alemanha. 2003.
- [8] Simão, R. e Belchior, A. Um Padrão de Qualidade para Componentes de Software, Anais do I Simpósio Brasileiro de Qualidade de Software, Gramado, Outubro. 2002a.
- [9] Simão, R. e Belchior, A. Características de Qualidade de Componentes de Software, Anais do II Workshop de Desenvolvimento Baseado em Componentes, Itaipava, Agosto. 2002b.
- [10] Villela, R. M. B. Busca e Recuperação de Componentes em Ambientes de Reutilização de Software. Tese de Doutorado, UFRJ-COPPE, Rio de Janeiro. 2000
- [11] Woodman, M. et al. Issues of CBD Product Quality and Process Quality. Proceedings of the 4th ICSE Workshop on Component-Based Software Engineering, Toronto, May. 2001.
- [12] Zadeh, L. A. Fuzzy Logic. IEEE Transaction Comput., vol. 25. 1988.

Anexo A: Características e Subcaracterísticas de Qualidade para Componentes de Software

| Características de Qualidade | Números <i>Fuzzy</i> |
|---|----------------------|
| 1. Funcionalidade | (2,04; 3,04; 3,63) |
| 1.1. Adequação | (2,51; 3,51; 3,87) |
| 1.1.1. Adequação tecnológica | (1,78; 2,77; 3,58) |
| 1.1.1.1. Adequação da metodologia | (1,91; 2,91; 3,69) |
| 1.1.1.2. Infra-estrutura tecnológica adequada | (1,87; 2,87; 3,63) |
| 1.1.1.3. Adequação do ambiente de desenvolvimento | (1,39; 2,35; 3,28) |
| 1.1.2. Adequação arquitetural | (2,27; 3,26; 3,85) |
| 1.1.3. Cobertura | (2,52; 3,52; 3,85) |
| 1.1.4. Completitude | (2,56; 3,56; 3,86) |
| 1.1.5. Corretitude | (2,94; 3,94; 4,00) |
| 1.1.6. Consistência | (2,71; 3,71; 3,97) |
| 1.2. Acurácia | (2,65; 3,65; 3,98) |
| 1.3. Autocontido | (2,09; 3,09; 3,67) |
| 1.4. Coesão funcional | (1,59; 2,59; 3,46) |
| 1.5. Interoperabilidade | (1,84; 2,84; 3,53) |
| 1.5.1. Compatibilidade de dados | (1,71; 2,71; 3,44) |
| 1.5.2. Interatividade | (2,10; 3,10; 3,68) |
| 1.5.3. Interoperabilidade interna | (1,79; 2,78; 3,52) |
| 1.6. Segurança de acesso | (2,19; 3,19; 3,62) |
| 1.6.1. Confidenciabilidade | (2,46; 3,46; 3,72) |
| 1.6.2. Imputabilidade | (1,59; 2,59; 3,31) |
| 1.6.3. Privacidade | (2,00; 3,00; 3,65) |
| 1.6.4. Vulnerabilidade | (2,39; 3,38; 3,67) |

