

# **Avaliação Fuzzy da qualidade de produtos de software desenvolvidos sob a filosofia de fábrica de software: Estudo de caso de uma softhouse situada em Vila Velha no Espírito Santo**

## **The Fuzzy evaluation of the quality of software product carried out under the philosophy of software factory: The case study of a softhouse in Vila Velha, Espírito Santo**

**Alfredo Nazareno Pereira Boente<sup>1</sup>**

**Francisco Antonio de Moraes Accioli Doria<sup>2</sup>**

**Carlos Alberto Nunes Cosenza<sup>3</sup>**

### **Resumo:**

A demanda por produtos de *software* tem aumentado significativamente em todo o mundo, trazendo novos desafios à engenharia de *software*. Essa realidade não é diferente no Brasil. Na cidade de Vila Velha, no Estado do Espírito Santo, certa *softhouse* desenvolve produtos de *software* sob a filosofia de fábrica de *software*, numa linha enxuta, em perspectiva *just-in-time*, com a efetiva qualidade. A filosofia de fábrica de *software* é um modelo de produção e uma alternativa para desenvolvimento de produtos de *software* em diversas escalas. A linha de produção de produtos de *software* que adota o conceito de fábrica de manufatura deve seguir as normas e parâmetros da engenharia de *software*, com vistas a ISO/IEC 12.207 e a ISO/IEC 9.126.1. Todos os parâmetros baseados pelas ISOs adotadas perfazem uma total preocupação com os aspectos da qualidade de *software*. Este trabalho tem por objetivo fazer uma avaliação *fuzzy* da qualidade de produtos de *software* desenvolvidos sob a filosofia de fábrica de *software*, a partir das opiniões de quinze especialistas em engenharia de *software* e sistemas.

**Palavras-chave:** Fábrica de *software*; engenharia de *software*; qualidade de *software*.

### **Abstract:**

Demand for software products has increased significantly all over the world, bringing new challenges to software engineering. The situation is not different in Brazil. In Vila Velha city, in the state of Espírito Santo, a software house develops products under the software factory philosophy, in a lean line, just-in-time perspective with effective quality. The software factory philosophy is a model of production and an alternative to developing software products in various scales. The production line of software products that adopts the concept of manufacturing factory must follow the standards and software engineering parameters, based on ISO/IEC 12.207 and ISO/IEC 9.126.1. All parameters were based on the ISOs show total concern with the aspects of software quality. This aims at making a fuzzy evaluation of the quality of software products developed by the software factory philosophy, through the opinions of fifteen specialists in software and systems engineering.

**Keywords:** Software factory; software engineering; software quality.

---

<sup>1</sup> Doutor em Engenharia de Produção (COPPE/UFRJ), Professor do Curso de Engenharia da Computação da Universidade Veiga de Almeida. E-mail: [alfredo.boente@uva.br](mailto:alfredo.boente@uva.br).

<sup>2</sup> Doutor em Física pelo Centro de Pesquisas Físicas, Professor do Curso de Engenharia de Produção da COPPE/UFRJ. E-mail: [fadoria63@gmail.com](mailto:fadoria63@gmail.com).

<sup>3</sup> Doutor em Engenharia de Produção pela Universidade Federal de Santa Catarina, Professor do Curso de Engenharia de Produção da COPPE/UFRJ. E-mail: [cosenzacoppe@gmail.com](mailto:cosenzacoppe@gmail.com).

## **Introdução**

Nos dias de hoje, a alta demanda por produtos de *software* personalizados e exclusivos é eminente em todo o mundo. Por esse motivo, desenvolver produtos de *software* com a efetiva qualidade é fundamental para o sucesso de qualquer *software*. Seguir as normas, os padrões e os parâmetros das ISO/IEC 12.207 e ISO/IEC 9.126-1 é, portanto, uma maneira de garantir o desenvolvimento de um produto de *software* eficiente e eficaz, com a devida qualidade.

Neste viés, o presente trabalho tem como objetivo a avaliação da qualidade de *software* de certa *software*, aqui nomeada como empresa “*RENALF*”, através da lógica *fuzzy*, por meio da coleta de dados advindos da opinião de profissionais especialistas no desenvolvimento de produtos de *software* sob a filosofia de fábrica de *software*, considerando que se trata de uma avaliação subjetiva, sujeita a imprecisões e a incertezas.

## **1. Referencial teórico**

### **1.1. Fábrica de *Software***

Fábrica de *software* é uma filosofia ou modelo de produção de *software* composto por um conjunto de recursos (humanos, materiais, tecnológicos etc.), processos e metodologias estruturados de forma semelhante às indústrias que adotam linha de produção, utilizando as melhores práticas para o processo de desenvolvimento, testes e manutenções de produtos de *software* (FERNANDES e TEIXEIRA, 2012).

