



PROPOSTA DE APLICAÇÃO DO SOFTWARE QUALITY FUNCTION DEPLOYMENT EM AMBIENTE COMPUTACIONAL

FURSTENAU, L.B¹; KIPPER L. M.²; FROZZA, R.³ HOPPE, D.⁴

PALAVRAS CHAVE: *Quality Function Deployment, Software, Ambiente Computacional, SQFD*

RESUMO

A qualidade de um *software* precisa estar alinhada com o grau de satisfação do cliente. Para que isto ocorra, é fundamental que os requisitos do cliente sejam devidamente identificados e atendidos a fim de suprir suas reais necessidades. O *QFD* (*Quality Function Deployment*) também conhecido como a *casa da qualidade* foi originalmente criado para melhorar a qualidade de produtos a partir do atendimento da voz do cliente. O *QFD* tem sido utilizado por pesquisadores tanto para desenvolver quanto para melhorar a qualidade de softwares. Esta pesquisa tem como objetivo propor a utilização do *QFD* como ferramenta de melhoria do ambiente computacional (*Ideaware*) desenvolvido no projeto de pesquisa "Ambiente Computacional para Desenvolver a Criatividade e Inovação nas Organizações". Para isto, realizou-se uma revisão da literatura a fim de identificar de que forma o *QFD* tem sido aplicado em softwares. Por fim, uma proposta da aplicação do *QFD* para melhorias do ambiente computacional é apresentada ao final deste trabalho.

PROPOSAL FOR THE APPLICATION OF SOFTWARE QUALITY FUNCTION DEPLOYMENT IN COMPUTATIONAL ENVIRONMENT

KEYWORDS: *Quality Function Deployment, Software, Computational Environment, SQFD*

ABSTRACT

The software quality needs to be aligned with the degree of customer satisfaction. Therefore, it is critical that customer requirements are properly identified and pleased in order to satisfy their real needs. QFD (Quality Function Deployment) also known as House of Quality was originally created to improve the quality of products from the Voice of Customer. QFD has been used by researchers both to develop and to improve the quality of software. This research aims to propose the use of QFD as a tool to improve the computational environment (*Ideaware*) developed in the research project "Computational Environment to Develop Creativity and Innovation in Organizations". To do so, a literature review was carried out to identify how QFD has been applied in software. Finally, a contribution of the application of QFD to improvements of the computational environment is presented at the end of this work.

¹ Graduado em Engenharia de Produção e aluno do Programa de Pós-Graduação em Sistemas e Processos Industriais - UNISC

² Doutora em Engenharia de Produção e professora do Programa de Pós-Graduação em Sistemas e Processos Industriais - UNISC

³ Doutora em Computação e professora do Programa de Pós-Graduação em Sistemas e Processos Industriais - UNISC

⁴ Mestre em Engenharia de Produção e professor do curso de Engenharia de Produção - UNISC

1 INTRODUÇÃO

O sucesso de uma organização e sua sobrevivência no mercado depende de características fundamentais como a criatividade e a inovação (MARTINS; TERBLANCHE, 2003). Para West (2002) e Van De Ven (2017), a criatividade é o processo da geração de novas ideias, enquanto que a inovação é reconhecida como sendo a introdução destas ideias, criando novas formas de realizar algo ou implementar melhorias naquilo que já existe. Desta forma, Mclean (2005) aponta que sem inovação, a criatividade tem pouco valor, assim como não é possível obter inovação sem a geração de ideias criativas. Logo, a inovação sem a criatividade é como um motor sem combustível. Neste sentido, a criatividade e a inovação devem ser desafiadas nas organizações, pois, conforme Stojcic, Hashi e Orlic (2018); Gumusluoglu e Ilsev (2009); Woodman, Sawyer e Griffin (1993), representam ser a chave para atingir altos níveis de desempenho e competitividade.

A geração e implementação de ideias advindas de sugestões dos colaboradores se tornou uma fonte de vantagem competitiva para impulsionar a criatividade e a inovação nas organizações (WEST, 2002; ZHOU e SHALLEY, 2003; ANDERSON e DE DREU, 2004; ANDERSON, POTOČNIK e ZHOU, 2014). Neste contexto, foi criado o projeto intitulado: “ambiente computacional para promover a criatividade e a inovação nas organizações” (apoio à Indústria criativa do Rio Grande do Sul – edital de chamada pública DCIT Nº 03/ 2015), desenvolvido no laboratório interativo de criatividade (LIC), vinculado ao TecnoUnisc, e financiado pela Secretaria de Desenvolvimento Econômico, Ciência e Tecnologia do Estado do Rio Grande do Sul (SDECT/RS) e pela Universidade de Santa Cruz do Sul - UNISC.

Este projeto envolve professores, estudantes e bolsistas das áreas da computação, engenharias e do Programa de Pós-graduação em Sistemas e Processos Industriais da UNISC. O *software* Ideiaware tem como objetivo desafiar as capacidades dos colaboradores das organizações parceiras do parque científico e tecnológico para que discutam ideias e trabalhem de forma colaborativa na composição e aprimoramento de ideias de produtos ou serviços, visando melhorias nos processos organizacionais. O presente trabalho visa contribuir para o projeto a partir da proposta de utilização do *QFD* aplicado ao *software* (ambiente computacional) desenvolvido.

O *QFD* é uma ferramenta da qualidade que auxilia na criação do desenvolvimento de produtos diferenciados, os quais dependem do entendimento de necessidades não atendidas e, muitas vezes, não compreendida advinda da voz do cliente, do inglês *voice of customer (VoC)* (COOPER e EDGETT, 2008). Chan e Wu (2002); Kumaravadeivel, Natarajan e Noorul Haq (2012) reforçam que a voz do cliente deve ser cuidadosamente analisada e desdobrada para a voz da engenharia. Carnevalli e Miguel (2008); Bottani (2009); Vinodh e Kumar Chinthra (2011) reforçam que o *QFD* permite a tradução de tais informações, auxiliando na criação e melhoria de produtos e serviços. Embora o *QFD* seja utilizado principalmente para produção de bens duráveis e não duráveis, este pode ser aplicado ao desenvolvimento de *softwares* (SCALVENZI e CAUCHICK MIGUEL, 2016). Autores como Haag, Raja e Schkade (1996); Karlsson (1997); Şen e Baraçlı (2010); Sun e Liu (2010), entre outros, apontam em seus estudos benefícios significativos da utilização do *QFD* em desenvolvimento e melhoria de *softwares*, sendo denominado também por *software quality function deployment (SQFD)*. Sener e Karsak (2012) afirmam que o *SQFD* é uma adaptação do *QFD* tradicional utilizado em produtos tangíveis para desenvolvimento de *softwares*.

