

AVALIAÇÃO DA QUALIDADE DE SOFTWARES DE APOIO À DECISÃO

Francisco Rodrigues Lima Junior (EESC - USP)

eng.franciscojunior@gmail.com

Luiz Cesar Ribeiro Carpinetti (EESC - USP)

carpinet@sc.usp.br

Lauro Osiro (EESC - USP)

lauro.osiro@gmail.com

Gilberto Miller Devos Ganga (EESC - USP)

ganga@dep.ufscar.br



Em ambientes organizacionais, diante de decisões que possuem alto nível estratégico, envolvem várias pessoas e requerem alto grau de estruturação, o uso de sistemas computacionais de apoio à decisão (Decision Support System - DSS) tem se tornado cada vez mais difundido. Esses sistemas apresentam diversas vantagens decorrentes do uso de modelos de decisão para realização de simulações. Contudo, para que um DSS forneça decisões efetivas é necessário escolher o sistema que melhor se adeque às necessidades do cliente. Neste contexto, o uso de modelos para avaliação da qualidade de DSS é capaz de apoiar a escolha do software mais adequado aos requisitos explicitados pelo cliente. Diante disso, este estudo propõe um modelo para avaliação da qualidade de softwares de apoio à decisão desenvolvido a partir de uma pesquisa bibliográfica e de um estudo de caso. O estudo de caso contempla um projeto acadêmico no qual alguns pesquisadores precisam escolher o DSS mais adequado às suas necessidades. Neste cenário, o modelo proposto foi utilizado para avaliar a qualidade de dois produtos de acordo com as normas ISO/IEC9126 e ISO/IEC14598. Mediante os requisitos dos pesquisadores do projeto e os pesos relativos de cada métrica utilizada na avaliação, o software Expert Choice Desktop foi o produto escolhido com 64,9% de preferência contra 35,1% do Super Decisions.

Palavras-chaves: Sistemas de apoio à decisão; Qualidade de Software; Método de Análise Hierárquica

1. Introdução

A escassez de recursos e a necessidade de obter soluções que maximizem benefícios e minimizem custos constituem cenários comuns em várias disciplinas das ciências naturais e sociais. Em muitos desses cenários, a gestão de recursos implica na estruturação e análise de problemas para justificar escolhas, o que contribui para que a tomada de decisão (*decision making*) tenha sua importância cada vez mais reconhecida em diversas áreas do conhecimento (ROY, 1986; ANDRADE, 1998).

Um processo genérico para tomada de decisão pode ser subdividido nas etapas de percepção, reconhecimento do problema, criação de alternativas, avaliação de alternativas e decisão final. Em ambientes organizacionais, diante de decisões que possuem alto nível estratégico, envolvem várias pessoas e requerem alto grau de estruturação, o uso de sistemas computacionais de suporte à decisão tem se tornado cada vez mais difundido (ROY, 1986; ANDRADE, 1998).

Turban (2005) define sistema de suporte à decisão (*Decision Support System – DSS*) como um tipo de sistema baseado em interação com o computador que auxilia agentes decisórios a utilizar dados e modelos para resolver problemas de decisão desestruturados. Desta forma, os DSS acoplam recursos intelectuais dos indivíduos a capacidades dos computadores para melhorar a qualidade das decisões.

O uso desses sistemas apresenta benefícios como maior facilidade de manipulação do modelo que em um sistema real, pois a experimentação é mais fácil e não interfere na rotina da organização. Além disso, o custo de uma análise da modelagem é muito mais baixo que o custo de um experimento similar conduzido em um sistema real. De forma análoga, o custo de erros de decisão cometidos durante um experimento de tentativa e erro é muito mais baixo que quando os modelos são usados em tempo real (HOLSAPPLE; WHINSTON, 1996; TURBAN, 2005).

Contudo, para que um DSS forneça decisões efetivas é necessário escolher o *software* que melhor se adeque às necessidades da organização e dos futuros usuários. Neste contexto, o uso de modelos para avaliação da qualidade desses produtos é capaz de apoiar a escolha do *software* mais adequado aos requisitos explicitados pelo cliente (PFLEEGER; FITZGERALD, 1991; XENOS; AZUMA; 1996; CHRISTODOULAKIS; 1997). Diante disso, este estudo propõe um modelo para avaliação da qualidade de *softwares* de apoio à decisão desenvolvido a partir de uma pesquisa bibliográfica e de um estudo de caso. O estudo de caso contempla um projeto acadêmico no qual alguns pesquisadores precisam escolher o DSS mais adequado às suas necessidades. Neste cenário, o modelo de avaliação proposto foi utilizado para avaliar a qualidade de dois produtos de acordo com as recomendações das normas vigentes.

A estrutura deste artigo está organizada da seguinte forma: a Seção 2 apresenta conceitos e normas referentes à avaliação da qualidade de softwares; a Seção 3 descreve a abordagem de um método multicritério utilizado neste estudo; a Seção 4 contempla a metodologia utilizada nesta pesquisa; a Seção 5 contempla a construção e aplicação do modelo; e a Seção 6 apresenta algumas considerações finais sobre este trabalho.

2. Avaliação da Qualidade de Software

A qualidade de um “produto de *software*” pode ser definida pela totalidade de características que lhe confere a capacidade de satisfazer as necessidades implícitas de seus usuários, podendo ser avaliada em três diferentes perspectivas: qualidade externa, qualidade interna e qualidade em uso (XENOS; AZUMA; 1996; CHRISTODOULAKIS; 1997; ABNT, 1999a).

A qualidade interna é a perspectiva da qualidade visível aos desenvolvedores e é avaliada nos produtos intermediários do *software* (diagramas, códigos de programas, documentação etc.). As métricas internas devem ser escolhidas pelo desenvolvedor do *software*. Como a qualidade de um produto depende fortemente da qualidade do seu processo de desenvolvimento, a qualidade interna irá refletir na qualidade externa do *software* (ABNT, 1999a; ABNT, 1999b).

