

Avaliação Fuzzy da Qualidade do Gerenciamento de Projeto de Engenharia Web: Estudo de Caso do Produto de Software Omega Residential Condominium Plus

Alfredo Nazareno Pereira Boente
alfredoboente@faetec.rj.gov.br
FAETERJ Caxias

Renata Miranda Pires Boente
renatinha.boente@gmail.com
TRF-RJ

Kilmer Pereira Boente
kilmer_pereira@yahoo.com.br
FAETERJ Caxias

Bruno O'Neil da Silva
brunooneil79@gmail.com
FAETERJ Caxias

Resumo: Este artigo reporta a tomada de decisão em ambiente fuzzy a partir da avaliação da qualidade do gerenciamento de projeto de engenharia web tomando como base o produto de software Omega Residential Condominium Plus desenvolvido pela empresa Omega do Brasil Informática Ltda. A pesquisa teve caráter descritivo e foi realizada junto à equipe de seis especialistas em testes de software voltados para aplicações de engenharia web da empresa Omega do Brasil Informática Ltda. Para levantamento e análise dos critérios de qualidade de produtos de software voltados para web foram utilizados questionários estruturados, cujos valores foram tabulados em planilha eletrônica de dados do aplicativo Microsoft Excel, auxiliando assim no processamento dos dados e na criação de gráficos. A partir dos resultados obtidos, foram apresentados os graus de presença e importância e o índice de qualidade do gerenciamento de projeto de engenharia web. Por fim, procurou-se identificar pontos vulneráveis que pudessem resultar num gerenciamento com um índice de qualidade inferior daquele esperado pela equipe de testes de software da empresa Omega do Brasil Informática Ltda., a partir das normas definidas pela Engenharia Web.

Palavras Chave: Gestão de Projetos - Lógica Fuzzy - Tomada de Decisão - Engenharia Web -

1. INTRODUÇÃO

O gerenciamento de projetos, na década de 90, não representava grande atratividade nas empresas dos mais variados ramos de atividades. Entretanto, com o advento da nova economia, este tema tornou-se de extrema importância. De acordo com Boente (2003) o motivo desta mudança é registrado pelo fato de que mais do que nunca o trabalho está se realizando em etapas bem definidas que se cumprem e dão lugar a uma nova etapa ou um novo projeto.

Neste contexto, o gerenciamento de projetos, engloba desde a definição, o uso de recurso e técnicas, a administração e o controle de certo projeto, independentemente da área de aplicação a qual ele esteja direcionado.

Na área de desenvolvimento de produtos de *software*, em especial aqueles voltados para a web, isso não é diferente. Naturalmente, sempre que um novo produto de *software* é implementado, faz-se necessário avaliar se o seu uso está adequado aos padrões de qualidade requeridos pelos parâmetros básicos da Engenharia de *Software*, segundo a norma ISO/IEC 12.207. A melhor forma adequada de realizar tal avaliação está na base das opiniões dos especialistas envolvidos no projeto de engenharia web. A partir daí, um processo decisório é gerado com vistas à melhoria contínua do produto de *software*, propriamente dito.

Deve-se considerar que as opiniões advindas dos especialistas podem apresentar aspectos subjetivos, imprecisos e de interpretação ambígua. Neste viés, de acordo com Moré (2004), a maior parte da linguagem natural contém ambiguidades e multiplicidade de sentidos. Em particular, os adjetivos que utilizamos para caracterizar objetos ou situações não nos permitem clareza suficiente, sendo ambíguos, em termos de amplitude de significados, não podendo, portanto, ser aferido pela matemática crisp.

Portanto, é necessário o uso da teoria dos conjuntos *fuzzy*, pois ela permite avaliar eficientemente a satisfação desses especialistas, baseado em suas opiniões acerca do produto de *software* avaliado, por entender que esse tipo de avaliação envolve aspectos subjetivos e imprecisos, difíceis de serem medidos pela matemática tradicional e pela estatística.

Este artigo reporta a aplicação de um modelo *fuzzy* utilizado para avaliação de produtos de *software* baseado na norma ISO/IEC 12.207.

2. GERENCIAMENTO DE PROJETOS DE ENGENHARIA WEB

O gerenciamento de projetos, segundo Boente (2009), pode ser dividido em nove áreas do conhecimento. Essas áreas do conhecimento são conhecidas como: gestão da integração do projeto, gestão do escopo do projeto, gestão de prazo do projeto, gestão de custo do projeto, gestão da qualidade do projeto, gestão de equipe e pessoas do projeto, gestão da comunicação do projeto, gestão de riscos do projeto e gestão de aquisições do projeto.