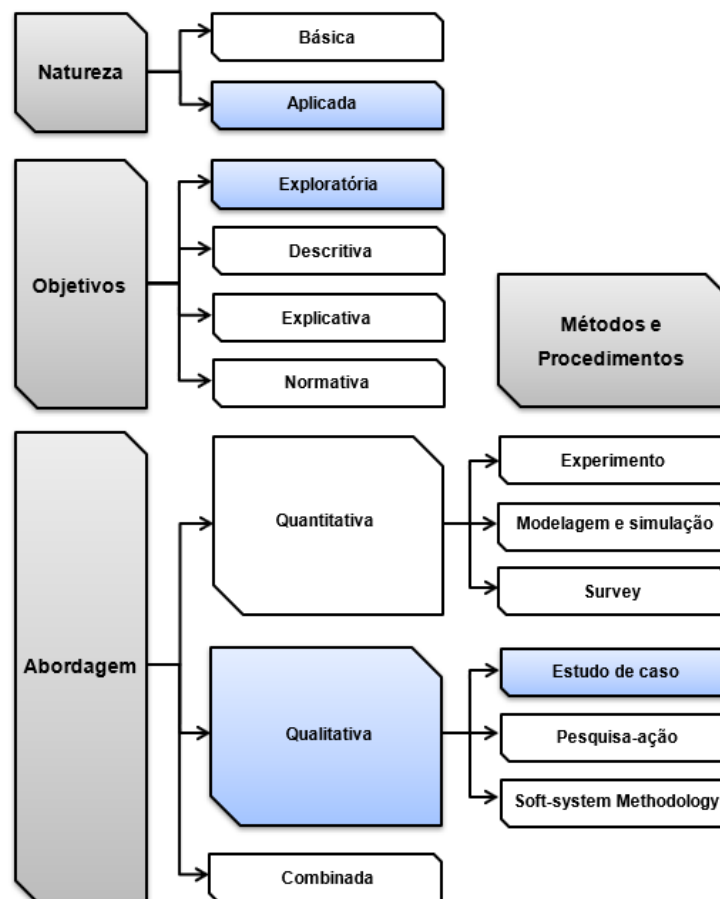
Desta forma, esta pesquisa teve como objetivo propor a utilização do *QFD* como ferramenta de melhoria do ambiente computacional (*Ideiaware*) desenvolvido no projeto de pesquisa Ambiente computacional para desenvolver a criatividade e inovação nas organizações”. Este trabalho será dividido em 4 seções, sendo que a

primeira já foi apresentada. A Seção 2 compreende os materiais e métodos utilizados. Na Seção 3 serão apresentadas as análises e leitura dos documentos da RL, com uma visão da utilização do *QFD* aplicado em *softwares*. A Seção 4 apresenta a proposta de aplicação do *QFD* para o ambiente computacional, bem como uma discussão sobre a implantação *QFD*. A conclusão e referências finalizam o artigo.

2 METODOLOGIA

Conforme Fiorese (2003); Gomes e Araújo (2005) para que seja possível alcançar um determinado objetivo, é fundamental que exista uma metodologia claramente definida e de eficácia comprovada, sendo um conjunto de processos que servirá de norte para o pesquisador. Além disso, Gerhardt et al. (2009); Turrioni e Mello (2012) apontam a importância em identificar a pesquisa quanto a sua abordagem, sua natureza, seus objetivos e seus procedimentos, bem como selecionar a modalidade de pesquisa adequada ao objeto de pesquisa. A Figura 1 apresenta a classificação da pesquisa científica para a engenharia de produção, bem como as características preenchidas em azul.

Figura 1. Classificação da pesquisa científica em engenharia de produção.



Fonte: Turrioni e Mello (2012).

Quanto à natureza esta pesquisa será aplicada, pois gera conhecimentos aplicáveis e possíveis de serem praticados a fim de solucionar problemas específicos, envolvendo verdades e interesses locais (GERHARDT et al., 2009). O objetivo é exploratório, pois preocupa-se com esta pesquisa proporcionar maior familiaridade com o problema levantado (RAUPP e BEUREN, 2006). A abordagem é qualitativa, pois buscou-se uma análise baseada no paradigma interpretativo (GERHARDT et al., 2009) a partir da leitura dos artigos. O método e procedimento adotado nesta pesquisa será o estudo de caso, pois conforme Miguel (2007) “o estudo de caso é um estudo de natureza empírica que investiga um determinado fenômeno, geralmente contemporâneo, dentro de um contexto real de vida”.

2.1 REVISÃO NARRATIVA DA LITERATURA

A Revisão Narrativa da Literatura (RNL) é utilizada para descrever o estado da arte de um assunto específico, sob o ponto de vista teórico ou contextual, ou seja, foi realizada inicialmente a análise da literatura a partir da interpretação e análise crítica pessoal do pesquisador (CORDEIRO, 2007; BOTELHO, DE ALMEIDA CUNHA e MACEDO, 2011). Além disso, Ramos Vosgerau e Romanowski (2014) afirmam que a RNL é adequada para fundamentação teórica. No entanto, Zupic e Čater (2015); Tranfield, Denyer e Smart (2003) apontam que a RNL possui viés por parte do pesquisador devido à falta de rigor na seleção de documentos. Neste sentido para diminuir esta carência de rigor, foi desenvolvido o referencial teórico a partir de buscas somente de artigos e revisões em bancos de dados como Scielo, Capes, Scopus, IEEEExplore, Google Academic, Web of Science e Science Direct. Além disso, para a escolha dos artigos, fora definido somente artigos com predominância na língua inglesa, de revistas com alto impacto acadêmico e com alto índice de citações, a fim de aumentar a confiabilidade das informações. Os termos escolhidos para busca foram: “*quality function deployment*”, “*house of quality*”, “*QFD*”, “*software quality*” e “*software*”. Nesta pesquisa a RNL foi utilizada para criar o referencial teórico de qualidade de *software* e *QFD*. Além disso, foi definido que o subperíodo seria de 1990 a 2018, ano em que iniciaram as documentações na literatura da base Scopus sobre *QFD* aplicado a softwares através dos estudos de Thackeray e Treack (1990).

3 REFERENCIAL TEÓRICO

Para atingir o objetivo deste trabalho e os resultados apresentados pelo mesmo, é necessário desenvolver um referencial teórico, o qual servirá de base para a formulação do estudo e, por conseguinte, dos resultados encontrados. Para tanto, nesta etapa serão descritos conceitos que foram pesquisados através da RNL em artigos científicos, relacionados à qualidade de *software*, *QFD* e *QFD* aplicado à *softwares*.