A qualidade externa é a perspectiva da qualidade visível aos usuários do sistema. Os aspectos que definem a qualidade externa devem ser avaliados em tempo de execução do *software* e, por isso, a avaliação englobará um sistema em execução que envolve *software* e *hardware*. Desta forma, os valores das métricas dependem de fatores além do *software*, já que se avalia o *software* como parte de um sistema em operação (ABNT, 1999a; ABNT, 1999b).

Já a qualidade em uso é o efeito combinado das características de qualidade internas e externas de um *software* em uma situação particular de uso. A qualidade em uso não é medida por meio das propriedades do *software* e sim pela capacidade do *software* de atingir metas específicas com efetividade, produtividade, segurança e satisfação em um contexto de uso especificado (ABNT, 1999a; ABNT 1999b).

A avaliação da qualidade de software requer o desenvolvimento de um modelo de qualidade para avaliação de *software*. Segundo Azuma (1999), estes modelos asseguram a qualidade por meio de um conjunto de características e do relacionamento entre estas, sendo capazes de prover uma base para a especificação de requisitos de qualidade.

Existem diversos guias de referência que podem ser usados para assegurar a qualidade de *software*, tais como o *Project Management Body of Knowledge* (PMBok) e o *Capability Maturity Model* (AZUMA, 1996), mas que servem especificamente para orientar desenvolvedores de *software* na garantia da qualidade interna e na gestão do processo de *software*. Além desses modelos, a avaliação da qualidade interna pode ser conduzida usando diversos modelos da literatura acadêmica. Dentre esses, incluem-se os modelos de Redmond e Ah-Chen (1990) e de Pfleeger e Fitzgerald (1991), que apoiam o desenvolvimento de métricas e a avaliação da qualidade interna.

Em outra via, diversos autores vem propondo modelos para avaliação da qualidade externa de *softwares* voltados para aplicações específicas. Hlupic *et al.* (1999) desenvolveu um modelo para avaliação da qualidade externa de *softwares* de simulação; Lima Junior e Fondazzi (2010) propuseram um modelo para avaliação da qualidade externa de *softwares* de gestão de projetos; Café, Santos e Macedo (2001) propuseram um modelo para apoiar a escolha de *software* de automação de bibliotecas que utiliza métricas com diferentes pesos.

A grande semelhança entre esses modelos é que a maioria é construída a partir de diretrizes propostas por entidades competentes mundialmente reconhecidas, como comitês de desenvolvedores de *software* e órgãos de normalização. No Brasil, a Associação Brasileira de Normas Técnicas recomenda o uso de algumas normas elaboradas pela *International Organization for Standardization* e pela *International Electrotechnical Commission*.

2.1 ISO/IEC14598

A série de normas ISO/IEC14598 (intitulada “Engenharia de *Software* - Avaliação de qualidade de *software*”) tem a finalidade de apoiar o desenvolvimento e a aquisição de *software* que atenda as necessidades do usuário final (ABNT, 1999a). Segundo a ISO/IEC14598 (ABNT, 1999a), um processo de avaliação da qualidade de *software* deve atender a quatro características básicas, as quais são definidas no Quadro 1.

Característica	Definição
Repetibilidade	A avaliação repetida de um mesmo produto, pelo mesmo avaliador, com a mesma especificação deve reproduzir resultados que podem ser aceitos como idênticos.
Reprodutibilidade	A avaliação do mesmo produto, com a mesma especificação de avaliação, por um avaliador diferente, deve produzir resultados que podem ser aceitos como idênticos.
Imparcialidade	A avaliação não deve ser influenciada frente a nenhum resultado particular.
Objetividade	Os resultados da avaliação devem ser factuais, ou seja, não influenciados pelos sentimentos ou opiniões do avaliador.

Quadro 1 – Características desejáveis à avaliação de *software*

As atividades críticas do processo de avaliação de *software* consistem na definição e medição de critérios, já que este produto é intangível e isso pode acarretar na formulação de critérios subjetivos e, conseqüentemente, em uma avaliação dificilmente replicável. Neste contexto, a série de normas ISO/IEC9126 foi desenvolvida para apoiar a definição de métricas de qualidade de qualquer tipo de *software*.

2.2 ISO/IEC14598

A série de normas ISO/IEC9126 (intitulada “Engenharia de *Software* - Qualidade de produto”) descreve um modelo de qualidade sob os aspectos da qualidade interna, externa e em uso e propõe um conjunto de 6 características para avaliação da qualidade de *software* (ABNT, 2003; JUNG, 2007), as quais definidas no Quadro 2.

Característica	Definição
Usabilidade	Capacidade do produto de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas.
Funcionalidade	Capacidade do produto de prover funções que atendam às necessidades implícitas e explícitas, quando o <i>software</i> estiver sendo utilizado em condições especificadas. Esta característica está relacionada ao que o <i>software</i> faz, enquanto as outras estão relacionadas a como e quando o <i>software</i> faz.
Confiabilidade	Capacidade do <i>software</i> de manter um nível de desempenho especificado, quando usado em condições especificadas.
Eficiência	Capacidade do <i>software</i> de apresentar desempenho apropriado, relativo à quantidade de recursos usados, sob condições especificadas.
Manutenibilidade	Capacidade do <i>software</i> de ser modificado. As modificações podem incluir correções, melhorias ou adaptações de <i>software</i> devido a mudanças no ambiente e nos requisitos funcionais.
Portabilidade	Capacidade do produto de ser transferido de um ambiente para outro.