A interação existente entre as atividades no gerenciamento de projetos das diversas áreas do conhecimento representa logicamente um elo bem consistente podendo até afetar outros processos do seu plano de projeto (LOPES, 2010).

Por padrão do PMBOK Guide é exigido um frequente balanceamento entre os objetivos definidos para certo projeto (PMI, 2012). Uma gestão de projetos eficaz requer um gerenciamento efetivo de todas essas possíveis interações, dentro do próprio projeto.

Para proporcionar um melhor entendimento da integração de atividades na gestão de projetos, e para enfatizar a importância da própria integração, é importante abordar o



gerenciamento de projetos em termos de seus processos e de suas interações a partir das nove áreas de conhecimento, conforme ilustrado na figura 1 (LÉLIS, 2009).

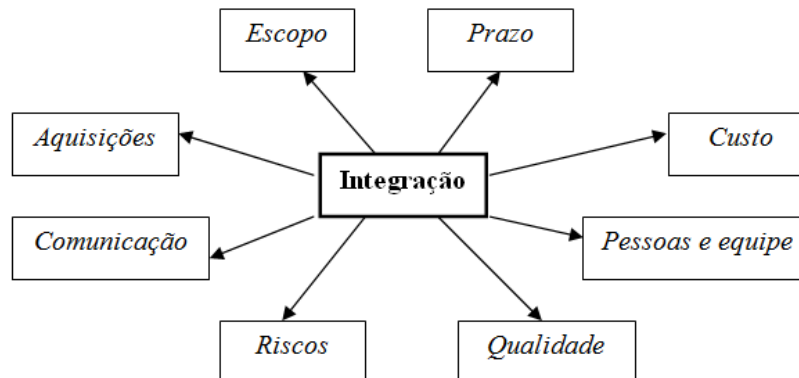


Figura 1: Áreas do conhecimento da gestão de projetos.

De acordo com Boente (2003), não se deve confundir projeto com processo, pois um projeto sempre termina, no entanto, um processo, não necessariamente, visto que um processo representa uma série de ações que tem como objetivo gerar certo resultado.

Os processos dos projetos são realizados por pessoas, e normalmente estão enquadrados como processos do gerenciamento de projetos e processos orientados ao produto, propriamente dito (MARTINS, 2011).

É fato considerar um significativo problema na gestão de projetos, em virtude de diversos aspectos de transição: as mudanças exigidas e requeridas em projetos.

Segundo Vargas (2009):

[...] a mudança em projetos requer reflexões sobre como chegar a um consenso sobre os objetivos e como se deve alcançá-los, como agregar os membros da equipe, como negociar prazos e orçamentos, e como reduzir os riscos e aumentar as chances de se alcançar o sucesso efetivo, para se obter um gerenciamento de projetos correto e pleno com a efetiva qualidade.

Essa questão acerca da importância do gerenciamento de projetos nos faz refletir que quanto maior a mudança requerida, mais inovações serão implementadas e, conseqüentemente, mais projetos surgirão.

3. ENGENHARIA WEB: QUALIDADE X TESTES

A engenharia web se apresentou com o objetivo de sanar as limitações mostradas pela engenharia de *software*, até então a única engenharia existente para a modelagem, construção e implementação de produtos de *software*, no que tange a modelagem e o desenvolvimento de aplicações voltadas para a web. De acordo com Lowe e Pressman (2009), a engenharia web propõe um arcabouço ágil, porém disciplinado, para a montagem de WebApps, aplicações voltadas para a web, de qualidade industrial.

Neste contexto, Pressman (2011) afirma que o arcabouço se refere a um conjunto de atividades que engloba um processo de WebE ou *Web Engineering*, que se aplica aos projetos de WebApp, tais como: *Comunicação*: representa as interações e colaborações com o cliente abrangendo o processo de engenharia de requisitos; *Planejamento*: descreve as ações de WebE, através de um plano incremental, que envolverão as tarefas técnicas, os riscos prováveis, os recursos que serão exigidos, produtos de trabalho a serem produzidos e o cronograma de trabalho; *Modelagem*: descreve os modelos que auxiliam o desenvolvedor web e o cliente a entender melhor os requisitos da WebApp; *Construção*: combina tecnologias

como XHTML, XML, JAVA dentre outros códigos próprios para WebApps, juntamente com os testes necessários identificar possíveis erros no código; *Implantação*: refere-se à entrega de uma WebApp ao cliente que terá que validar e aprova o produto de *software*.

Ágil refere-se diretamente a equipe de trabalho, *time*, uma equipe leve, que seja capaz de responder a mudanças durante o desenvolvimento de um projeto de *software* (PRESSMAN, 2011). Portanto, a engenharia web apresenta abordagens sistemáticas, disciplinadas e quantificáveis para o desenvolvimento de aplicações baseadas na web com foco na qualidade.