| Características de Qualidade | Números <i>Fuzzy</i> |
|--|----------------------|
| 1.7. Conformidade com a funcionalidade | (1,73; 2,72; 3,44) |
| 2. Confiabilidade | (2,08; 3,08; 3,68) |
| 2.1. Maturidade | (2,04; 3,02; 3,62) |
| 2.1.1. Memória de estado | (2,20; 3,18; 3,69) |
| 2.1.2. Ausência de erros | (2,53; 3,51; 3,88) |
| 2.1.3. Sinalização | (2,05; 3,01; 3,61) |
| 2.1.4. Integridade | (2,12; 3,12; 3,68) |
| 2.1.5. Não volatilidade de versão | (1,80; 2,76; 3,54) |
| 2.1.6. Degradabilidade | (1,63; 2,60; 3,37) |
| 2.2. Tolerância a falhas | (2,18; 3,18; 3,77) |
| 2.2.1. Robustez | (2,39; 3,39; 3,85) |
| 2.2.2. Prevenção à contingência | (1,97; 2,97; 3,68) |
| 2.3. Recuperabilidade | (2,00; 2,99; 3,56) |
| 2.3.1. Facilidade de desfazer serviços | (1,67; 2,64; 3,37) |
| 2.3.2. Auxílio em situações de erro | (1,74; 2,74; 3,45) |
| 2.3.3. Auditabilidade | (2,34; 3,33; 3,66) |
| 2.3.4. Tratamento de erros | (2,59; 3,58; 3,93) |
| 2.4. Avaliabilidade | (1,95; 2,95; 3,64) |
| 2.4.1. Verificabilidade | (1,91; 2,91; 3,68) |
| 2.4.2. Validabilidade | (1,99; 2,99; 3,60) |
| 2.5. Conformidade com a confiabilidade | (2,33; 3,33; 3,90) |
| 3. Usabilidade | (1,67; 2,64; 3,43) |
| 3.1. Acessibilidade | (1,59; 2,56; 3,34) |

| Características de Qualidade | Números Fuzzy |
|--|----------------------|
| 3.1.1. Identificação do componente | (1,63; 2,57; 3,23) |
| 3.1.2. Classificação do componente | (1,55; 2,55; 3,45) |
| 3.2. Legibilidade | (1,74; 2,74; 3,51) |
| 3.2.1. Clareza | (1,93; 2,93; 3,65) |
| 3.2.2. Concisão | (1,66; 2,66; 3,44) |
| 3.2.3. Estilo | (1,54; 2,54; 3,33) |
| 3.2.4. Correção no uso do método | (2,02; 3,02; 3,93) |
| 3.2.5. Uniformidade de terminologia | (1,74; 2,74; 3,47) |
| 3.2.6. Uniformidade no grau de abstração | (1,63; 2,63; 3,39) |
| 3.3. Inteligibilidade | (1,61; 2,58; 3,36) |
| 3.3.1. Contextualização | (2,01; 3,01; 3,62) |
| 3.3.1.1. Estrutura básica do componente | (1,87; 2,87; 3,49) |
| 3.3.1.2. Variabilidade | (2,06; 3,06; 3,67) |
| 3.3.1.3. Dependências | (2,07; 3,06; 3,66) |
| 3.3.1.3.1. Dependências de contexto | (2,37; 3,37; 3,87) |
| 3.3.1.3.2. Dependências arquiteturais | (1,89; 2,89; 3,57) |
| 3.3.1.3.3. Dependências tecnológicas | (2,04; 3,02; 3,60) |
| 3.3.2. Justificabilidade | (1,54; 2,53; 3,39) |
| 3.3.3. Utilidade | (1,67; 2,59; 3,37) |
| 3.3.4. Relevância | (1,32; 2,27; 3,11) |
| 3.4. Facilidade de uso | (2,34; 3,34; 3,90) |
| 3.4.1. Uniformidade de comportamento | (2,36; 3,36; 3,87) |
| 3.4.2. Uniformidade de interface | (2,33; 3,32; 3,92) |
| 3.5. Apreensibilidade | (1,44; 2,43; 3,31) |
| 3.5.1. Conteúdo orientado ao usuário | (1,32; 2,31; 3,21) |
| 3.5.2. Disponibilidade de auxílios | (1,66; 2,66; 3,48) |
| 3.5.3. Suporte a usuários | (1,38; 2,38; 3,29) |
| 3.6. Operacionalidade | (2,00; 3,00; 3,68) |
| 3.6.1. Manipulabilidade | (1,98; 2,98; 3,66) |
| 3.6.1.1. Disponibilidade de artefatos | (2,06; 3,06; 3,83) |
| 3.6.1.2. Estilo arquitetural | (2,13; 3,13; 3,66) |
| 3.6.1.3. Rastreabilidade | (1,28; 2,28; 3,11) |
| 3.6.2. Controlabilidade | (2,02; 3,01; 3,69) |
| 3.7. Atratividade | (0,54; 0,98; 1,87) |
| 3.8. Conformidade com a usabilidade | (1,54; 2,43; 3,36) |
| 4. Eficiência | (2,02; 3,01; 3,72) |
| 4.1. Comportamento em relação ao tempo | (2,18; 3,18; 3,71) |
| 4.1.1. Tempo de ligação do componente | (1,39; 2,39; 3,12) |