Quadro 2 – Características que definem a qualidade de *software*

Conforme ilustra a Figura 1, cada característica da qualidade é definida por um conjunto de subcaracterísticas correspondentes e sua avaliação também envolve a conformidade com as normas e convenções relacionadas às mesmas.



Fonte: ABNT (2003)

Figura 1 – Conjunto de subcaracterísticas da qualidade de *software*.

A definição de cada uma dessas subcaracterísticas mostradas na Figura 1 é apresentada no Quadro 3.

Subcaracterística		Definição
Usabilidade	Inteligibilidade	Capacidade de possibilitar um usuário compreender se o <i>software</i> é apropriado e como ele pode ser usado para tarefas e condições de uso específicas.
	Apreensibilidade	Capacidade de possibilitar ao usuário entender a sua aplicação.
	Operacionalidade	Capacidade de propiciar ao usuário operá-lo e controlá-lo.
	Atratividade	Capacidade de ser atrativo ao usuário.
Funcionalidade	Adequação	Capacidade de prover um conjunto apropriado de funções para tarefas e objetivos do usuário especificados, e também afeta a operacionalidade.
	Acurácia	Capacidade de prover, com o grau de precisão necessário, resultados ou efeitos corretos.
	Interoperabilidade	Capacidade de interagir com um ou mais sistemas especificados (compatibilidade).
	Segurança de acesso	Capacidade de proteger informações e dados, de forma que as pessoas não autorizadas não possam lê-los ou modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.
Confiabilidade	Maturidade	Capacidade de evitar falhas decorrentes de defeitos no <i>software</i> .
	Tolerância a falhas	Capacidade de garantir um nível de desempenho especificado em caso de defeitos no <i>software</i> ou de violação de sua interface especificada.
	Recuperabilidade	Capacidade de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Eficiência	Comportamento em relação ao tempo	Capacidade de fornecer tempos de resposta e processamento apropriados quando o <i>software</i> executa suas funções.
	Utilização de recursos	Capacidade de usar tipos e quantidades apropriados de recursos enquanto o <i>software</i> executa sob condições apropriadas.
Manutenibilidade	Analisabilidade	Capacidade de permitir o diagnóstico de deficiências ou causas de falhas no <i>software</i> .
	Modificabilidade	Capacidade de permitir que uma modificação seja implementada.
	Estabilidade	Capacidade de evitar efeitos inesperados decorrentes de tais modificações.
	Testabilidade	Capacidade de permitir ser avaliado quando modificado.
Portabilidade	Adaptabilidade	Capacidade de ser adaptado para ambientes especificados sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo <i>software</i> considerado.
	Capacidade de ser instalado	Capacidade de ser instalado em um ambiente especificado. É importante por afetar a operacionalidade e a adequação.
	Coexistência	Capacidade de coexistir com outros produtos de <i>softwares</i> independentes em um ambiente compartilhando recursos comuns.
	Capacidade para substituir	Ser usado em substituição a outro produto de <i>software</i> especificado, com o mesmo propósito e no mesmo ambiente.

Quadro 3 – Subcaracterísticas da qualidade de *software*

Analisando o Quadro 3, é possível constatar que a forma pela qual as subcaracterísticas da qualidade são definidas pela ISO/IEC9126 não permite sua medição direta devido à subjetividade presente na interpretação e medição destas. Por isso, é necessário definir métricas correspondentes a cada subcaracterística da qualidade. Todo atributo quantificável externo ou interno ao *software* que esteja relacionado a uma característica da qualidade pode ser definido como uma métrica. Neste sentido, a elaboração de um modelo de avaliação da qualidade requer que os futuros usuários e demais *stakeholders* definam um conjunto de

atributos que expressem seus requisitos e que estejam relacionados às características da qualidade consideradas (AZUMA, 1996; JUNG, 2007).

O Quadro 4 exemplifica o desdobramento de uma subcaracterística da qualidade em atributos quantificáveis, definidos a partir dos requisitos dos usuários e do contexto de uso, realizado com base na ISO/IEC9126.

Característica	Subcaracterística	Atributos
Funcionalidade	Adequação	O software provê Agendamento de Tarefas?
		Provê Gráfico de <i>Gantt</i> ?
		Provê Redes de atividades
		Provê gerenciamento da Estrutura Organizacional?

Fonte: Lima Junior e Fondazzi (2010)

Quadro 4 – Desdobramento de uma característica da qualidade em atributos

A avaliação de um *software* utilizando muitos critérios pode ser inviável devido à grande quantidade de tempo despendida para avaliação. Por isso, a ISO/IEC9126 deve servir de referência para a criação de modelos de avaliação adequados com o tempo, recursos humanos e informações disponíveis (ABNT, 1999b). Obviamente, de acordo com o contexto de uso, as métricas de avaliação de *software* possuirão níveis de importância diferentes. Portanto, os valores relativos aos pesos das métricas também devem ser determinados pelas partes envolvidas.

Porém, um fator complicador é a natureza subjetiva da medição desses pesos relativos, já que a dificuldade de um agente decisório de expressar suas preferências sobre esses pesos com valores precisos pode comprometer a avaliação.

Para lidar com esse problema, métodos de decisão multicritério capazes de lidar com incertezas podem ser utilizados (BHUTTA; HUQ, 2002).

Um método bastante utilizado devido à sua simplicidade e facilidade de entendimento é o Método de Análise Hierárquica (AHP) (SAATY, 2008). Por ser capaz de lidar com fatores intangíveis e subjetivos, o AHP pode ser de grande valor na avaliação dos pesos das características e subcaracterísticas da qualidade de *software*.