Os sistemas de engenharia web possuem alguns componentes essenciais divididos em forma de camadas, conforme ilustra a figura 2.



Figura 2: Camadas da engenharia web.

A engenharia web apresenta características próprias da engenharia de *software*, como a tecnologia em camadas, buscando enfatizar sempre a questão da qualidade. De acordo com Lowe e Pressman (2009), a camada de processo serve como base para o controle do gerenciamento de projetos de engenharia web, a partir do estabelecimento de pontos importantes, como os métodos técnicos, os marcos estabelecidos, as garantias da qualidade, as mudanças necessárias, dentre outros. A camada de métodos determina como deve ser feito para que certo produto de *software* seja construído. A camada de ferramentas disponibiliza o apoio automatizado e semi-automatizado para as camadas de processos e métodos. Um ambiente automatizado de apoio à engenharia de *software* é estabelecido quando existe a integração de ferramentas, de modo que as informações criadas possam ser usadas e compartilhadas por outras ferramentas.

Conforme afirma Gerti (2006), a engenharia web apresenta métodos que abrangem um conjunto de tarefas técnicas que permitam a um engenheiro web a entender e construir uma WebApp de alta qualidade, nos mesmos moldes da engenharia de *software*. Os métodos da WebE são os seguintes: comunicação, análise de requisitos, projeto, construção e testes.

Lowe e Pressman (2009) afirmam que o ideal para o engenheiro web é gastar bastante tempo no início para entender as necessidades do negócio e os objetos do produto, mesmo que os detalhes da WebApp sejam vagos.

Um arcabouço de processo WebE genérico lhe oferece a capacidade de obter um conhecimento do que é o problema (LOWE e PRESSMAN, 2011). Assim, o processo de engenharia Web está diretamente relacionado com o arcabouço que é aplicado interativamente à medida que cada componente é implementado à WebApp.

Segundo Lowe e Pressman (2009), as WebApps são entregues de modo incremental. Dessa forma as atividades de arcabouço ocorrerão rapidamente à medida que cada incremento de WebApp for desenvolvido e entregue. Para isto a comunicação é a atividade que estabelece o “destino” para um projeto de sistema de engenharia web.

O processo de modelagem de um sistema de engenharia web é muito importante e relevante ao projeto (Sommerville, 2011). Trata-se de uma atividade que cria uma ou mais

representações conceituais de algum aspecto da WebApp a ser construída. Uma representação conceitual abrange documentos escritos, esboços, diagramas esquemáticos, modelos gráficos, cenários escritos, protótipos em papel ou executáveis e códigos executáveis.

Segundo Lélis (2009), durante a modelagem de um projeto de engenharia Web ocorrem duas ações essenciais: a análise e o projeto de sistema. Pressman (2011) afirma que a modelagem da análise é responsável por examinar os requisitos dos interessados utilizando informações coletadas durante a atividade de comunicação como ponto de partida. A modelagem do projeto é uma atividade central da engenharia Web. Ele tem o objetivo de produzir um modelo ou uma representação que apresente firmeza, comodidade e deleite.

As ferramentas de modelagem utilizadas num projetos de sistemas de engenharia web servem de apoio ao seu desenvolvimento, propriamente dito (SOMMERVILLE, 2011). As tarefas do desenvolvimento do modelo do projeto contemplam as seguintes atividades: projetar a interface, projetar a estética da WebApp, projetar o esquema de navegação, projetar a arquitetura da WebApp, projetar o conteúdo e a estrutura de suporte, projetar os componentes funcionais, selecionar padrões de projetos adequados, projetar mecanismos de segurança e privacidade apropriados e revisar o projeto.

Lowe e Pressman (2009) afirmam que após a construção da WebApp a mesma deve ser obrigatoriamente testada. As tarefas a seguir perfazem o planejamento das ações de teste da WebApp: testar todos os componentes da WebApp, testar a navegação, testar a usabilidade, testar a segurança e o desempenho, e testar o incremento da WebApp para diferentes configurações. Percebe-se, portanto que o foco está voltado para a qualidade.

De acordo com Lowe e Pressman (2009), quando as atividades do arcabouço forem aplicadas para cada incremento de WebApp, uma coleção de atividades guarda-chuva ocorre em segundo plano com o objetivo de facilitar o gerenciamento de sistemas de engenharia web.