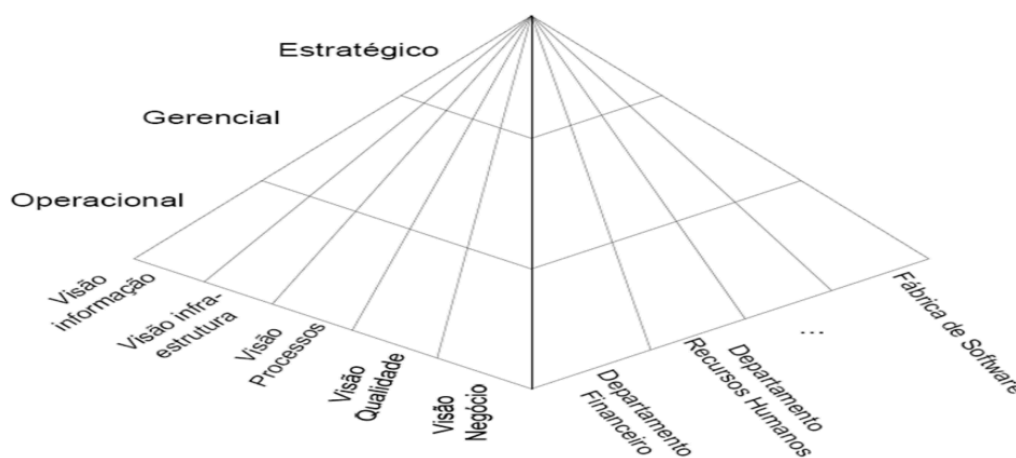
Em sua operação, faz uso de indicadores de qualidade e produtividade em cada etapa do ciclo de vida de desenvolvimento de *software*, buscando maximizar a reutilização de componentes, reduzindo, portanto, os custos de produção (TENÓRIO e VALLE, 2010).

Na filosofia de fábrica de *software*, utiliza-se o conceito de células de competência, conforme ilustra a figura 1, segundo o qual cada profissional especialista é alocado na célula em que tem a efetiva competência, numa visão de produção em massa, ou seja, de grande volume de produtos por demanda (MAXIMILIANO, 2000).



**Figura 1** - Células de Competência de uma Fábrica de *Software*.  
**Fonte:** Adaptado de Tenório e Valle, 2010.

Henry Ford foi o primeiro a aplicar esse conceito de produção em massa, quando montou uma fábrica para automóveis, na qual o produto era padronizado quanto ao material, à mão de obra, ao *design* e ao mínimo custo possível. Na mesma visão, Fernandes e Teixeira (2012) afirmam que a fábrica de *software* busca adaptar de forma positiva o processo de linha de montagem implantado por Ford, perfazendo uma integração com as estratégias organizacionais, conforme ilustra a figura 2.



**Figura 2** - Visões da Fábrica de *Software* Integrada.  
**Fonte:** Borsoi, 2008.

A fábrica de *software* é um ambiente de desenvolvimento de *software* constituído por processos que envolvem o ciclo de vida de *software*, categorizados em níveis hierárquicos corporativos: estratégico, gerencial e operacional (BORSOI, 2008).

## 1.2. Qualidade de *Software*

De acordo com Boente (2013), antes mesmo de falar em qualidade de *software*, é necessário entender o conceito de qualidade. Qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas de certo indivíduo (BARTIÉ, 2002).

De acordo Duarte e Falbo (2008), “um *software* ou sistema de informação tem qualidade quando está adequado à empresa, ao cliente e/ou usuário e atende a padrões de qualidade predefinidos”. Assim, a qualidade de *software* consiste na conformidade existente aos requisitos funcionais de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo o produto de *software* profissionalmente desenvolvido (PRESSMAN, 2012).

Nesse viés, Guerra e Colombo (2009) afirmam que a qualidade do produto de *software* é o conjunto das propriedades que determinam sua habilidade em satisfazer às necessidades para as quais ele foi criado ou mesmo superá-las.

Para obter a qualidade desejada de produtos de *software*, fazem-se necessários modelos que viabilizem a avaliação da qualidade desses produtos (BELCHIOR, 1997). As organizações internacionais de normalização ISO/IEC vêm trabalhando conjuntamente em um modelo que permita avaliar a qualidade dos produtos de *software*, mais especificamente as ISO/IEC 12.207 e ISO/IEC 9.126-1 (RINCON, 2009).

### 1.2.1. ISO/IEC 12.207

A ISO/IEC 12.207 é a norma ISO/IEC que define processo de desenvolvimento de *software*. De acordo com Pressman (2012), a norma internacional ISO/IEC 12.207, conforme ilustrado na figura 3, tem como objetivo estabelecer uma estrutura comum para os processos de ciclo de vida de produtos de *software* visando auxiliar a organização a compreender todos os componentes presentes na aquisição e no fornecimento de *software*, e perfazendo, assim, produtos de *software* eficientes e eficazes.

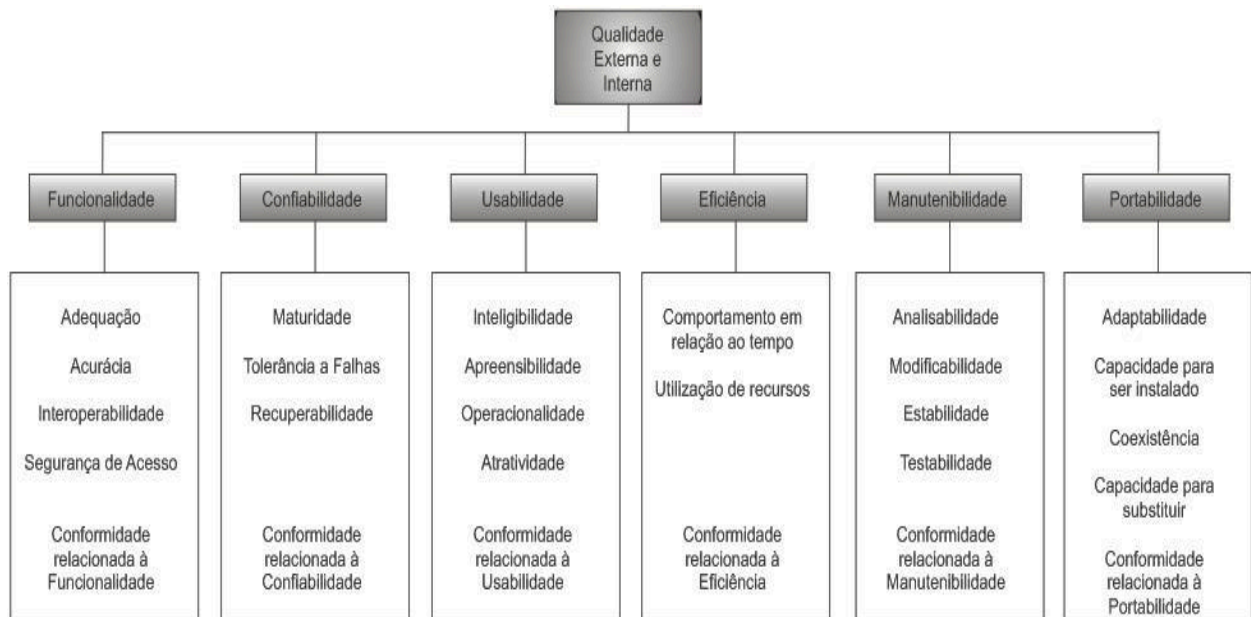
Os processos de ciclo de vida, ditos processos de adaptação, controlados pela ISO/IEC 12.207, dividem-se em processos fundamentais, processos de apoio e processos organizacionais.