3.1. QUALIDADE DE SOFTWARE

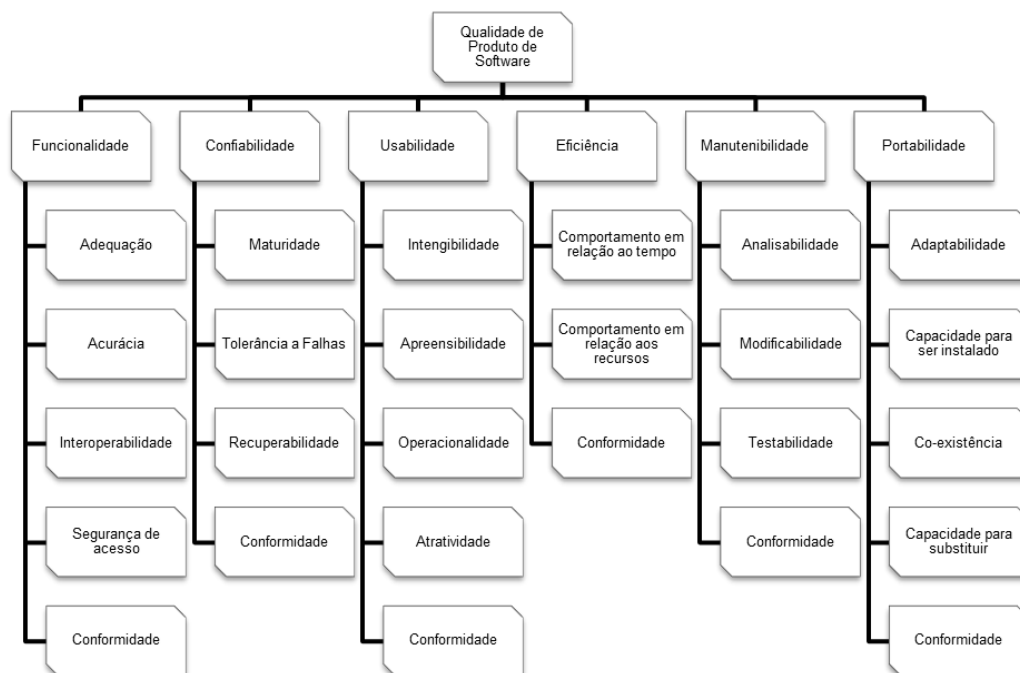
A NBR ISO/IEC 9126-1 define qualidade como a “totalidade de características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas”. A norma categoriza os atributos de qualidade de *software* em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) as quais são, por sua vez, subdivididas em subcaracterísticas (Figura 2). As subcaracterísticas podem ser medidas por meio de métricas internas e externas. Tais características são abordadas em estudos como forma de medição da qualidade de *softwares* por Kannabiran e Sankaran (2011);

Khosravi *et al.* (2017). Além disso, Eriksson e McFadden (1993); Basili e Selby (1984) apontam que a qualidade de *software* está relacionada com o ponto de vista da pessoa que está analisando o produto, ou seja, existem diferenças de análises entre usuários, desenvolvedores, administradores e organizações, existindo assim diversas formas de medir a qualidade de um *software*.

A qualidade de um *software* deve ser gerenciada e medida a partir do estágio de desenvolvimento do produto e durante etapas de operação (KIM; KIM, 2018). Para Basili e Rombach (1988) existem várias maneiras de mensurar a qualidade de um *software* e que desta forma é possível analisar custos, eficácia, correções, confiabilidade, manutenibilidade, eficiência, facilidade de uso, etc. Por outro lado, Jorgensen (1999) ressalta que a qualidade do *software* precisa estar alinhada com o grau de satisfação do cliente. Para que isto ocorra, é fundamental que os requisitos do cliente sejam devidamente identificados e atendidos a fim de suprir suas necessidades (KONDO, 2001). Para Nagappan, Murphy e Basili (2008) os indicadores de qualidade de *software* são essenciais aos engenheiros para garantir a confiabilidade do sistema, estimando e priorizando partes importantes.

Aspectos como a estrutura organizacional e o fator humano podem afetar drasticamente no que diz respeito a qualidade de *software* devido à falta de coordenação entre equipes, falta de confiança e falha na comunicação (HERBSLEB e GRINTER, 1999). Suali *et al.* (2017) evidencia em RSL características semelhantes e reforça que a coordenação entre os desenvolvedores é crucial para a qualidade de *software*. Bird (2011) demonstra que a qualidade de um *software* está intimamente relacionada com o alto nível de responsabilidade do desenvolvedor em relação à uma parte do *software*, cujo termo é conhecido como *ownership*. Além disso, estudos empíricos comprovam que a qualidade do *software* e o número de falhas pós-lançamento está relacionada com a fragmentação das contribuições entre os desenvolvedores. Neste sentido, é necessário que a equipe mantenha o foco nos objetivos definidos, pois a divisão de contribuições pode afetar a qualidade e eficiência do *software* (PINZGER, NAGAPPAN e MURPHY 2011).

Figura 2. Características de qualidade para software.



Fonte: NBR ISO/IEC 9126-1

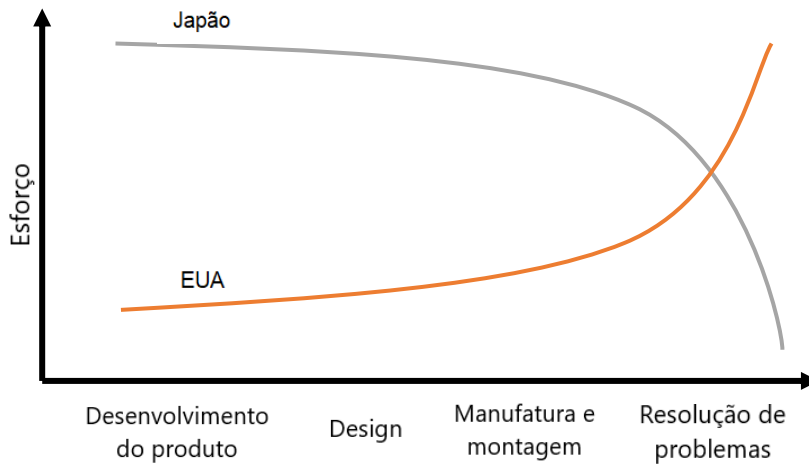
Para o processo de melhoria de um *software* é preciso que haja o apoio da direção; avaliações para motivação; melhorias na organização; times experientes; equipe treinada dentro da organização para modificar a cultura; estimular profissionais a trocarem experiências com outros dentro da organização; equipes devem escolher sua melhor forma de melhorar sua parte, possibilitando aumento da responsabilidade; envolver outros setores no processo de desenvolvimento; comunicar de forma constante e manter a pressão para atingir os resultados (WOHLWEND e ROSENBAUM, 1994). Os autores citados afirmam que não existem padrões universais para medir a qualidade de um *software* pois depende da pessoa que está avaliando e em quais circunstâncias o produto está sendo submetido. Além disso, diversos fatores influenciam para a qualidade de *software* como fatores humanos, cultura organizacional, coordenação entre equipes, entre outros, e que a qualidade deve ser medida durante o desenvolvimento do produto, bem como durante sua utilização pelo usuário, a fim de desenvolver melhorias futuras, pois segundo Moran, O'Connor e Borowitz (2013) não é possível melhorar o que não é medido. Neste raciocínio, Marx (2018) reforça que “se você não pode medir, você não pode mudar”

3.2 QUALITY FUNCTION DEPLOYMENT - QFD

O conceito e a metodologia do *QFD* foi criado pelo professor Akao e Mizuno entre 1960 e 1970. Também conhecida como a casa da qualidade, do inglês *house of quality*, esta técnica foi aplicada em 1972 pela empresa japonesa Mitsubishi Heavy Industries - Kobe Shipyard & Machinery Works que desenvolveu as matrizes do *QFD*, as quais se tornaram o núcleo da ferramenta (AKAO e MAZUR, 2003). Após a Ford e a Xerox levaram para os Estados Unidos em 1986, e que depois acabou por se expandir pela Europa (HAUSER, 1988). Isto diferenciou as indústrias japonesas das americanas e de outras localidades, pois elas conseguiam assegurar a produção de produtos com alta qualidade e atender as demandas dos clientes, com tempo de entrega e preços competitivos (THACKERAY e TREECK, 1990). Os autores citados anteriormente apontam que a principal diferença estava na forma com que os engenheiros gerenciavam o processo de desenvolvimento do produto, a Figura 3 demonstra que os engenheiros japoneses concentravam seus esforços nas fases iniciais de desenvolvimento do produto e *design*, enquanto que os americanos em atividades de resolução de problemas.