3. Método AHP

O AHP foi desenvolvido por Thomas Saaty na década de 1970 e atualmente é provavelmente o método de decisão mais utilizado no Brasil. O AHP é um método compensatório capaz de priorizar alternativas mediante a consideração de múltiplos critérios em situações em que estes possuem natureza intuitiva, racional ou qualitativa (BHUTTA; HUQ, 2002; SAATY, 2008).

Conforme mostra a Figura 2, no AHP a representação de um problema de decisão é feita em uma estrutura hierárquica com a finalidade de capturar os elementos básicos do problema. Em alguns casos, os critérios de decisão são decompostos em um conjunto de subcritérios que também devem ser representados na hierarquia. Nessa estrutura, os elementos são comparados par a par com outros elementos do mesmo nível com a finalidade de deduzir preferências. Os julgamentos comparativos necessários ao modelo são obtidos a partir da opinião de

especialistas no domínio de problema em questão. Por último, é realizada uma recomposição hierárquica que sintetiza as preferências obtidas anteriormente para encontrar a melhor decisão (SAATY, 1980; BHUTTA; HUQ, 2002).

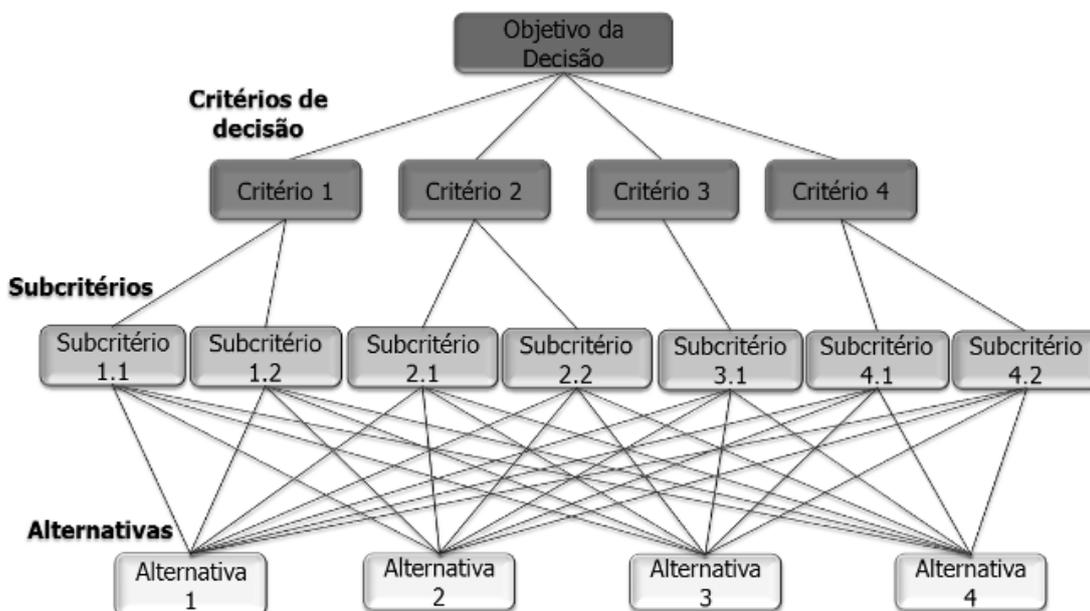


Figura 2 - Estrutura de uma hierarquia de decisão

Para representar a opinião dos especialistas durante as comparações par a par entre os elementos, Saaty (1980) propôs a escala de medidas mostrada no Quadro 5.

Pontuação Numérica	Julgamento Verbal ou Preferência
1	Igual Importância
3	Importância pequena de uma sobre a outra
5	Importância grande ou essencial
7	Importância muito grande ou demonstrada
9	Importância absoluta
2, 4, 6, 8	Valores Intermediários

Quadro 5 - Escala de medidas.

A realização de julgamentos comparativos faz necessária a verificação da coerência das comparações (*consistency ratio* – CR). Considera-se frequentemente que os julgamentos são coerentes se não ultrapassarem o limite $CR \leq 0,10$. Alguns autores recomendam rever os julgamentos em que $CR \geq 0,20$ (SAATY, 1980; SAATY, 2008). Após atestar a consistência dos julgamentos e sintetizar as preferências, deve-se realizar a classificação, eliminação e/ou ordenação de alternativas, o que varia conforme a natureza do problema de decisão (ROY, 1996).

Em suma, o AHP é um método de fácil entendimento que não requer cálculos complexos. A maior dificuldade de aplicação parece estar associada à necessidade de requerer demasiados julgamentos nos casos em que o modelo possui muitos critérios. Diante da capacidade de lidar com avaliações qualitativas e de estruturação hierárquica, a utilização do AHP se mostra bastante adequada à avaliação da qualidade de *software*.

4. Procedimentos de Pesquisa

O método de pesquisa utilizado para condução deste estudo está estruturado em três etapas, conforme ilustra a Figura 3.