<b>Processos Fundamentais</b>		<b>Processos de Apoio</b>		<b>Processo de Adaptação</b>
Aquisição		Documentação		
Fornecimento		Gerência de Configuração		
Desenvolvimento	Operação	Garantia da Qualidade		
		Verificação		
		Validação		
		Revisão Conjunta		
	Manutenção	Auditoria		
		Usabilidade		
		Gerência de Resolução de Problemas		
		Gerência de mudanças		
Avaliação do Produto				
<b>Processos Organizacionais</b>				
Gerência	Engenharia de Domínio	Melhoria		
Gestão de Ativos	Infra-estrutura			
Gestão Programa de Reuso	Recursos Humanos			

**Figura 3** - ISO/IEC 12.207.  
**Fonte:** Devmedia, 2007.

### 1.2.2. ISO/IEC 9.126-1

A qualidade de *software* é apresentada a partir da norma ISO/IEC 9.126-1, conforme ilustrado na figura 4, classificando-se, portanto, como externa e interna (TISURI e KARAM, 2013).



**Figura 4** - ISO/IEC 9.126-1.  
**Fonte:** Boente, 2015.

Essas características fazem com que os especialistas em desenvolvimento de produtos de *software* fiquem ou não satisfeitos (BARTIÉ, 2002).

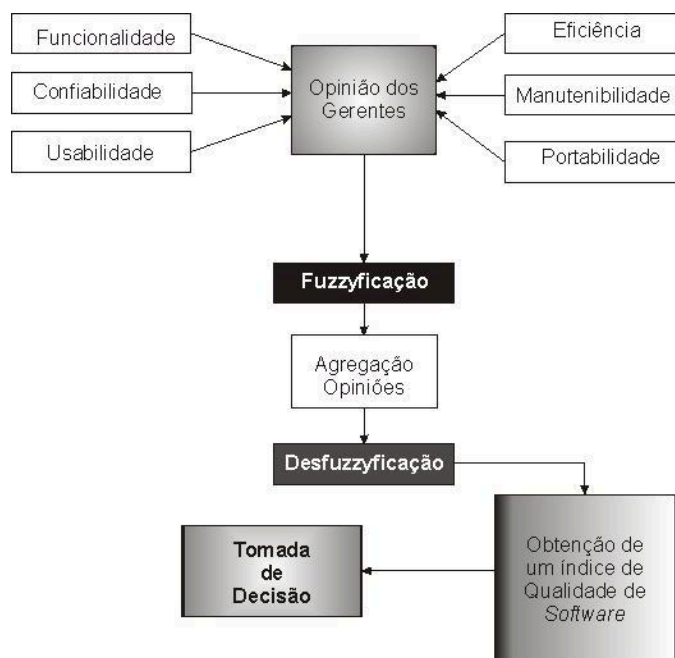
### 1.3. Lógica Fuzzy

De acordo com Boente (2013), o professor Lotfi Asker Zadeh formalizou uma das maiores revoluções no setor matemático: a Lógica *Fuzzy*. Essa teoria trata dos conjuntos não totalmente verdadeiros nem tampouco totalmente falsos. Em outras palavras, a lógica *fuzzy* deve ser vista como uma teoria matemática formal para a representação de incertezas e imprecisões.

Pedricz e Gomide (2007) afirmam que a lógica *fuzzy* pode ser definida como uma lógica que suporta os modos de raciocínio aproximado ao nível de exatos. Baseia-se na teoria dos conjuntos nebulosos e apresenta o princípio da dualidade, segundo o qual dois eventos opostos podem coexistir.

### 1.4. Modelo Fuzzy para Avaliação da Qualidade de Software

Considerando que avaliação da qualidade de produtos de *software* a partir da opinião dos profissionais especialistas apresenta aspectos de representação imprecisos, compostos, em sua maioria, por conceitos subjetivos, vagos e de avaliação não trivial, propõe-se o uso da lógica *fuzzy*.