As chamadas atividades guarda-chuva são fundamentais para um projeto de sistema de web-E bem-sucedido (LOWE e PRESSMAN, 2009). Segundo os autores essas atividades são voltadas para o gerenciamento de projetos de sistemas de engenharia web e são caracterizadas pela gerência de mudanças, cujo objetivo é gerenciar o efeito da mudança à medida que cada incremento é desenvolvido, integrando ferramentas que auxiliam na gerência de todo o conteúdo de WebApp; pela garantia da qualidade, cujo objetivo é definir e conduzir as tarefas que ajudam a garantir a qualidade do incremento implementado em cada produto de trabalho; pela gerência de risco, que considera os riscos de projeto e técnicos à medida que um incremento é desenvolvido; e pela gerência de projetos, que visa acompanhar e monitorar o progresso à medida que um incremento é desenvolvido.

De acordo com Sommerville (2011) a gerência de mudanças, a garantia da qualidade, a gerência de riscos e a gerência de projetos em si, permitem que uma WebApp seja implementada com a efetiva qualidade.

4. TEORIA DOS CONJUNTOS FUZZY

Em 1965, o Professor Lotfi A. Zadeh formalizou o que, anos depois vinha a ser uma das maiores revoluções no setor matemático: a Lógica *Fuzzy*. Esta teoria trata dos conjuntos não totalmente verdadeiros nem tampouco dos totalmente falsos (COSENZA et. al., 2006). A lógica *fuzzy*, portanto, deve ser vista como uma teoria matemática formal para a representação de incertezas. Essa teoria é baseada no fato de que os conjuntos existentes no mundo real não possuem limites precisos (SIMÕES e SHAW, 2007).

Os conjuntos *fuzzy* podem ser vistos como uma generalização da noção de conjunto na qual a função de pertinência assume valores no intervalo [0,1] (FARIA, et. al., 2008). Através

da representação matemática de um conjunto ordenado de conceitos da linguagem natural através de conjuntos *fuzzy*, Prucol (2006) afirma que a discretização fuzzy $\{A_1, A_2, \dots, A_n\}$ do universo Ω tal que $\forall x \in \Omega, \exists A_i, \mu_{A_i}(x) \neq 0$, pode gerar a figura 3, que apresenta um exemplo de discretização *fuzzy* para o caso de uma variável dividida em cinco conjuntos *fuzzy*.

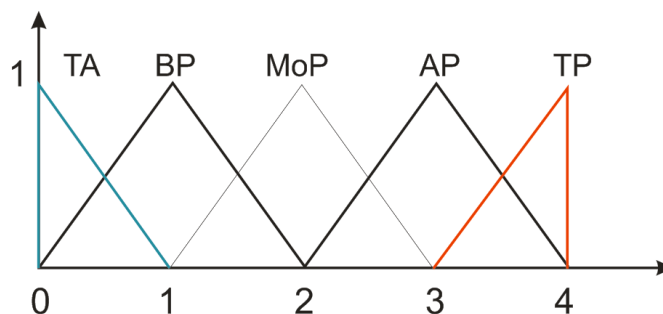


Figura 3: Representação da discretização *fuzzy*.

A representação da informação em diversos níveis de generalização, conforme afirma Boente (2009), é permitida por meio de diferentes discretizações do universo. Quanto maior o número de conjuntos *fuzzy*, maior será a precisão encontrada.

Segundo Belchior (1997) o processo de agregação permite obter um grau de consenso entre as informações disponíveis, calculando-se um valor final. Se estes dados forem extraídos de especialistas, se têm a taxa de aceitação ou rejeição entre eles, isto é, o grau pelo qual especialistas concordam em suas estimativas, tornando possível a elaboração de classificações das avaliações realizadas.

5. AVALIAÇÃO FUZZY DA QUALIDADE DE PROJETOS DE WEB-E

Zimmermann (1996) afirma que os conjuntos *fuzzy* representam os conceitos vagos, expressos na linguagem natural, conforme o contexto o qual são inseridos.