| Características de Qualidade | Números Fuzzy |
|--|----------------------|
| 4.1.2. Capacidade de recepção e processamento de informações | (2,30; 3,30; 3,80) |
| 4.1.3. Capacidade de geração de informação | (2,25; 3,25; 3,75) |
| 4.2. Comportamento em relação aos recursos | (1,87; 2,86; 3,66) |
| 4.3. Comportamento em relação ao estado | (1,41; 2,28; 3,05) |
| 4.3.1. Suporte a manutenção do estado entre as interações | (1,60; 2,51; 3,35) |
| 4.3.2. Capacidade de serialização | (1,07; 1,95; 2,78) |
| 4.3.3. Suporte à persistência | (0,65; 1,31; 2,21) |
| 4.3.4. Suporte a transações | (2,05; 3,00; 3,53) |
| 4.4. Escalabilidade | (2,56; 3,56; 3,92) |
| 4.5. Nível de granularidade adequado | (2,11; 3,11; 3,89) |
| 4.6. Conformidade com a eficiência | (1,86; 2,86; 3,77) |
| 5. Manutenibilidade | (1,91; 2,91; 3,64) |
| 5.1. Analisabilidade | (2,01; 3,01; 3,70) |
| 5.1.1. Corretitude da arquitetura | (2,01; 3,01; 3,70) |
| 5.2. Implementabilidade | (1,88; 2,86; 3,60) |
| 5.2.1. Viabilidade econômica | (1,79; 2,77; 3,52) |
| 5.2.2. Viabilidade financeira | (1,60; 2,56; 3,46) |
| 5.2.3. Viabilidade mercadológica | (1,77; 2,77; 3,51) |
| 5.2.4. Viabilidade de cronograma | (1,99; 2,99; 3,68) |
| 5.2.5. Viabilidade de mão de obra | (1,94; 2,93; 3,65) |
| 5.2.6. Viabilidade tecnológica | (2,32; 3,31; 3,90) |
| 5.3. Modificabilidade | (2,07; 3,06; 3,70) |
| 5.3.1. Modificabilidade corretiva | (2,24; 3,24; 3,73) |
| 5.3.2. Evolutibilidade | (1,76; 2,75; 3,63) |
| 5.3.3. Extensibilidade | (2,12; 3,12; 3,73) |
| 5.4. Estabilidade | (1,99; 2,99; 3,62) |
| 5.5. Testabilidade | (1,38; 2,38; 3,32) |
| 5.5.1. Autoteste de inicialização | (1,46; 2,46; 3,41) |
| 5.5.2. Pacote de testes | (1,54; 2,54; 3,47) |
| 5.5.3. Geração de relatórios de testes | (0,87; 1,87; 2,83) |
| 5.6. Conformidade com a manutenibilidade | (1,86; 2,86; 3,75) |
| 6. Portabilidade | (2,16; 3,16; 3,77) |
| 6.1. Adaptabilidade | (1,95; 2,95; 3,61) |
| 6.1.1. Generalidade | (1,94; 2,93; 3,61) |
| 6.1.1.1. Tipificação | (1,94; 2,93; 3,61) |
| 6.1.2. Capacidade para ser configurado | (1,96; 2,96; 3,61) |
| 6.2. Capacidade para ser instalado | (2,31; 3,31; 3,87) |
| 6.3. Coexistência | (2,25; 3,25; 3,87) |
| 6.4. Substituíbilidade | (2,41; 3,40; 3,83) |
| 6.5. Conformidade com a portabilidade | (1,70; 2,69; 3,53) |