**Figura 5** - Modelo *Fuzzy* para Avaliação da Qualidade de Software.

**Fonte:** Boente, 2013.

Partindo da hipótese de que é possível avaliar a qualidade de produtos de *software* através da lógica *fuzzy*, surge a ideia de usar o modelo *fuzzy* apresentado, conforme ilustração da figura 5. Esse modelo utiliza as opiniões dos profissionais especialistas no desenvolvimento de produtos de *software* sob a filosofia de fábrica de *software*, como dados coletados para a presente pesquisa.

Para avaliar a qualidade de produtos de *software*, consideraram-se como constructos os aspectos de funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade do *software*, expressos na norma de qualidade de produtos de *software* ISO/IEC 9.126-1.

## **2. Estudo de Caso da *Soft*house “RENALF”**

Na cidade de Vila Velha, no Estado do Espírito Santo, uma *soft*house denominada empresa “RENALF”, com mais de 15 anos de mercado, resolveu fazer uma auditoria acerca da qualidade dos produtos de *software* por ela desenvolvidos sob a filosofia de fábrica de *software*. Para tanto, foram entrevistados quinze (15) respondentes com experiência no desenvolvimento de produtos de *software*.

Para aplicação do modelo *fuzzy* foram necessárias seis etapas, conforme descritas a seguir.

### **2.1. Etapa 1: Determinação das Variáveis Linguísticas do Modelo**

Nesta etapa, as variáveis linguísticas foram determinadas a partir dos seis (06) constructos de qualidade de *software* (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade).

### **2.2. Etapa 2: Escolha dos Termos Linguísticos**

Para verificar o grau de presença dos critérios de qualidade de produto de *software*, foram escolhidos cinco (05) termos linguísticos: totalmente ausente, baixa presença, moderadamente presente, altamente presente e totalmente presente, conforme ilustrado no quadro 1.

Para verificar o grau de importância da presença dos critérios de qualidade de produto de *software*, também foram escolhidos cinco (05) termos linguísticos: sem importância, pouco importante, moderadamente importante, importante e muito importante, conforme ilustrado no quadro 2.

Escala	Equivalência	Descrição
0	Totalmente ausente	Indica total ausência do critério de qualidade avaliado
1	Baixa presença	Indica um baixo grau de presença do critério de qualidade avaliado
2	Moderadamente presente	Indica um grau de presença moderada do critério.
3	Altamente presente	Indica um alto grau de presença do critério, mas não de forma plena.
4	Total Presença	Indica que não há dúvidas de que o critério está totalmente presente.

**Quadro 1** - Termos linguísticos para o grau de presença dos critérios de qualidade de *software*.  
**Fonte:** Boente, 2013.

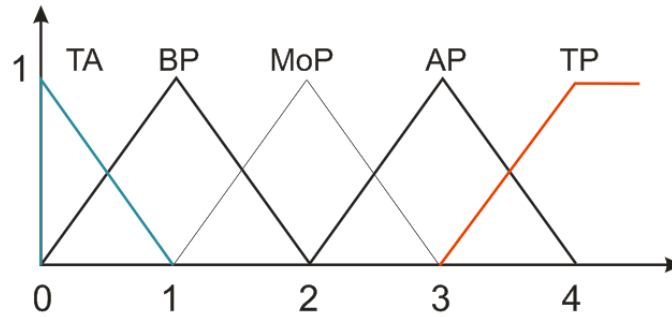
Escala	Equivalência	Descrição
0	Sem importância	Indica que o critério que está sendo apresentado não tem nenhuma importância.
1	Pouco importante	Indica que o critério que está sendo apresentado tem pouca importância.
2	Moderadamente importante	Indica que o critério que está sendo apresentado tem importância em algumas circunstâncias, mas nem sempre.
3	Importante	Indica que o critério que está sendo apresentado é importante.
4	Muito importante	Indica que não há dúvidas com que o critério que está sendo apresentado é imprescindível.

**Quadro 2** - Termos linguísticos para o grau de importância dos critérios de qualidade de *software*.  
**Fonte:** Boente, 2013.

### 2.3. Etapa 3: Criação das Funções de Pertinência dos Termos Linguísticos Apresentados

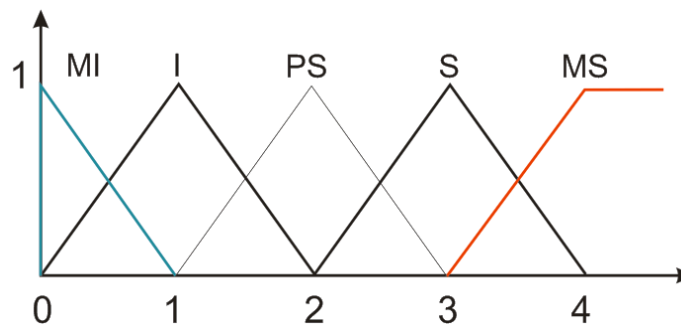
Para representar as avaliações imprecisas e subjetivas a partir das opiniões dos respondentes foram escolhidos os conjuntos *fuzzy* triangulares, por sua capacidade de representar essa incerteza.

O conjunto *fuzzy* correspondente aos graus de presença de critérios de qualidade de produto de *software* foi: totalmente ausente (TA), baixa presença (BP), moderadamente presente (MoP), altamente presente (AP) e total presença (TP), conforme ilustrado na figura 6.



**Figura 6** - Conjunto *fuzzy* dos termos referentes ao grau de presença dos critérios de qualidade de *software*.  
**Fonte:** Boente, 2013.

O conjunto *fuzzy* correspondentes aos graus de importância dos critérios de qualidade de produto de *software* foi: muito importante (MI), importante (I), moderadamente importante (MoI), pouco importante (PI) e sem importância (SI), conforme ilustrado na figura 7.