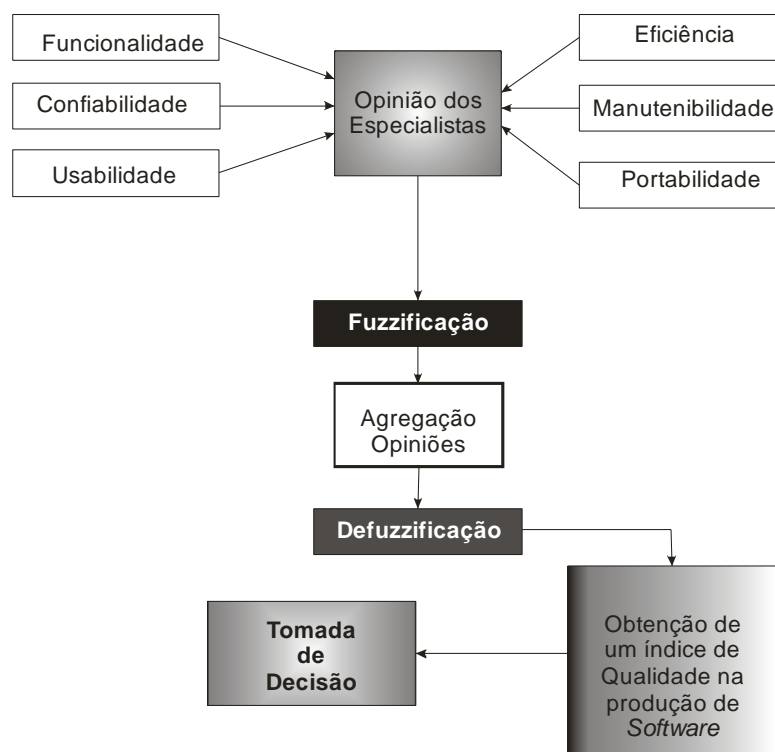


Figura 4: Descrição do modelo *fuzzy* para avaliação da qualidade do produto de *software*.

Neste viés, e partindo da hipótese que é possível mensurar o índice de qualidade do gerenciamento de projeto de Web-E, por meio da teoria dos conjuntos *fuzzy*, utilizou-se um modelo *fuzzy* que fosse capaz de realizar tal avaliação. Esse modelo *fuzzy*, conforme ilustra a figura 4 utilizou como base as opiniões dos seis especialistas envolvidos na produção do *software Omega Residential Condominium Plus*, desenvolvido pela empresa Omega do Brasil Informática, cujos dados foram coletados através de questionários estruturados, devidamente tabulados em planilha do Microsoft Excel, para que fosse possível julgar os aspectos de qualidade no gerenciamento de projeto de engenharia web. Para aplicar o modelo, foi necessário usar oito etapas conforme descrito a seguir.

Primeira etapa: Determinação das variáveis linguísticas do modelo. De acordo com Ross (2004) uma variável linguística é toda variável qualitativa não mensurável quantitativamente. Nesta etapa as variáveis linguísticas foram determinadas através de uma revisão bibliográfica que permitiu identificar seis construtos de qualidade de produtos de *software* - funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Esses termos foram transformados em 18 subcritérios de qualidade de produto de *software*.

Segunda etapa: Escolha dos termos linguísticos a serem utilizados para as medições a serem realizadas a partir dos dados coletados por meio dos questionários estruturados ministrados na etapa seguinte. Para medir o quão presente estão os critérios de qualidade na produção de *software*, considerou-se cinco termos linguísticos: totalmente ausente, baixa presença, moderadamente presente, altamente presente e totalmente presente. Também, para medir o quão importante resulta a presença dos critérios de qualidade na produção de *software*, foram escolhidos cinco outros termos linguísticos: sem importância, pouco importante, moderadamente importante, importante e muito importante.

Terceira etapa: Criação das funções de pertinências para os termos *fuzzy* apresentados na segunda etapa. Para representar as avaliações imprecisas e subjetivas nas opiniões dos especialistas foram escolhidos os conjuntos triangulares *fuzzy* pela capacidade que possuem de representar essa incerteza e para que seja computada facilmente pelo computador (TANAKA, 1997). O quadro 1, ilustra muito bem os conjuntos *fuzzy* correspondentes aos graus de presença de critérios de qualidade na produção de *software* foram: totalmente ausente (TA), baixa presença (BP), moderadamente presente (MoP), altamente presente (AP) e total presença (TP), assim como os conjuntos *fuzzy* correspondentes aos graus de importância dos critérios de qualidade na produção de *software* foram: muito importante (MI), importante (I), moderadamente importante (MoI), pouco importante (PI) e sem importância (SI).

Valor do termo <i>fuzzy</i>	Número triangular <i>fuzzy</i>	Grau de Presença	Grau de Importância
4	(3, 4, 4)	Total Presença (TP)	Muito Importante (MI)
3	(2, 3, 4)	Alta Presença (AP)	Importante (I)
2	(1, 2, 3)	Moderada Presença (MoP)	Moderadamente Importante (MoI)
1	(0, 1, 2)	Baixa Presença (BP)	Pouco Importante (PI)
0	(0, 0, 1)	Totalmente Ausente (TA)	Sem Importância (SI)

Quadro 1: Conjuntos triangulares *fuzzy* correspondentes aos conjuntos presença e importância.

Quarta etapa: Elaboração dos questionários estruturados. Nesta etapa foram elaborados três questionários com o objetivo de levantar o grau de presença e importância de critérios de qualidade na produção de *software* e determinar o peso de cada um dos

especialistas envolvidos na produção do produto de *software Omega Residential Condominium Plus*.