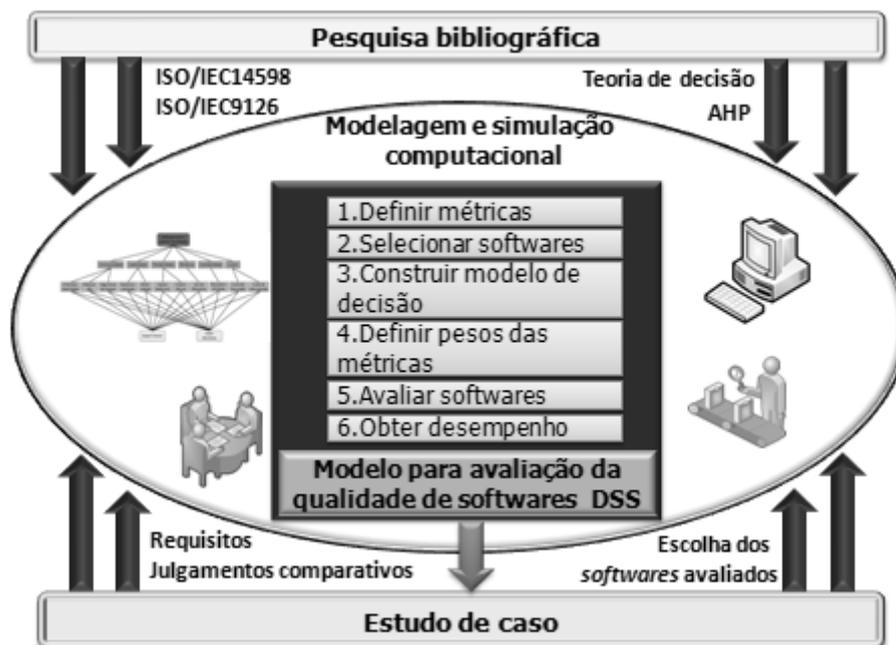


Figura 3 – Estrutura da metodologia de pesquisa

A **pesquisa bibliográfica** visou desenvolver competências para a construção do modelo de avaliação da qualidade de *softwares* de apoio à decisão. Para isso, foram consultados a série de normas ISO/IEC9126, ISO/IEC14598, livros e artigos sobre teoria de decisão e AHP.

A etapa de **modelagem e simulação computacional** foi conduzida paralelamente ao **estudo de caso** com a finalidade de coletar *inputs* necessários ao desenvolvimento do modelo: requisitos de *software* funcionais e não funcionais, preferências entre métricas e escolha dos *softwares* de apoio à decisão a serem avaliados.

Um modelo de decisão foi construído para coletar as preferências dos pesquisadores do projeto e calcular a importância relativa das métricas. Seguindo as definições de Pidd (2004), as simulações realizadas neste modelo são caracterizadas como discretas, estáticas e determinísticas.

Por último, a avaliação dos *softwares* selecionados pelos pesquisadores foi realizada por meio da análise documental de todo material de suporte disponibilizado pelos fabricantes, do estudo e do uso dos *softwares* pelos avaliadores.

5. Apresentação e discussão dos resultados

Seguindo a ISO/IEC9126, o modelo de avaliação da qualidade de *softwares* DSS é baseado em um conjunto de atributos desejáveis ao *software* que estão relacionados a subcaracterísticas da qualidade. Cada característica da qualidade deve ser definida por uma ou

mais subcaracterísticas, que, por sua vez, deve ser avaliada por meio de um ou mais atributos objetivos. A Figura 4 ilustra a hierarquia das métricas do modelo.

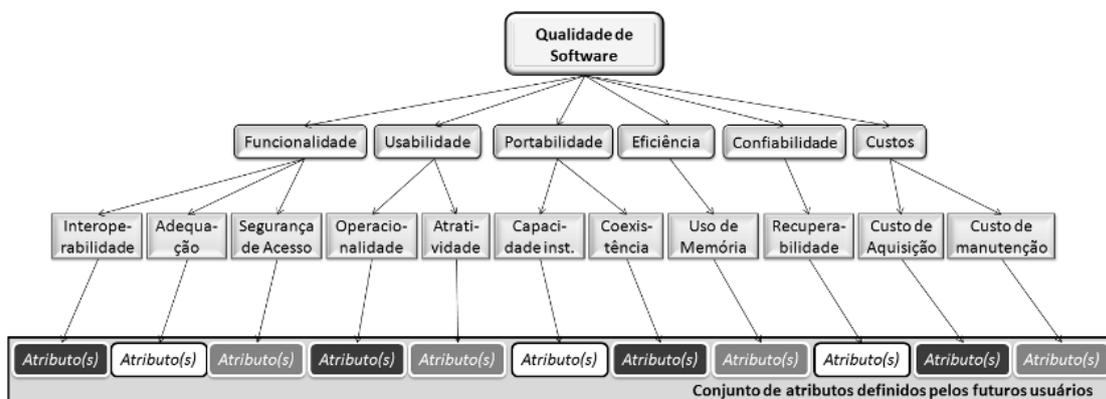


Figura 4 - Hierarquia do problema de avaliação de *softwares*

5.1 Seleção de *softwares*

O principal critério considerado durante a seleção de *softwares* para avaliação foi a capacidade do produto de operacionalizar um ou mais métodos multicritério voltados para situações de decisão sob incerteza. Além disso, foram considerados critérios como disponibilidade de informações sobre o produto e facilidade de *download*. Portanto, os *softwares* escolhidos foram o *Expert Choice Desktop 11.5* e o *Super Decisions 2.0.8*.

O *Expert Choice* é um DSS que permite sintetizar *inputs* qualitativos e quantitativos de múltiplos *stakeholders* usando o método AHP. Já o *Super Decisions* permite modelar problemas de decisão em formato de redes e resolvê-los usando o Método de Análise em Redes, que é um método de decisão multicritério baseado no AHP, mas que possibilita representar relações de dependência entre os elementos da rede (SAATY, 1996).

5.2 Definição dos pesos

Os pesos relativos de cada característica da qualidade e de suas subcaracterísticas devem ser definidos por meio da coleta de julgamentos de pessoas envolvidas. Para facilitar a coleta e o processamento desses dados, podem ser utilizados modelos de decisão desenvolvidos nos próprios *softwares* avaliados.

A Figura 5 mostra a interface do modelo utilizado para determinar os pesos das características e subcaracterísticas da qualidade. Foram coletadas as preferências de três pesquisadores que forneceram julgamentos obedecendo à escala de Saaty.