King (1987); Erikkson e McFadden (1993) descrevem o *QFD* como um sistema que permite incluir todos os colaboradores no processo de desenvolvimento e no aprimoramento contínuo de um produto. O *QFD* foi originalmente criado para analisar a voz do cliente, a fim de desenvolver produtos com maior qualidade para atender ou superar as expectativas do cliente, logo as principais funções do *QFD* são o desenvolvimento de produtos, gestão e análise das necessidades do cliente (CHAN e WU, 2002). Estes autores apontam que as aplicações do *QFD* foram expandidas para áreas do serviço, governo, bancos, saúde, educação e pesquisa, bem como em processos de *design*, planejamento, tomada de decisão, engenharia, gerenciamento, trabalho em equipe, sincronização e custeio, entre outros.

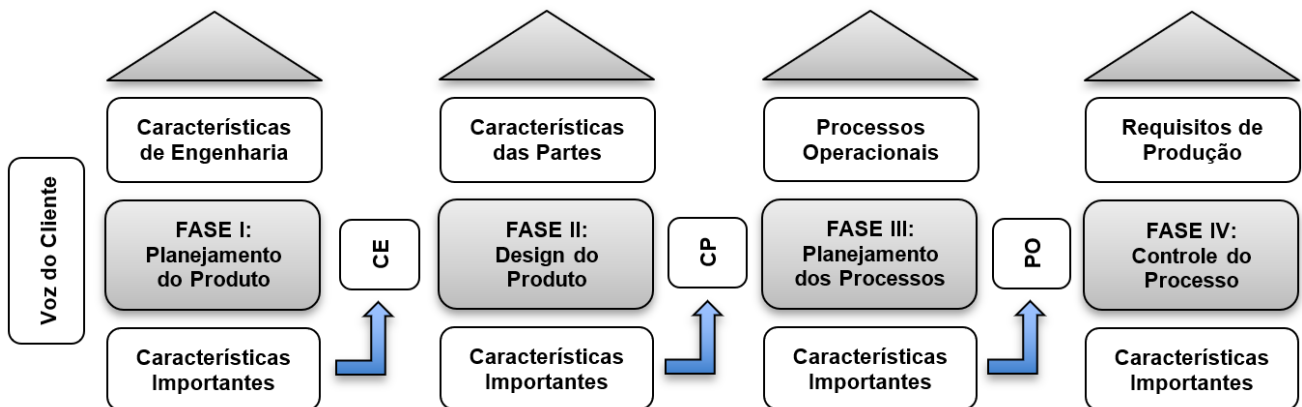
Figura 3. Comparação do esforço entre os engenheiros japoneses e americanos durante o processo de desenvolvimento do produto.



Fonte: Thackeray e Treeck (1990).

Neste sentido, Govers (2001) salienta que o QFD não é apenas uma ferramenta, mas uma filosofia de gerenciamento da qualidade que auxilia no aumento da satisfação do cliente, envolvimento entre os colaboradores e desenvolve uma gestão pró-ativa. Conforme Bouchereau e Rowlands (2000) a aplicação completa do QFD se dá em pelo menos 4 fases: I - Planejamento do produto; II - Design do produto; III - Planejamento dos processos; IV - Controle do processo. conforme descrito a seguir e na Figura 4.

Figura 4. As 4 fases do QFD.



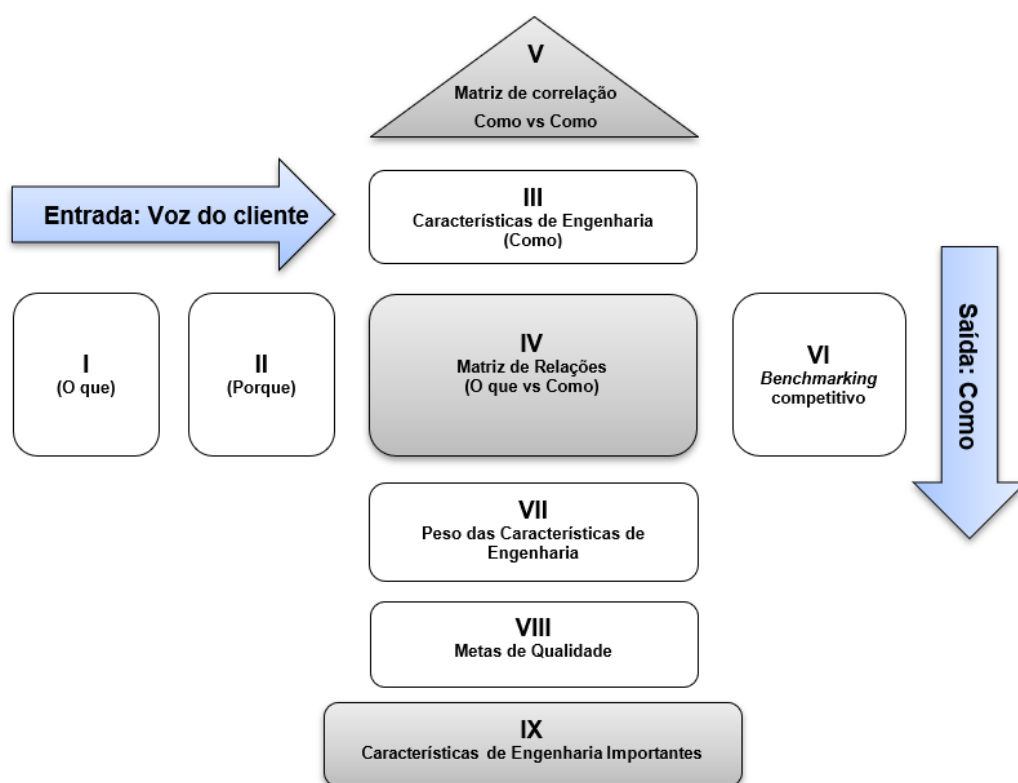
Fonte: Bouchereau e Rowlands (2000).

Para cada uma das fases é utilizado uma matriz que desdobra as necessidades dos clientes e os requisitos técnicos relacionados a ela. É considerado durante o processo a voz do cliente (como visto anteriormente), pois identifica o que (*what*) o cliente deseja, como (*how*) e quanto (*how much*) isso será realizado ou melhorado no processo considerando o menor custo possível. Cada fase gera um resultado (como) que irá servir de insumo para a próxima fase (o que) (CHAN e WU, 2005; BOUCHEREAU e ROWLANDS, 2000; AKAO, 2004). Guazzi et al. (1999) aponta algumas características fundamentais para aplicar o QFD:

- **Fazer perguntas ao consumidor:** questionar os clientes sobre suas reais necessidades e escutar na maneira que eles se expressam (voz do cliente), a fim de compreender a forma que definem e percebem os produtos na perspectiva deles;
- **Utilizar experiência e conhecimento da equipe multifuncional:** devem ser realizadas discussões entre equipes para identificar características mensuráveis que irão de encontro às necessidades e desejos do cliente;
- **Priorizar e concentrar esforços nas características mensuráveis:** Isto irá permitir que a voz do cliente seja preservada em todas as fases do desenvolvimento;
- **Marketing:** realizar o gerenciamento das expectativas do cliente e as ações dos concorrentes pela área de marketing, de tal forma que o produto ou serviço possa ser lucrativo por todo o seu ciclo de vida através da inovação constante.