Quinta etapa: Aplicação dos questionários estruturados aos especialistas em produção de *software*. Em seguida os questionários foram aplicados aos seis especialistas envolvidos no processo de produção do *software Omega Residential Condominium Plus* da empresa Omega do Brasil Informática.

Sexta etapa: Para identificar os pesos dos seis especialistas em produção de *software*, coeficiente de importância, foram usadas diferentes escalas de medição, atribuídas de acordo com os itens de interesse levantados, *nível de experiência, prática, nível de conhecimento, grau de instrução, participação em projetos de produção de software, participação em congressos e currículo do especialista*. Cada item avaliado recebeu uma pontuação subjetiva entre 0 e 1, a partir da seguinte fórmula:

$$CI_{esp_i} = \frac{\sum_{i=1}^7 AvalEsp_i}{\sum_{i=1}^6 AvalEsp}$$

Sétima etapa: As respostas obtidas, *fuzzificação*, a partir dos questionários aplicados foram coletadas e tabuladas no Microsoft Excel.

Oitava etapa: Tratamento dos dados. Na fase de agregação das opiniões *fuzzy* foi utilizada a fórmula da média *fuzzy*. Neste caso considerou-se a avaliação *fuzzy* de cada um dos seis especialistas por critério $(Aval)_{crit}$, ponderada pelo coeficiente de importância de cada um deles (CI_{esp}) , obtendo-se assim os triângulos *fuzzy* agregados $(a,m,b)_{agreg}$.

$$(a, m, b)_{agreg_j} = \sum_{i=1}^6 \sum_{j=1}^{18} CI_{esp_i} * (Aval)_{crit_j}$$

Para se obter um valor preciso ou *crisp* representativo por cada conjunto *fuzzy*, *defuzzificação*, ou seja, um único valor numérico discreto que melhor representasse o valor *crisp* inferido das variáveis linguísticas utilizou-se a seguinte fórmula:

$$V_{crisp} = \frac{(a + 2m + b)_{agreg_j}}{4}$$

Onde a , m e b são os valores do triângulo *fuzzy* obtido a partir da agregação das opiniões dos seis especialistas. Cada valor *crisp* obtido foi dividido pelo valor máximo entre todos os valores *crisp*, processo de normalização, isto é, o valor normalizado calculado de acordo com a fórmula a seguir:

$$V_{norm} = \frac{V_{crisp}}{V_{max}}$$

Uma vez conhecidos os valores *crisp* foram calculadas as distâncias, *gap*, entre as demandas de qualidade do produto de *software*, graus de importância dos critérios de qualidade, e o atendimento a essas exigências pela empresa, graus de presença desses critérios de qualidade. Para isto, de acordo com Hsu e Chen (1996), tivemos que subtrair do valor *crisp* correspondente ao grau de presença o valor *crisp* referente ao grau de importância, para cada um dos dezoito critérios apresentados, a partir da seguinte fórmula:

$$DH_{crisp} = VP_{crisp} - VI_{crisp}$$



Os valores de distância, *gap*, são ilustrados na tabela 1. Uma distância negativa significa que a oferta de qualidade na produção de *software*, não cobre as necessidades ou expectativas.

Tabela 1: GAP entre as demandas de critérios de qualidade na produção de *software*

Critérios	Pres.	Imp.	Gap
Q01	2,94	2,83	0,11
Q02	2,91	2,84	0,07
Q03	2,83	2,78	0,05
Q04	2,88	2,69	0,19
Q05	2,28	2,98	-0,70
Q06	3,17	2,83	0,34
Q07	2,84	2,83	0,01
Q08	2,91	2,81	0,10
Q09	2,83	2,73	0,10
Q10	2,88	2,79	0,11
Q11	3,08	2,98	0,10
Q12	3,19	2,63	0,56
Q13	2,94	2,90	0,04
Q14	2,90	2,84	0,06
Q15	2,83	2,82	0,01
Q16	2,99	2,79	0,20
Q17	2,66	3,14	-0,48
Q18	3,16	2,83	0,33

Com base nos resultados, pode-se avaliar que existe um hiato e nesse caso é preciso aplicar recursos organizacionais com o objetivo de melhorar os critérios de qualidade Q05, ora identificado como *tempo de resposta*, e Q17, ora identificado como *tempo de carregamento*.