**Figura 7** - Conjunto *fuzzy* dos termos referentes ao grau de importância dos critérios de qualidade de *software*.  
**Fonte:** Boente, 2013.

#### 2.4. Etapa 4: Elaboração e Aplicação dos Questionários Estruturados à Amostra

Foram elaborados dois (02) questionários estruturados acerca do processo de avaliação da qualidade de produtos de *software*, um para coletar dados sobre a presença de critérios de qualidade de *software* e o outro para coletar dados sobre a importância de critérios de qualidade de *software*, cada um com 41 perguntas baseadas nos seis (06) constructos identificados na ISO/IEC 9.126-1 (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade).

Em momento oportuno, ambos os questionários estruturados foram aplicados a uma amostra de quinze (15) profissionais da empresa “RENALF”, especialistas no desenvolvimento de produtos de *software* sob a filosofia de fábrica de *software*.

## 2.5. Etapa 5: Coleta, Tabulação e Tratamento de Dados

As respostas obtidas através do processo de fuzzificação, gerada a partir dos questionários estruturados aplicados aos profissionais especialistas, respondentes desta pesquisa, foram coletadas e tabuladas, mediante a ferramenta Ms-Excel, conforme ilustram os quadros 3 e 4, a seguir:

Item	Profissionais Especialistas Respondentes															Fuzzificação			Defuzzificação	Processo de
	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	G11	G12	G13	G14	G15	Nº Triangular	Valor Crisp	Normalização		
I01	3	3	3	2	4	2	3	2	2	2	2	2	2	3	3	1,53	2,53	3,47	2,52	0,83
I02	3	2	3	2	4	2	1	2	2	2	2	2	1	3	3	1,27	2,27	3,20	2,25	0,74
I03	3	2	3	2	3	2	1	2	2	2	2	2	1	3	3	1,20	2,20	3,20	2,20	0,72
I04	2	3	3	3	3	3	2	2	2	2	2	2	2	2	3	1,40	2,40	3,40	2,40	0,79
I05	4	3	4	2	4	2	1	2	2	2	2	2	3	4	3	1,67	2,67	3,40	2,60	0,85
I06	3	3	3	2	4	2	2	2	2	2	2	2	2	3	3	1,47	2,47	3,40	2,45	0,80
I07	3	2	4	2	4	3	2	2	2	3	2	2	2	3	4	1,67	2,67	3,47	2,62	0,86
I08	3	3	3	1	4	3	4	2	2	2	2	2	2	3	3	1,60	2,60	3,47	2,57	0,84
I09	3	4	3	1	4	2	4	2	2	2	2	2	1	3	3	1,53	2,53	3,33	2,48	0,81
I10	3	3	3	2	3	3	2	2	2	2	2	2	2	3	3	1,47	2,47	3,47	2,47	0,81
I11	4	1	3	1	3	3	0	2	2	2	2	2	3	4	3	1,40	2,33	3,20	2,32	0,76
I12	4	2	2	0	3	1	0	2	3	1	1	3	2	2	3	0,93	1,87	2,80	1,87	0,61
I13	4	2	4	2	4	2	1	2	2	2	2	2	2	4	1	1,40	2,40	3,13	2,33	0,77
I14	3	2	3	0	3	4	1	2	2	2	2	2	2	3	3	1,33	2,27	3,20	2,27	0,74
I15	3	2	3	1	3	2	0	2	4	2	3	2	2	3	2	1,29	2,22	3,16	2,22	0,73
I16	3	3	3	1	3	2	1	2	3	3	2	3	2	3	3	1,49	2,49	3,49	2,49	0,82
I17	4	3	3	1	3	2	0	2	3	4	3	3	2	3	4	1,73	2,67	3,47	2,63	0,86
I18	1	2	1	2	2	1	3	2	3	2	2	1	2	2	1	0,76	1,76	2,76	1,76	0,58
I19	1	2	1	2	2	1	1	2	2	2	2	1	2	2	1	0,58	1,58	2,58	1,58	0,52
I20	4	3	3	2	3	3	2	2	4	2	3	2	2	4	2	1,68	2,68	3,55	2,65	0,87
I21	4	2	3	2	3	2	2	2	2	2	2	2	1	4	2	1,33	2,33	3,20	2,30	0,75
I22	3	1	1	2	3	2	2	2	2	2	2	2	2	3	2	1,07	2,07	3,07	2,07	0,68
I23	4	3	3	2	3	2	1	2	1	2	2	1	2	4	1	1,17	2,17	3,03	2,13	0,70
I24	4	2	4	2	4	3	2	2	2	2	2	2	2	4	2	1,60	2,60	3,33	2,53	0,83
I25	3	3	4	2	4	1	2	2	2	2	2	2	2	3	4	1,53	2,53	3,33	2,48	0,81
I26	3	4	4	2	4	1	2	2	2	2	2	2	2	3	4	1,60	2,60	3,33	2,53	0,83
I27	3	2	3	2	1	3	1	2	2	2	2	2	2	3	3	1,20	2,20	3,20	2,20	0,72
I28	3	3	3	2	4	2	1	2	3	2	2	4	2	4	4	1,67	2,67	3,60	2,65	0,87
I29	3	2	3	2	4	1	2	2	3	0	3	3	0	4	3	1,47	2,33	3,20	2,33	0,77
I30	3	2	3	2	4	3	2	2	4	3	4	3	3	4	3	1,97	2,97	3,77	2,92	0,96
I31	3	3	3	2	4	3	2	2	2	2	2	2	4	3	3	1,67	2,67	3,53	2,63	0,86
I32	4	0	4	2	4	3	2	2	2	2	2	2	1	3	4	1,53	2,47	3,20	2,42	0,79
I33	1	1	1	2	1	3	2	2	2	1	3	1	1	3	1	0,53	1,53	2,53	1,53	0,50
I34	3	3	4	2	4	3	3	2	3	4	2	4	3	3	4	2,13	3,13	3,80	3,05	1,00
I35	2	3	3	2	4	3	2	2	4	2	1	2	3	3	2	1,43	2,43	3,37	2,42	0,79
I36	3	2	3	2	3	3	2	2	3	4	4	4	1	3	3	1,80	2,80	3,60	2,75	0,90
I37	2	3	3	2	2	3	2	2	3	2	2	2	2	2	2	1,22	2,22	3,22	2,22	0,73
I38	2	2	3	2	4	0	0	2	2	2	2	2	2	2	2	1,07	1,93	2,87	1,95	0,64
I39	3	2	3	2	2	0	2	2	2	2	2	2	1	3	3	1,13	2,07	3,07	2,08	0,68
I40	3	3	4	2	3	3	2	2	2	1	2	3	1	3	3	1,44	2,44	3,44	2,44	0,80
I41	3	3	3	2	3	3	2	2	3	4	3	3	2	3	3	1,80	2,80	3,73	2,78	0,91