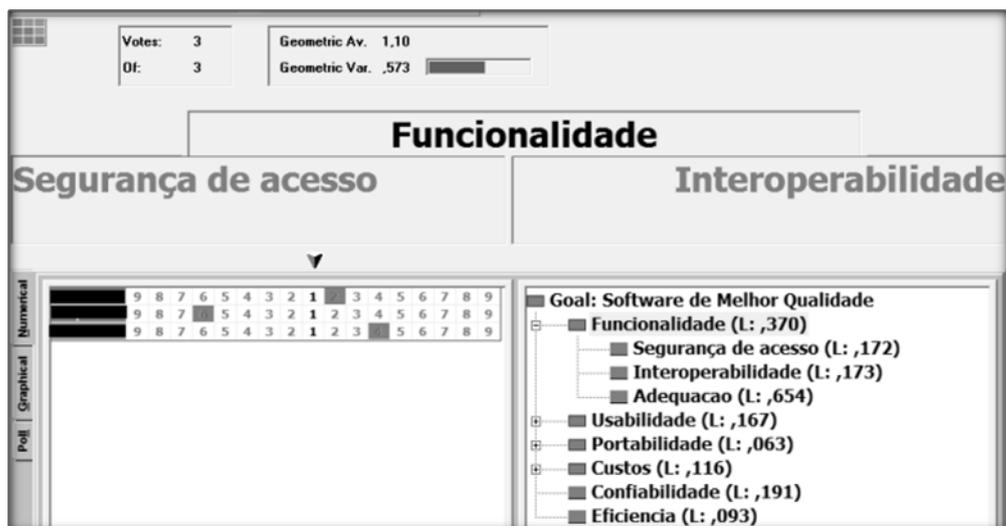


Figura 5 – Interface do modelo de decisão

Após o julgamento dos pesos, é necessário verificar a consistência destes e sintetizar os índices de preferências globais. Seguindo os procedimentos do AHP, os julgamentos dos agentes decisórios devem ser agrupados para, posteriormente, obter o vetor de prioridades.

No estudo de caso, o valor do índice de inconsistência fornecido pelo modelo de decisão foi de 0.01, o que evidencia que os julgamentos coletados foram consistentes. A partir da síntese de prioridades das características e das subcaracterísticas da qualidade, o modelo forneceu os pesos globais normalizados mostrados na Figura 6.

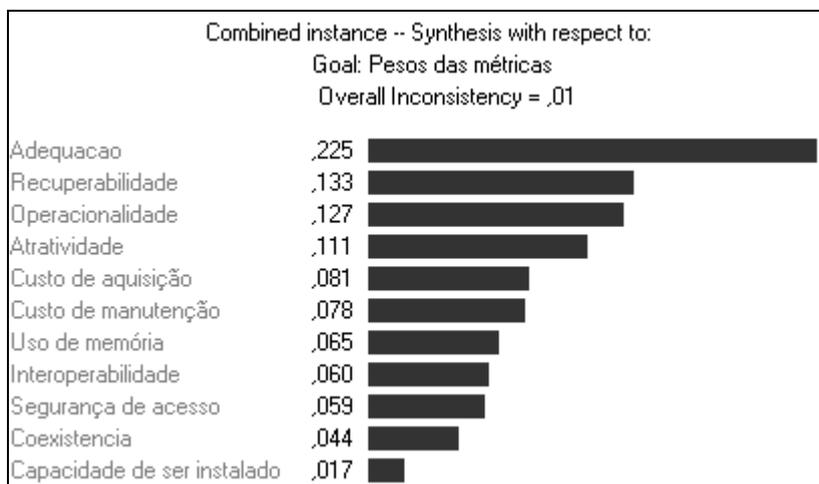


Figura 6 – Pesos das métricas

5.3 Avaliação dos Softwares

A avaliação de *softwares* deve ser conduzida por um ou mais avaliadores utilizando um conjunto de atributos predefinidos pelo cliente que deve expressar “o que o *software* deve fazer” e “como isso deve ser feito”. Além disso, esses atributos devem ser facilmente mensuráveis pelo usuário.

O Quadro 6 apresenta o conjunto de atributos definidos pelos pesquisadores do projeto para avaliação dos *softwares*. Além dos atributos para as características da qualidade definidas pela ISO/IEC9126, foram definidos atributos relacionados a custos, já que este critério é pertinente em processos de escolha. Além do conjunto de atributos utilizados na avaliação, o Quadro 6 apresenta o resultado da avaliação do *Expert Choice Desktop* (EC) e do *Super Decisions* (SD) em relação a cada atributo.