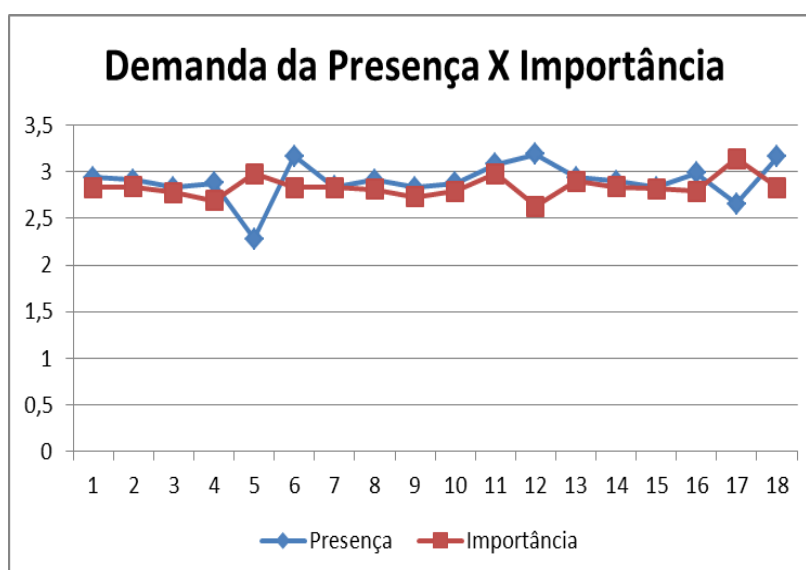


Figura 5: Demanda do Grau de presença X grau de importância.

As distâncias positivas apresentadas significam que a oferta supera a demanda e nesse caso não é preciso aplicar recursos. Pode-se melhor visualizar a representação dessas distâncias através da figura 5.

Nota-se que tanto no critério Q05 quanto no critério Q17, o valor da presença desses itens está menor que o valor da importância, perfazendo assim critérios de qualidade de *software* ainda não atendidos.

6. METODOLOGIA

A presente pesquisa é uma investigação descritiva, pois ao avaliar as variáveis que serviram de parâmetro para estimar as opiniões dos seis especialistas envolvidos no processo de produção de *software* quanto aos aspectos da qualidade, teve como principal objetivo tornar algo inteligível descrevendo-lhe os motivos. Ela também é caracterizada como explicativa, por ter criado uma teoria aceitável a respeito de um fato ou fenômeno, o índice de qualidade.

Além de ter um cunho bibliográfico, a pesquisa é caracterizada também como pesquisa de campo, pois levou em conta seu caráter empírico, onde as fontes coletadas ocorreram por meio de questionários estruturados visando assim, levantar o grau de qualidade na produção de *software*, a partir da opinião dos seis especialistas respondentes.

7. RESULTADOS E CONCLUSÕES

Considerando os valores já normalizados, foi possível determinar um índice de qualidade do produto de *software Omega Residential Condominium Plus*, a partir do uso da seguinte fórmula, média ponderada *fuzzy*:

$$I_{Qual} = \frac{\sum_{i=1}^{18} (G_{IMP_i} * G_{PRES_i})}{\sum_{i=1}^{18} G_{IMP_i}} = 0,94$$

Portanto, existe uma alta possibilidade, 0,94, de que os critérios de qualidade estejam presentes no produto de *software Omega Residential Condominium Plus*, desenvolvido pela empresa Omega do Brasil Informática tomando em conta os graus de importância (G_{imp}) e presença (G_{pres}) desses critérios. Pelos resultados obtidos pode-se observar que as maiores distâncias aparecem nos critérios Q05 - *Tempo de Resposta* e Q17 - *Tempo de Carregamento*. São estas as variáveis que merecem uma atenção especial, no processo de tomada de decisão para aumentar o índice de qualidade efetivamente.

Constata-se que independentemente do índice de qualidade do produto de *software Omega Residential Condominium Plus* ter sido considerado elevado, 0,94, ainda falta um caminho por percorrer, 0,03, para chegar ao “ótimo” que subscreva a chamada qualidade total. Neste sentido haverá necessidade de aplicar recursos nos itens Q05 e Q17, por apresentarem apresentarem insuficiência de recursos ou falta de informação, o que requer um tratamento minucioso em busca de melhorar o grau de presença destes critérios de qualidade na produção do *software Omega Residential Condominium Plus*.

Este artigo apresentou o aspecto da tomada de decisão a partir da avaliação *fuzzy*, do produto de *software Omega Residential Condominium Plus*, visando atingir a melhoria contínua da qualidade. Isso foi possível a partir de um conjunto de 18 subcritérios de qualidade necessários à produção de *software*, identificados a partir dos modelos de qualidade



ISO/IEC 9126 e ISO/IEC 12.119, abordados nesta pesquisa, e ilustrados na figura 4, através do modelo *fuzzy* apresentado.