A casa da qualidade (Figura 5) é a principal ferramenta de planejamento do QFD (WANG, XIE e GOH, 1998). Wu e Liao (2018) salientam que a casa da qualidade é uma ferramenta que permite traduzir a voz do cliente em requisitos de engenharia. O termo voz do cliente, do inglês *voice of customer*, teve sua origem no processo de aplicação do QFD (GRIFFIN e HAUSER, 1993).

Figura 5. A Casa da Qualidade.



Fonte: Bouchereau e Rowlands (2000); Akao (2004); Rozenfeld et al. (2006).

A voz do cliente precisa ser minuciosamente analisada e desdobrada para a voz da engenharia (características de engenharia) (COOPER e EDGETT, 2008), pois muitas vezes as reais necessidades dos clientes não são compreendidas da forma correta, levando a erros e falhas de desenvolvimento do produto, e por consequência a insatisfação do cliente e retrabalhos (CHAN e WU, 2002); (KUMARAVADIVEL, NATARAJAN e NOORUL HAQ, 2012). As matrizes do QFD ajudam a equipe de desenvolvimento do produto a definir metas para

os problemas que são mais importantes para o cliente e pensar em “como” isso pode ser alcançado tecnicamente, bem como avaliar produtos dos concorrentes através de *benchmarking* (BOUCHEREAU E ROWLANDS, 2000).

4 SOFTWARE QUALITY FUNCTION DEPLOYMENT – SQFD

Para atingir a total satisfação dos clientes é preciso unir a voz do cliente com o processo de desenvolvimento de produto através da utilização do *QFD* (KUMARAVADIVEL; NATARAJAN e NOORUL HAQ, 2012). Neste raciocínio, incorporar a voz do cliente dentro da organização é uma excelente estratégia competitiva, pois permite: compreender as reais necessidades dos clientes; possuir o controle e registro de requisitos e melhorias; reduzir o tempo de projeto e diminuição de alterações no produto (LIU, 2000; ELBOUSHI e SHERIF 1997). Kahraman, Ertay e Büyüközkan (2006) reforçam que entender as necessidades dos clientes aumenta drasticamente a aceitabilidade dos produtos no mercado e reduz tempo de desenvolvimento. Neste sentido, o *QFD* é uma ferramenta estratégica que auxilia as organizações a trazerem a voz do cliente para o processo de desenvolvimento de produtos e serviços (KARSAK, SOZER e ALPTEKIN, 2003). Para Sullivan (1986) a voz do cliente deve ser incorporada não somente durante as fases de desenvolvimento do produto, mas durante todo seu ciclo de vida. Alrabghi (2013) comenta que o *QFD* reduz a falha de comunicação entre os desenvolvedores de *softwares* e consumidores, logo, os usuários entendem quais melhorias com potencial podem ser adicionadas ao produto e começam a procurar por oportunidades no próprio ambiente de trabalho.

A indústria criativa de *software* tem aplicado o *SQFD* para aperfeiçoar a qualidade de seus produtos e para melhor atender as necessidades dos consumidores (SENER e KARSAK, 2012). Segundo Gong et al (2012) mais de 50% dos projetos de desenvolvimento de *software* não obtêm sucesso devido aos erros na análise de requisitos dos consumidores. Além disso, Sonda et al (2000); Scalvenzi e Cauchick Miguel (2016) afirmam que o processo de desenvolvimento de *software* utiliza a criatividade e não a repetitividade, tornando difícil os controles formais. Assim, o *SQFD* apoia o desenvolvimento de *softwares* no sentido de definir e priorizar indicadores, possibilitando maior controle dos processos durante sua criação, até fases posteriores de apoio ao usuário e manutenção. Outro ponto importante evidenciado por Barnett e Raja (1995) é o apoio que a ferramenta proporciona à equipe de desenvolvimento, a partir da priorização dos requisitos exigidos pelos consumidores, permitindo uma maior assertividade durante o processo de desenvolvimento ou modificação para melhorias.

Haag (1992) realizou um estudo referente ao desenvolvimento de 25 *softwares* em 6 empresas: IBM, AT&T, Hewlett-Packard, Texas Instruments, Digital Equipment e CSK. Estas organizações apresentaram resultados significativos a partir da utilização do *SQFD* para melhorar a qualidade de seus produtos, no entanto, os métodos não foram divulgados devido ao sigilo de informações.

Haag, Raja e Schkade (1996) apontam em seus estudos a comparação entre a aplicação do *SQFD* e abordagens tradicionais em 37 companhias de *software*, evidenciando resultados superiores relacionado à comunicação entre clientes e desenvolvedores, atendimento de requisitos, redução de erros nos sistemas, modificação do produto, tempo de teste e documentação consistente e completa. Herzwurm et al (1998) realizou em 1997 estudos empíricos de 16 projetos que aplicaram o *QFD* (sendo 7 de *softwares*), a partir de perguntas aos desenvolvedores, os quais responderam que a ferramenta proporcionou um aumento na qualidade do produto, maior cooperação das pessoas envolvidas no projeto, bem como economias durante o desenvolvimento

dos produtos por ter como foco no que é prioridade para os clientes. Segundo King (1987) ao priorizar as demandas é possível desenvolver respostas inovadoras com baixo custo e alta qualidade.