Quadro 3 - Grau de presença dos critérios de qualidade de *software*.

Fonte: Elaboração própria.

A agregação das opiniões *fuzzy* foi obtida através da fórmula da média *fuzzy*. No caso, considerou-se a avaliação *fuzzy* de cada profissional especialista desenvolvedor do produto de *software* sob a filosofia de fábrica de *software* respondente por critério  $(Aval)_{crit}$ , ponderada pelo

coeficiente de importância de cada um deles ( $CI_{resp}$ ), obtendo-se, portanto, os triângulos *fuzzy* agregados  $(a,m,b)_{agreg}$  que nos quadros 3 e 4, aparecem representados na coluna “Fuzzificação N° Triangular”, a partir da seguinte fórmula:

$$(a) \quad (a,m,b)_{agreg_j} = \sum_{i=1}^{15} \sum_{j=1}^{41} CI_{resp_i} * (Aval)_{crit_j}$$

Item	Profissionais Especialistas Respondentes															Fuzzificação			Valor	Valor
	G01	G02	G03	G04	G05	G06	G07	G08	G09	G10	G11	G12	G13	G14	G15	N° Triangular	Crisp	Normal		
Q01	3	3	4	2	4	3	2	2	2	2	2	2	3	3	3	1,60	2,60	3,47	2,57	0,81
Q02	3	3	4	2	3	3	2	2	2	2	2	2	3	3	3	1,60	2,60	3,53	2,58	0,81
Q03	3	3	2	2	3	3	2	2	2	3	2	2	3	3	2	1,47	2,47	3,47	2,47	0,77
Q04	4	3	4	2	4	3	2	2	2	3	2	2	3	3	3	1,80	2,80	3,60	2,75	0,86
Q05	4	3	4	2	4	4	1	2	2	3	4	2	3	3	3	1,93	2,93	3,60	2,85	0,90
Q06	2	3	2	2	4	4	2	2	2	3	2	4	3	4	3	1,80	2,80	3,53	2,73	0,86
Q07	4	4	2	2	4	4	2	2	2	2	2	4	3	4	2	1,87	2,87	3,47	2,77	0,87
Q08	2	4	4	2	4	3	2	2	2	2	2	3	4	3	3	1,73	2,73	3,47	2,67	0,84
Q09	2	4	4	2	4	4	2	2	2	2	3	2	3	4	3	1,87	2,87	3,53	2,78	0,87
Q10	4	2	4	2	4	3	2	2	2	2	3	2	3	3	3	1,73	2,73	3,53	2,68	0,84
Q11	3	3	4	2	4	3	4	2	2	2	2	2	3	3	3	1,80	2,80	3,60	2,75	0,86
Q12	3	3	3	2	2	3	2	2	2	2	2	2	3	3	3	1,47	2,47	3,47	2,47	0,77
Q13	3	4	4	2	4	3	2	2	2	3	2	2	3	3	4	1,87	2,87	3,60	2,80	0,88
Q14	3	4	3	2	4	3	2	2	2	3	2	2	3	4	4	1,87	2,87	3,60	2,80	0,88
Q15	4	4	4	2	4	3	2	2	2	3	2	2	4	4	4	2,07	3,07	3,60	2,95	0,93
Q16	2	4	4	2	4	3	2	2	2	2	3	4	4	3	3	1,93	2,93	3,60	2,85	0,90
Q17	3	4	4	2	4	3	2	2	2	2	2	4	4	4	3	2,00	3,00	3,60	2,90	0,91
Q18	3	3	3	2	3	3	2	2	2	4	2	2	4	3	3	1,73	2,73	3,60	2,70	0,85
Q19	3	3	3	2	3	3	2	2	2	2	2	2	4	3	3	1,60	2,60	3,53	2,58	0,81
Q20	4	3	4	2	4	3	2	2	2	2	2	3	3	3	3	1,73	2,73	3,53	2,68	0,84
Q21	4	3	4	2	3	3	2	2	2	2	2	2	4	3	4	1,80	2,80	3,53	2,73	0,86
Q22	2	3	4	2	4	3	2	2	2	2	4	2	4	2	4	1,80	2,80	3,47	2,72	0,85
Q23	4	3	4	2	4	3	2	2	2	3	4	2	4	3	4	2,07	3,07	3,67	2,97	0,93
Q24	4	4	4	2	4	3	2	2	2	3	4	2	4	3	3	2,07	3,07	3,67	2,97	0,93
Q25	4	3	4	2	4	3	2	2	2	2	2	2	4	3	3	1,80	2,80	3,53	2,73	0,86
Q26	3	4	4	2	4	3	2	2	2	2	2	4	4	2	3	1,87	2,87	3,53	2,78	0,87
Q27	3	4	4	2	4	3	2	2	2	2	2	3	4	3	3	1,87	2,87	3,60	2,80	0,88
Q28	3	4	4	2	4	3	2	2	2	4	2	4	4	3	2	2,00	3,00	3,60	2,90	0,91
Q29	2	2	3	2	3	2	3	2	2	2	2	2	2	3	2	1,27	2,27	3,27	2,27	0,71
Q30	2	3	3	2	4	4	2	2	2	2	2	2	4	3	1	1,53	2,53	3,33	2,48	0,78
Q31	2	3	3	2	4	3	3	2	2	2	2	2	2	3	1	1,40	2,40	3,33	2,38	0,75
Q32	0	0	0	2	0	3	3	2	2	2	2	2	0	1	0	0,67	1,27	2,27	1,37	0,43
Q33	0	0	0	2	0	3	3	2	2	2	2	4	4	1	0	1,00	1,67	2,53	1,72	0,54
Q34	4	4	4	2	4	3	2	2	2	2	2	2	3	3	3	1,80	2,80	3,53	2,73	0,86
Q35	4	4	4	2	4	3	2	2	2	2	2	2	3	4	3	1,87	2,87	3,53	2,78	0,87
Q36	3	4	4	2	4	4	2	2	2	2	2	2	4	4	3	1,93	2,93	3,53	2,83	0,89
Q37	3	4	4	2	4	4	2	2	4	4	2	4	4	4	3	2,33	3,33	3,73	3,18	1,00
Q38	3	3	4	2	4	4	2	2	4	3	2	3	4	4	3	2,13	3,13	3,73	3,03	0,95
Q39	3	3	3	2	3	4	4	2	2	2	2	2	4	3	3	1,80	2,80	3,60	2,75	0,86
Q40	3	3	4	2	3	3	4	2	2	2	2	2	4	3	3	1,80	2,80	3,60	2,75	0,86
Q41	3	3	4	2	3	3	1	2	3	4	2	2	4	3	3	1,80	2,80	3,60	2,75	0,86
																			Σ=	34,54