A abordagem *fuzzy* permitiu tratar de forma matematicamente sólida, medidas subjetivas a incertezas, obtidas a partir da opinião dos seis especialistas da empresa Omega do Brasil Informática, respondentes desta pesquisa.

Foi obtido um índice de qualidade do produto de *software Omega Residential Condominium Plus*, 0,94, considerado muito alto. Apesar disso, foram identificadas lacunas não tão satisfatórias, o que permitiu aprimorar o processo de tomada de decisão acerca desse produto de *software* em busca da melhoria contínua da qualidade aplicada na produção de *software*, da empresa Omega do Brasil Informática, para projetos de engenharia web.

8. REFERÊNCIAS

- BELCHIOR, A.D.** Um Modelo Fuzzy para Avaliação da Qualidade de Software. Tese de Doutorado, Engenharia de Sistemas e Computação, COPPE, Universidade Federal do Rio de Janeiro, UFRJ, RJ, 1997.
- BOENTE, A.** Gerenciamento & Controle de Projetos. Rio de Janeiro: Axcel Books, 2003.
- BOENTE, A.N.P.** Um Modelo Fuzzy para Avaliação da Qualidade de Produtos de Software e da Satisfação dos Gerentes de Projetos numa Fundação Pública Estadual. Dissertação de Mestrado, Administração e Desenvolvimento Empresarial, Universidade Estácio de Sá, RJ, 2009.
- BOENTE, K.P.** Uma Metodologia de Gestão de Rede Wireless para o Núcleo de Informática do Campus Madureira da Universidade Estácio de Sá. Monografia, Faculdades Integradas de Jacarepaguá, Rio de Janeiro: 2009.
- COSENZA, H.J.S.R.** et al. Aplicação de Um Modelo de Hierarquização Como Instrumento para Tomada de Decisão: Caso de uma Multinacional. In: XXVI Encontro Nacional de Engenharia de Produção, ENEGEP, 2006, Fortaleza, 2006.
- FARIA, M.N.** et al. Um Sistema de Avaliação em EAD baseado em Lógica Fuzzy. Revista Eletrônica Horizonte Científico. Minas Gerais, edição 2008, dez/2008.
- GERTI, K.** Web Engineering: The Discipline of Systematic Development of Web Application. Boston: John Wiley & Sons, 2006.
- HIS-MEI-HSU.; CHEN-TUNG-CHEN.** Aggregation of fuzzy opinions under group decision making. Fuzzy Sets and Systems. vol 29 pp. 279-285,. 1996.
- LÉLIS, J.C.** Garantia de sucesso em gestão de projetos. Rio de Janeiro; Brasport, 2009.
- LOPES, A.J.** Experiência em Gestão de Projetos: Diário de bordo. Rio de Janeiro: Brasport, 2010.
- MARTINS, J.C.C.** Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML. Rio de Janeiro: Brasport, 2011.
- MORÉ, J.D.** Aplicação da lógica Fuzzy na avaliação da confiabilidade humana nos ensaios não destrutivos por ultra-som. Tese de Doutorado submetida ao programa de pós-graduação de Engenharia Metalúrgica e dos Materiais da Universidade Federal do Rio de Janeiro, COPPE, 2004.
- PMI.** PMI - Project Management Institute. A Guide to the Project Management Body of Knowledge, 2012.
- PRESSMAN, R.S.** Engenharia de software. 10 ed. São Paulo: McGraw-Hill, 2011.
- PRUCOLE, E.S.** Avaliação de Combinações de Classificadores Fuzzy. Dissertação de Mestrado, Engenharia Civil, COOPE/URFJ, Universidade Federal do Rio de Janeiro, COOPE, RJ, 2006.
- ROSS, T. J.** Fuzzy Logic with Engineering Applications. John Wiley & Sons, Ltd, Chincester, England, 2004.
- SIMÕES, M.G. e SHAW, I.S.** Controle e Modelagem Fuzzy. 2. ed. Revisada e Completa. São Paulo: Blucher: FAPESP, 2008.
- SOMMERVILLE, I.** Engenharia de Software. 9 ed. São Paulo: Pearson, 2011.
- TANAKA, K.** An Introduction to Fuzzy Logic for Practical Applications. Springer-Verlag, New York, 1997.



IX SEGeT 2012

**SIMPÓSIO DE EXCELÊNCIA EM
GESTÃO E TECNOLOGIA**

Tema: Gestão, Inovação e Tecnologia para a Sustentabilidade

VARGAS, R.V. Gerenciamento de Projetos. 7 edição. Rio de Janeiro: Brasport, 2009.

ZIMMERMANN, H. J. Fuzzy Set Theory and Its Applications. 3rd ed., Kluwer Academic Publishers, Boston, MA, 1996.