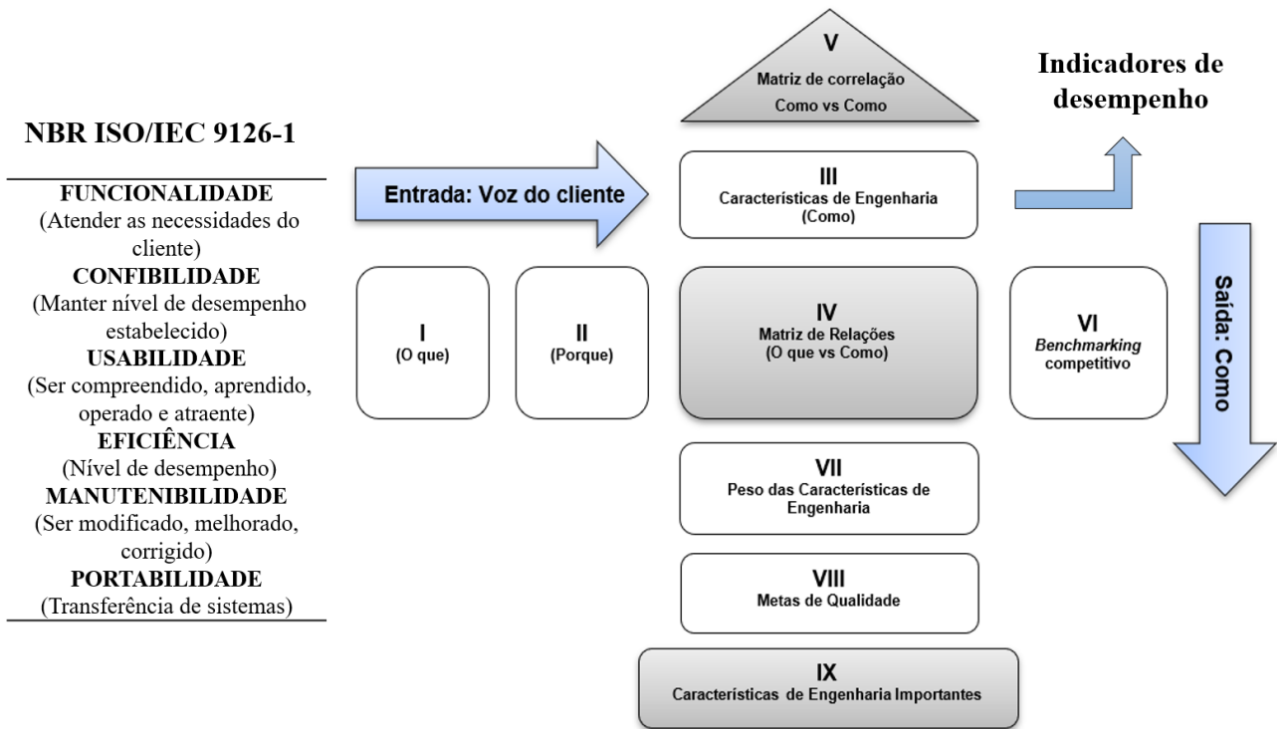
O *SQFD* relaciona os requisitos específicos do cliente com requisitos da engenharia de forma coerente (BARNETT e RAJA, 1995). Integra a satisfação dos requisitos dos clientes em todas atividades de desenvolvimento do *software* e aumenta a comunicação entre clientes e desenvolvedores, bem como entre a própria equipe de projeto (LIU, 2000). Além disso, o *SQFD* melhora a qualidade da mão de obra terceirizada e auxilia as companhias de *software* a se tornarem mais competitivas no mercado global (XIONG, WANG e WU, 2008) e também aumenta a qualidade dos produtos, reduzindo a necessidade de manutenção (HARTY, 2001). Neste contexto, o *SQFD* é uma ferramenta bem-sucedida do ponto de vista científico quando aplicada em *softwares* (ALRABGHI, 2013).

A utilização do *SQFD* implica em benefícios em *softwares* como melhoria de pesquisa e desenvolvimento (KIHARA, HUTCHINSON, DIMANCESCU, 1994; SHAH, GUO e XU, 2011) maior compreensão das necessidades dos clientes (BÜYÜKÖZKAN e ÇİFÇİ, 2012; KUO, 2013; SINGH et al., 2013); maior compreensão dos processos organizacionais (STUBBS e DIAZ, 1994; MOYNIHAN e BATSON, 2006); geração de perspectiva de prevenção de defeitos (ERIKKSON e MCFADDEN, 1993; STUBBS e DIAZ, 1994); identificação de lacunas nos sistemas (MOYNIHAN e BATSON, 2006; YADAV, BILEN e SOLANKI, 2006); redução de retrabalhos (ELBOUSHI e SHERIF, 1997; SHAH, GUO e XU, 2011); melhoria na organização e documentação (ERIKKSON e MCFADDEN, 1993; MOYNIHAN e BATSON, 2006); melhoria da comunicação interna (STUBBS e DIAZ, 1994; JACOBS, LUKE e REED, 1995; HERZWURM e SCHOCKERT, 2003); possibilita a criação de indicadores de desempenho (YADAV, BILEN e SOLANKI, 2006; YAMAKAWA, AOKI e MIGUEL, 2014) redução da complexidade de *softwares* (KIHARA, HUTCHINSON, DIMANCESCU, 1994; PAI, 2002); maior envolvimento com o cliente (HAAG, RAJA e SCHKADE, 1996; ELBOUSHI e SHERIF 1997; PAI, 2002); maior compreensão dos objetivos e metas organizacionais por parte dos colaboradores (ELBOUSHI e SHERIF 1997; SENER e KARSAK, 2010). No entanto, algumas dificuldades em relação a utilização do *SQFD* foram apontadas pelos autores como a necessidade de investimento adicional com encarregado pelas informações e facilitadores (ERIKKSON e MCFADDEN, 1993; HERZWURM e SCHOCKERT, 2003) necessidade de treinamento (KARLSSON, 1997); dificuldade de implementação devido à alta complexidade (HERZWURM e SCHOCKERT, 2003).

5 PROPOSTA DE IMPLANTAÇÃO DO QFD

A proposta de implantação do *QFD* para melhorias do ambiente computacional está estruturada em seis etapas. Na primeira etapa será aplicado um questionário fechado com 30 empresas envolvidas no projeto conforme apêndice A. Este questionário está estruturado conforme a NBR ISO/IEC 9126-1 que aponta 6 características fundamentais para obter qualidade em *softwares*: funcionalidade, confiabilidade, usabilidade, eficiência manutenibilidade e portabilidade. Estas características foram desdobradas em subcaracterísticas conforme a norma e reescritas para serem compreendidas pelos clientes. O resultado deste questionário será utilizado como a “voz do cliente” (*what*) na primeira matriz do *QFD* (Figura 6). Após, as subcaracterísticas serão desdobradas em requisitos técnicos ou “voz da engenharia” (*how*). Pretende-se com tais requisitos técnicos criar indicadores de desempenho para o *software*. O exemplo de aplicação da primeira matriz do *QFD* pode ser visualizado no apêndice B.

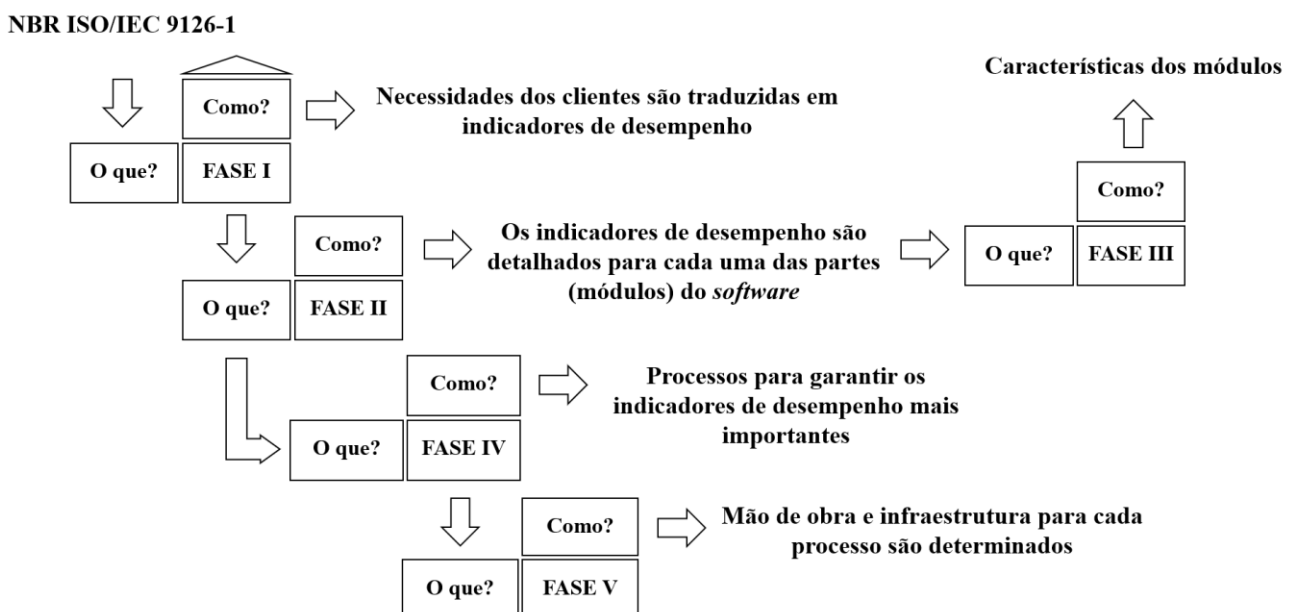
Figura 6. A Casa da Qualidade aplicada em software.