Subcaracterísticas	Atributos
Adequação (S1)	O <i>software</i> oferece algum recurso que oriente o usuário passo-a-passo durante o processo de m
	O <i>software</i> fornece uma visualização gráfica sistêmica dos elementos componentes do modelo
	O <i>software</i> provê visualização gráfica da Hierarquia de decisão?
	O <i>software</i> provê visualização gráfica da Rede de nós?
	O <i>software</i> permite a determinação de pesos usando a técnica <i>rating mode</i> ?
	O <i>software</i> oferece alternativas de interface para entrada de julgamentos?
	O <i>software</i> possui mecanismo gráfico para criar e ponderar relações de dependência entre nós
	O <i>software</i> possibilita a agregação de julgamentos de múltiplos agentes decisórios?
	O <i>software</i> possibilita a atribuição de pesos aos julgamentos dos agentes decisórios?
	O <i>software</i> oferece diferentes métodos para agregação de julgamentos dos decisórios?
	O <i>software</i> possibilita de forma simples a inclusão de critérios e alternativas após síntese de re
	O <i>software</i> fornece índice de consistência dos julgamentos?
	O <i>software</i> permite o uso de preferências <i>fuzzy</i> para expressar julgamentos?
	O <i>software</i> permite a geração de relatórios?
	O <i>software</i> realiza análise de sensibilidade?
	O <i>software</i> permite fazer análise de sensibilidade utilizando diferentes recursos gráficos?
	O <i>software</i> apresenta demonstração de uso no site oficial do fabricante produto?
O <i>software</i> apresenta tutorial no site oficial ou internamente?	
O <i>software</i> possui guias rápidos de ajuda?	
O <i>software</i> contém arquivo de ajuda (“ <i>tips</i> ”)?	
Interoperabilidade (S2)	O <i>software</i> interage com outros sistemas permitindo importar/exportar dados?
	É possível integrar o produto com algum <i>software</i> de construção de sistemas <i>fuzzy</i> ?
	É possível integrar o produto com algum <i>software</i> do <i>Microsoft Office</i> ?
Segurança de acesso (S3)	O <i>software</i> permite o uso de senhas de acesso para cada usuário?
	O <i>software</i> propicia acesso às informações em diferentes níveis hierárquicos?
Recuperabilidade (S4)	Em caso de falha no sistema, o <i>software</i> provê recuperação de dados não salvos?
Operacionalidade (S5)	O <i>software</i> dispõe de atalhos das funcionalidades básicas (salvar, abrir, etc) no menu gráfico pr
	O <i>software</i> dispõe de atalhos das funcionalidades principais (inserir critério, alternativa, etc) n
Atratividade (S6)	É possível customizar os menus gráficos na interface do <i>software</i> ?
Uso de Memória (S7)	O <i>software</i> utiliza menor quantidade de memória RAM dentre os <i>softwares</i> avaliados?
Capacidade para ser instalado (S8)	Mediante os pré-requisitos de instalação, o <i>software</i> pode ser considerado fácil de instalar e co
Coexistência (S9)	O <i>software</i> compartilha recursos de <i>hardware</i> com outros aplicativos sem falhas frequentes?
Custo de aquisição (S10)	A distribuição do <i>software</i> é gratuita?
Custo de manutenção (S11)	Caso o fabricante ofereça suporte ao usuário, este é gratuito?
	O recebimento de atualizações de novas versões do <i>software</i> é gratuito?

Quadro 6 – Listagem de atributos e avaliação de *softwares Expert Choice* e *Super Decisions*

5.4 Desempenho Final

Por último, deve-se calcular o desempenho final dos *softwares* por meio da multiplicação dos pesos dos atributos com as pontuações normalizadas de cada alternativa. A pontuação das alternativas em relação a cada subcaracterística da qualidade é equivalente à quantidade relativa de atributos correspondentes que o *software* é capaz de atender satisfatoriamente. Por exemplo, durante a avaliação dos produtos no estudo de caso, o *software* SD atendeu a 11 dos 20 atributos avaliados na subcaracterística S1 e, portanto, a pontuação obtida foi igual a 0,55.

A Tabela 1 mostra o conjunto de valores utilizados para calcular o desempenho final dos *softwares* avaliados no estudo de caso, bem como o valor do índice de preferência final de cada alternativa (desempenho final normalizado).

Características	Subcaracterísticas	Peso	Pontuação das alternativas	
			EC	SD
C1	S1	0,225	0,65	0,50
	S2	0,060	0,66	0,33
	S3	0,059	1,00	0,00
C2	S4	0,133	1,00	0,00
C3	S5	0,127	1,00	0,50
	S6	0,111	1,00	0,00
C4	S7	0,065	1,00	0,00
C5	S8	0,017	1,00	1,00
	S9	0,044	0,00	1,00
C6	S10	0,081	0,00	1,00
	S11	0,078	0,00	0,50
Desempenho final normalizado			0,649	0,351

Tabela 1 – Cálculo do desempenho final dos *softwares*

De acordo com os dados mostrados na Tabela 1, o *Expert Choice Desktop* é o *software* mais adequado às necessidades do projeto de pesquisa, com preferência de aproximadamente 64,9%. Em contrapartida, o *Super Decisions* se mostra como uma alternativa inviável, com preferência de aproximadamente 35,1%. Dentre os 35 atributos de avaliação propostos no modelo, o EC atendeu a 22 (62,9%) enquanto o SD atendeu a 16 (45,7%).

5.5 Análise e discussão dos resultados

Os resultados mostram que o EC possui muito mais funcionalidades ao usuário para construção de modelos que o SD. Dentre essas, incluem-se suporte à decisão em grupo, proteção das informações por meio de senhas e diferentes recursos gráficos para construção do modelo, coleta de julgamentos e análise de sensibilidade. Embora ambos os produtos permitam rastrear julgamentos inconsistentes, esses verificam a consistência de forma diferente. Enquanto é necessário requerer um relatório de inconsistência dos julgamentos no SD, o EC fornece um valor de inconsistência global para os julgamentos de um usuário específico ou do grupo em tempo real na própria interface de coleta de dados.

Além disso, o EC se destacou por permitir a integração com o *Microsoft Project* e com bancos de dados desenvolvidos em *Oracle*. Todavia, ambos os *softwares* não possibilitam o uso de números *fuzzy* para representar imprecisão nos dados do modelo e não implementam a técnica *rating mode*, que é bastante útil por ser capaz de diminuir a quantidade de julgamentos requeridos pelo modelo (SAATY, 2008).

Quanto aos requisitos não funcionais, no que diz respeito à confiabilidade, o EC apresentou melhor desempenho por ser capaz de recuperar arquivos não salvos em caso de falhas do sistema. O *software* também apresentou melhor desempenho em eficiência por usar em média 60% da memória de trabalho utilizada pelo SD. No entanto, o SD se destacou em portabilidade por ser capaz de coexistir em um ambiente compartilhando recursos de *hardware* com outros aplicativos sem gerar falhas. Enquanto o SD não apresentou nenhuma falha durante os testes realizados, o EC travou nas situações em que foi submetido a alguns dias de funcionamento sem interrupções.