Quadro 4 - Grau de importância dos critérios de qualidade de *software*.

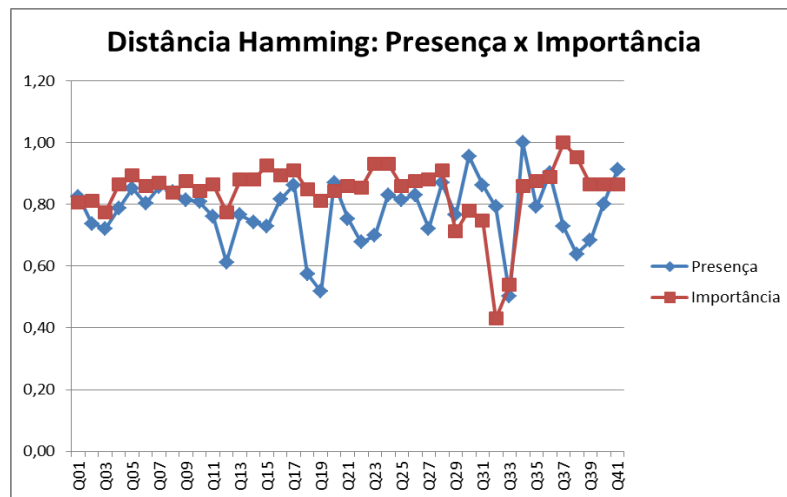
Fonte: Elaboração própria.

A partir da fórmula (b) foi possível calcular o valor crisp, um número real proveniente ao processo de defuzzificação, e em seguida, calcular o valor normal, ou seja, um número real com propriedades de acordo com a teoria dos conjuntos *fuzzy*, através da fórmula (c).

$$(b) \quad V_{crisp} = \frac{(a + 2m + b)_{agreg_j}}{4} \qquad (c) \quad V_{norm} = \frac{V_{crisp}}{V_{max}}$$

## 2.6. Etapa 6: Análise de Dados e Tomada de Decisão

A partir dos graus de presença e importância dos critérios de qualidade de produtos de *software*, foi calculada a Distância *Hamming*, descrita pela fórmula (d), conforme ilustra da figura 8.



**Figura 8** - Distância existente entre os Graus de Presença e Importância dos Critérios de Qualidade de *Software*.  
**Fonte:** Elaboração própria.

$$(d) \quad Dist_{crisp} = V_{crisp} P - V_{crisp} I$$

Analisando as distâncias, consegue-se observar que os critérios Q19 (Acessibilidade: Existe cópia de segurança da especificação fora do local de trabalho?), Q32 (Consistência Interna: A especificação não apresenta conflitos lógicos ou temporais entre requisitos do produto que dependem do tempo?), Q37 (Precisão: Os requisitos do produto estão descritos de forma a poderem ser validados?) e Q38 (Necessidade dos Requisitos: Os aspectos descritos na especificação são considerados imprescindíveis?), são os critérios em que se identificaram as maiores distâncias, sendo, por isso, considerados critérios problemáticos, que interferem diretamente na qualidade do produto de *software* desenvolvido e necessitam de um tratamento prioritário a fim de melhorar o índice de qualidade aferido.