Fonte: Adaptado de Bouchereau e Rowlands (2000); Akao (2004); Rozenfeld et al. (2006).

Na segunda etapa (Figura 7) será utilizada a matriz das partes do produto (apêndice C). Nesta matriz serão relacionados os indicadores de desempenho (*what*) com os módulos do ambiente computacional (*how*).

Figura 7. Proposta de implantação do QFD em software.



Fonte: Dos autores.

A terceira etapa é caracterizada pela aplicação da matriz das características das partes do produto (apêndice D). Nesta matriz serão relacionadas os módulos do *software* (*what*) com as características dos módulos (*how*). A quarta etapa é caracterizada pela aplicação da matriz dos processos (apêndice E). Nesta matriz será relacionado os indicadores de desempenho da primeira matriz (*what*) com os processos (*how*). A quinta etapa é caracterizada pela matriz dos recursos humanos (apêndice F). Esta matriz relaciona os processos do *software* (*what*) com os recursos humanos necessários para aprimorar e gerenciar o *software* (*how*). Da mesma forma, a sexta etapa relaciona os processos (*what*) com a infraestrutura necessária (*how*) (apêndice G).

6 CONCLUSÃO

Espera-se que com a aplicação do *SQFD* haja um melhor envolvimento do cliente com os participantes do projeto de pesquisa. Desta forma, é essencial que os colaboradores compreendam os objetivos e metas organizacionais de modo a entenderem que os processos organizacionais e que a qualidade dos *softwares* precisa estar alinhada com às necessidades dos clientes. No entanto, para monitorar a qualidade do *software* é fundamental a criação de indicadores de desempenho, os quais irão dar suporte ao processo de tomada de decisão aos gestores e desenvolvedores e, conseqüentemente, aumentar a eficiência na utilização de recursos e identificar lacunas no sistema. Contudo, a maior organização e documentação, assim como a perspectiva gerada pelo uso *SQFD* auxilia para a redução de defeitos e, conseqüentemente, reduz retrabalhos e gargalos, gerando assim uma maior satisfação dos colaboradores. Trabalhos futuros devem ser realizados com foco na utilização do *SQFD* para desenvolvimento e melhoria de *softwares*, bem como a união de novas técnicas que possam complementar o *SQFD*.

AGRADECIMENTOS

Os autores agradecem ao CNPq e a Fapergs, pelo apoio financeiro, à UNISC e ao Programas de Pós-Graduação em Sistemas e Processos Industriais.

REFERÊNCIAS

- AKAO, Y. Quality function deployment. 2004.
- AKAO, Y., MAZUR, G. H. The leading edge in QFD: past, present and future. *International Journal of Quality & Reliability Management*, v. 20, n. 1, p. 20-35, 2003.
- ALRABGHI, L. O. QFD in software engineering. 2013. Tese de Doutorado. Kent State University.
- ANDERSON, N., DE DREU, C. KW, NIJSTAD, B. A. The routinization of innovation research: A constructively critical review of the state-of-the-science. *Journal of Organizational Behavior*, v. 25, n. 2, p. 147-173, 2004.
- ANDERSON, N., POTOČNIK, K., ZHOU, J. Innovation and creativity in organizations: A state-of-the-science review, prospective commentary, and guiding framework. *Journal of Management*, v. 40, n. 5, p. 1297-1333, 2014.
- BARNETT, W. D., RAJA, M. K. Application of QFD to the software development process. *International Journal of Quality & Reliability Management*, v. 12, n. 6, p. 24-42, 1995.
- BASIL, V. R., ROMBACH, H. D. The TAME project: Towards improvement-oriented software environments. *IEEE Transactions on software engineering*, v. 14, n. 6, p. 758-773, 1988.

- BASILI, V. R., SELBY, R. W. Data collection and analysis in software research and management. *Proceedings of the American Statistical Association and Biomeasure Society*, p. 13-16, 1984.
- BIRD, C. et al. Don't touch my code!: examining the effects of ownership on software quality. In: *Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering*. ACM, 2011. p. 4-14
- BOTELHO, L. L. R., DE ALMEIDA CUNHA, C. C., MACEDO, M. O método da revisão integrativa nos estudos organizacionais. *Gestão e Sociedade*, v. 5, n. 11, p. 121-136, 2011.
- BOTTANI, E. A fuzzy QFD approach to achieve agility. *International Journal of Production Economics*, v. 119, n. 2, p. 380-391, 2009.
- BOUCHEREAU, V., ROWLANDS, H. Methods and techniques to help quality function deployment (QFD). *Benchmarking: An International Journal*, v. 7, n. 1, p. 8-20, 2000.
- BÜYÜKÖZKAN, G., ÇİFÇİ, G. A new incomplete preference relations based approach to quality function deployment. *Information Sciences*, v. 206, p. 30-41, 2012.
- CARNEVALLI, J. A., MIGUEL, P. C. Review, analysis and classification of the literature on QFD—Types of research, difficulties and benefits. *International Journal of Production Economics*, v. 114, n. 2, p. 737-754, 2008.
- CHAN, L-K., WU, M-L. A systematic approach to quality function deployment with a full illustrative example. *Omega*, v. 33, n. 2, p. 119-139, 2005.
- CHAN, L-K., WU, M-L. Quality function deployment: A literature review. *European journal of operational research*, v. 143, n. 3, p. 463-497, 2002.
- COOPER, R. G., EDGETT, S. J. Maximizing productivity in product innovation. *Research-Technology Management*, v. 51, n. 2, p. 47-58, 2008.
- CORDEIRO, A. M. et al. Revisão sistemática: uma revisão narrativa. 2007.
- ELBOUSHI, M. I.; SHERIF, J. S. Object-oriented software design utilizing Quality Function Deployment. *Journal of Systems and Software*, v. 38, n. 2, p. 133-143, 1997.
- ERIKKSON, I., MCFADDEN, F. Quality function deployment: A tool to improve software quality. *Information and Software Technology*, v. 35, n. 9, p. 491-498, 1993.
- FALAGAS, M. E. et al. Comparison of PubMed, Scopus, web of science, and Google scholar: strengths and weaknesses. *The FASEB journal*, v. 22, n. 2, p. 338-342, 2008.
- FIGLIOLA, R. Metodologia da pesquisa: como planejar, executar e escrever um trabalho científico. João Pessoa: EDU, 2003.
- GERHARDT, T. E. et al. Métodos de pesquisa.[Organizado por] Tatiana Engel Gerhardt e Denise Tolfo Silveira; coordenado pela Universidade Aberta do Brasil-UAB/UFRGS e pelo Curso de Graduação Tecnológica-Planejamento e Gestão para o Desenvolvimento Rural da SEAD/UFRGS. Porto Alegre: Editora da UFRGS, 2009.
- GOMES, F. P.; ARAÚJO, R. M. Pesquisa Quanti-Qualitativa em Administração: uma visão holística do objeto em estudo. *Seminários em Administração*, v. 8, p. 1-11, 2005.
- GONG, L. et al., A Method for Determining Importance Degree of Customer Requirements in Software Quality Function Deployment. *Advances in Systems Science and Applications*, v. 12, n. 1, p. 46-53, 2012.
- GOVERS, C. PM. QFD not just a tool but a way of quality management. *International Journal of Production Economics*, v. 69, n. 2, p. 151-159, 2001.
- GRIFFIN, A., HAUSER, J. R. The voice of the customer. *Marketing science*, v. 12, n. 1, p. 1-27, 1993.