Por último, em relação aos custos, o SD apresenta um desempenho muito melhor que o EC. O SD foi desenvolvido por um grupo que fornece aos usuários o *software*, serviços de suporte, tutoriais e atualizações do produto gratuitamente. Contudo, a empresa também fabrica um DSS comercial com bastante projeção no mercado (*Decisions Lens*). Já o EC é o principal produto comercializado pela empresa e, por isso, os módulos adicionais do *software*, cursos de treinamento e consultoria para implantação do sistema são vendidos para os usuários. Portanto, nota-se que a avaliação dos custos foi influenciada pelo papel estratégico que os produtos possuem nos negócios de seus fabricantes e não reflete as vantagens financeiras decorridas da aquisição dos produtos avaliados em relação a preços de mercado.

Diante dos resultados da avaliação, constata-se que o EC está longe de satisfazer todos os requisitos dos usuários do projeto, principalmente pela ausência de mecanismos que permitam o uso de recursos advindos da Teoria dos conjuntos *fuzzy*. Ainda assim, o EC foi adquirido pelo projeto de pesquisa por ser a solução disponível mais viável. Apesar da baixa pontuação obtida pelo SD, este *software* parece ser adequado para fins didáticos e para modelagem de problemas que não requerem decisão em grupo.

6. Considerações Finais

Este estudo propôs um modelo para avaliação da qualidade de *softwares* de apoio à decisão e avaliou a qualidade de dois *softwares* que implementam métodos multicritério. As alternativas foram avaliadas utilizando 6 aspectos de qualidade, os quais foram mensurados por meio de 35 atributos objetivos. Apesar de ter sido desenvolvido dentro de um cenário específico, o modelo pode ser facilmente adaptado e utilizado para apoiar a escolha de *softwares* em outros cenários. Além disso, o modelo possui a vantagem de não requerer *softwares* específicos para modelagem e simulação, já que os próprios produtos avaliados podem ser usados na construção do modelo para ponderar as métricas.

Os resultados da avaliação mostraram que o *Expert Choice* é o *software* de melhor qualidade e, portanto, o mais adequado para o projeto. Porém, vale ressaltar que a avaliação da qualidade foi realizada sob a perspectiva dos usuários do projeto e que cada cenário de uso implica em atributos de avaliação específicos. Neste sentido, podem existir situações em que a escolha pelo *Super Decision* seja mais viável aos usuários.

Assim como não existe um único conjunto de requisitos adequado para avaliar a qualidade de um tipo de *software* em vários cenários, não existe uma única maneira de coletar requisitos e de conduzir a avaliação de *software*. No entanto, é necessário se basear em modelos de referência para qualidade, modelos de avaliação de qualidade previamente existentes e/ou normas vigentes para que todos os aspectos pertinentes à qualidade de *software* sejam incorporados ao modelo de avaliação.

Agradecimentos

À CAPES e à FAPESP pelo financiamento desta pesquisa.

REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.** *NBR ISO14598: Engenharia de Software - Avaliação de Qualidade do Produto de Software.* Rio de Janeiro, 1999a. 18p.
- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.** *NBR ISO9126: Engenharia de Software - Qualidade do Produto.* Rio de Janeiro, 2003. 21p.
- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS.** *Subcomitê de software. Guia para utilização das normas sobre avaliação de qualidade de produto de software – ISO9126 e ISO14598.* Curitiba, 1999b. 65p.
- ANDRADE, E. L.** *Introdução à Pesquisa Operacional – Métodos e modelos para tomada de decisão.* 1 ed. LTC: Rio de Janeiro, 1998.
- AZUMA, M.** *Software products evaluation system: quality models, metrics and processes - International Standards and Japanese Practice.* Information and Software Technology, Vol.38, p.145-154, 1996.
- BHUTTA, K.S. & HUQ, F.** *Supplier selection problem: a comparison of the Total Cost Ownership and Analytic Hierarchy Process approaches.* Supply Chain Management, Vol.7, p.126-135, 2002.
- CAFÉ, L.; SANTOS, C. & MACEDO, F.** *Proposta de um método para escolha de um software de automação de bibliotecas.* Ciência da Informação, Vol.30,p.70-79, 2001.
- HLUPIC, V. et al.** *Evaluation Framework for Simulation Software.* International Journal of Advanced Manufacturing Technology, Vol.15, p.366-382, 1999.
- JUNG, H.** *Validating the external quality subcharacteristics of software products according to ISO/IEC 9126.* Computer Standards & Interfaces, Vol.29, p.653-661, 2007.
- PFLEEGER, S. L. & FITZGERALD, J. C.** *Software metrics tool kit: support for selection, collection and analysis.* Information and software technology, Vol.33, p.477-482, 1999.
- PIDD, M.** *Computer Simulation in management Science.* 5 ed. United Kingdom: John. Wiley & Sons, 2004. 328p.
- REDMOND, J. & AH-CHEN, A.** *Software Metrics – A User's perspective.* Journal of Systems Software. Vol.13,p.97-110, 1990.
- ROY, B.** *Multicriteria methodology for decision aiding.* Dordrecht: Kluwer, 1996.
- SAATY, T. L.** *Decision Making with the Analytic Hierarchy Process.* Internacional Journal of Services Sciences, Vol.1,n.1, p.83-98, 2008.
- SAATY, T. L.** *Decision Making With Dependence And Feedback: The Analytic Network Process.* RWS Publications: Pittsburgh, 1996.
- SAATY, T. L.** *The Analytic Network Process.* 1 ed. McGraw Hill: New York, 1980.
- TURBAN, E. et al.** *Decision Support and Business Intelligence Systems.* 8 ed. Pearson Prentice Hall, 2005. 750p.