Através da fórmula (e), buscou-se calcular o índice de qualidade de produtos de *software* desenvolvidos sob a filosofia de fábrica de *software*, tendo-se obtido 0,78, considerado como um índice de qualidade bom.

$$(e) \quad I_{Qual} = \frac{\sum_{i=1}^{41} (G_{IMP_i} \times G_{PRES_i})}{\sum_{i=1}^{41} G_{IMP_i}} = 0,78$$

Dessa forma, a partir da análise das variáveis disponíveis para o processo de avaliação da qualidade de produto de *software* desenvolvido sob a filosofia de fábrica de *software*, destacaram-se prioritariamente: Q19 - Acessibilidade, Q32 - Consistência Interna, Q37 - Precisão e Q38 - Necessidade dos Requisitos, que, por sua vez, merecem uma atenção especial, por parte dos gerentes de desenvolvimento de produtos de *software* da *softhouse* “RENALF”, visando aumentar efetivamente a qualidade dos produtos de *software* desenvolvidos por esta empresa.

Ainda que o índice de qualidade apontado nesta pesquisa – 0,78 – tenha sido considerado bom, é importante ressaltar que a equipe de desenvolvimento de produtos de *software* sob a filosofia de fábrica de *software*, está cometendo algumas falhas que precisam ser melhoradas ou superadas, a fim de aumentar o índice de qualidade encontrado.

### 3. Considerações Finais

Neste trabalho, utilizou-se um modelo *fuzzy* para avaliação da qualidade de produtos de *software* desenvolvidos sob a filosofia de fábrica de *software*, a partir de um conjunto de 41 critérios de qualidade de software, baseados pela ISO/IEC 12.207 e ISO/IEC 9.126-1.

A abordagem *fuzzy* permitiu tratar, de forma matematicamente sólida, medidas subjetivas sujeitas a incertezas e imprecisões, obtidas a partir da opinião de 55 profissionais especialistas respondentes desta pesquisa.

Foi obtido um índice de qualidade de produto de *software* (0,78), considerado “bom”. Apesar disso, foram identificadas lacunas de insatisfação de qualidade dos produtos de *software* desenvolvidos sob a filosofia de fábrica de *software* pela *softhouse*, o que permitiu aprimorar o processo de tomada de decisão da empresa, em busca da melhoria dessa qualidade.

## Referências:

- BARTIÉ, A. **Garantia da Qualidade de Software**. Rio de Janeiro: Campus. 2002.
- BELCHIOR, A. D. **Um Modelo Fuzzy para Avaliação da Qualidade de Software**. Tese de Doutorado do Departamento de Engenharia de Sistemas e Computação da COPPE/UFRJ, da Universidade Federal do Rio de Janeiro. 1997.
- BOENTE, A. N. P. **Proposição de um Modelo Fuzzy para Tomada de Decisão acerca da Avaliação da Qualidade do Produto de Software AVA Moodle Utilizado no Curso de Pós-Graduação em Tecnologias Educacionais do IST-Rio e da Satisfação de seus Usuários**. Tese de Doutorado do Departamento de Engenharia de Produção da COPPE/UFRJ, da Universidade Federal do Rio de Janeiro. 2013.
- BOENTE, R. M. P. **Avaliação Fuzzy da Aprendizagem Organizacional**. Revista EDU.TEC, v. 1, n. 1, e. 2, Revista Científica Digital da FAETEC. 2015.
- BORSOI, B. **Arquitetura de processo aplicada na integração de fábricas de software**. Tese de Doutorado em Engenharia Elétrica do Departamento de Engenharia Elétrica da USP, Universidade de São Paulo. 2008.
- DEVMEDIA. **Qualidade e Software: entenda os principais conceitos sobre testes e inspeção de software**. Edição Especial. Edição especial. Paraná: Engenharia de Software Magazine. 2007.
- DUARTE, K.C.; FALBO, R.A. **Uma Ontologia de Qualidade de Software**. In X Encontro de Estudantes de Informática de Tocantins, 2008, Palmas, Anais do X Encontro de Estudantes de Informática de Tocantins. Palmas: Centro Universitário Luterano de Palmas. 2008.
- FERNANDES, A.A.; TEIXEIRA, D.S. **Fábrica de Software: implantação e gestão de operações**. São Paulo: Atlas. 2012.
- GUERRA, A.C.; COLOMBO, R.M.T. **Tecnologia da Informação: Qualidade de Produto de Software**. Porto Alegre: PBQP Editora. 2009.
- MAXIMIANO, A.C.A. **Introdução à Administração**. 5 ed. São Paulo: Atlas. 2000.
- RINCON, A.M. **Qualidade de Software**. In XI Encontro de Estudantes de Informática de Tocantins, 2009, Palmas, Anais de XI Encontro de Estudantes de Informática de Tocantins. Palmas: Centro Universitário Luterano de Palmas. 2009.
- PEDRICZ, W.; GOMIDE, F. **Fuzzy System Engineering**. New York: Wiley. 2007.
- PRESSMAN, R.S. **Engenharia de Software**. 11th ed. São Paulo: Bookman. 2012.
- TENÓRIO, F.G.; VALLE, R. **Fábrica de Software**. São Paulo: FGV Editora. 2010.
- TSUI, F.; KARAM, O. **Fundamentos de Engenharia de Software**. 2 ed. Rio de Janeiro: LTC Editora. 2013.