- GUAZZI, D. M. et al. Utilização do QFD como uma ferramenta de melhoria contínua do grau de satisfação de clientes internos: uma aplicação em cooperativas agropecuárias. 1999.
- GUMUSLUOGLU, L., ILSEV, A. Transformational leadership, creativity, and organizational innovation. *Journal of business research*, v. 62, n. 4, p. 461-473, 2009.
- HAAG, S. E. A field study of the use of quality function deployment (QFD) as applied to software development. 1992.
- HAAG, S., RAJA, M., SCHKADE, L. L. Quality function deployment usage in software development. *Communications of the ACM*, v. 39, n. 1, p. 41-49, 1996.
- HAAG, Stephen Eugene. A field study of the use of quality function deployment (QFD) as applied to software development. 1992.
- HARTY, D. Quality Function Deployment-An Overview of QFD and its Applications to Software Engineering. *Lockheed Martin Information Systems*, 2001.
- HAUSER, J. R. et al. The house of quality. 1988.
- HERBSLEB, J. D., GRINTER, R. E. Splitting the organization and integrating the code: Conway's law revisited. In: *Proceedings of the 21st international conference on Software engineering*. ACM, 1999. p. 85-95.
- HERZWURM, G. et al. Success factors of QFD projects. In: *Proceedings of the World Innovation and Strategy Conference*. 1998. p. 27-41.
- JACOBS, D. A., LUKE, S. R., REED, B. M. Using quality function deployment as a framework for process measurement. *Engineering Management Journal*, v. 7, n. 2, p. 5-9, 1995.
- JØRGENSEN, M. Software quality measurement. *Advances in engineering software*, v. 30, n. 12, p. 907-912, 1999.
- KAHRAMAN, C., ERTAY, T., BÜYÜKÖZKAN, G. A fuzzy optimization model for QFD planning process using analytic network approach. *European Journal of Operational Research*, v. 171, n. 2, p. 390-411, 2006.
- KANNABIRAN, G., SANKARAN, K. Determinants of software quality in offshore development—An empirical study of an Indian vendor. *Information and Software Technology*, v. 53, n. 11, p. 1199-1208, 2011.
- KARLSSON, J. Managing software requirements using quality function deployment. *software quality journal*, v. 6, n. 4, p. 311-326, 1997.
- KARSAK, E. E., SOZER, S., ALPTEKIN, S. Emre. Product planning in quality function deployment using a combined analytic network process and goal programming approach. *Computers & industrial engineering*, v. 44, n. 1, p. 171-190, 2003.
- KHOSRAVI, A. et al. Toward software quality enhancement by Customer Knowledge Management in software companies. *Telematics and Informatics*, 2017.
- KIHARA, T., HUTCHINSON, C E., DIMANCESCU, Dan. Designing software to the voice of the customer: New uses of QFD and quantification method of type III for decomposition of the requirements. *Quality Engineering*, v. 7, n. 1, p. 113-137, 1994.
- KIM, S-H., KIM, W-J. Evaluation of software quality-in-use attributes based on analysis network process. *Cluster Computing*, p. 1-14, 2018.
- KING, R. Listening to the voice of the customer: Using the quality function deployment system. *Global Business and Organizational Excellence*, v. 6, n. 3, p. 277-281, 1987.
- KONDO, Y. Customer satisfaction: how can I measure it?. *Total Quality Management*, v. 12, n. 7-8, p. 867-872, 2001.

- KUMARAVADIVEL, A., NATARAJAN, U., NOORUL HAQ, A. Performance measurement and determination of optimal base stock level inventory system to improve the customer satisfaction in the Six Sigma environment. *International Journal of Productivity and Quality Management*, v. 9, n. 3, p. 382-403, 2012.
- KUO, T. C. Mass customization and personalization software development: a case study eco-design product service system. *Journal of Intelligent Manufacturing*, v. 24, n. 5, p. 1019-1031, 2013.
- LIU, X. F. Software quality function deployment. *IEEE Potentials*, v. 19, n. 5, p. 14-16, 2000.
- MARTINS, E. C., TERBLANCHE, F. Building organisational culture that stimulates creativity and innovation. *European journal of innovation management*, v. 6, n. 1, p. 64-74, 2003.
- MARX, M. et al. If you can't measure it-you can't change it—a longitudinal study on improving quality of care in hospitals and health centers in rural Kenya. *BMC health services research*, v. 18, n. 1, p. 246, 2018.
- MCLEAN, L. D. Organizational culture's influence on creativity and innovation: A review of the literature and implications for human resource development. *Advances in developing human resources*, v. 7, n. 2, p. 226-246, 2005.
- MIGUEL, P. A. C. et al. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. *Revista Produção*, v. 17, n. 1, p. 216-229, 2007.
- MORAN, V., O'CONNOR, S., BOROWITZ, M. International approaches to measuring the quality of mental health care. *Epidemiology and psychiatric sciences*, v. 22, n. 1, p. 3-7, 2013
- MOYNIHAN, G. P., BATSON, R. G. Development of a matrix methodology for database reengineering and improvement: an e-government case study. *Electronic Government, an International Journal*, v. 3, n. 2, p. 190-203, 2006.
- NAGAPPAN, N., MURPHY, B., BASILI, V. The influence of organizational structure on software quality. In: Software Engineering, 2008. ICSE'08. *ACM/IEEE 30th International Conference on*. IEEE, 2008. p. 521-530.
- PAI, W. C. A quality-enhancing software function deployment model. *Information Systems Management*, v. 19, n. 3, p. 20-24, 2002.
- PINZGER, M., NAGAPPAN, N., MURPHY, B. Can developer-module networks predict failures?. In: Proceedings of the 16th ACM SIGSOFT *International Symposium on Foundations of software engineering*. ACM, 2008. p. 2-12.
- RAMOS VOSGERAU, D. S., ROMANOWSKI, J. P. Estudos de revisão: implicações conceituais e metodológicas. *Revista Diálogo Educacional*, v. 14, n. 41, 2014.
- RAUPP, F. M., BEUREN, I. M. Metodologia da Pesquisa Aplicável às Ciências. _____ Como elaborar trabalhos monográficos em contabilidade: teoria e prática. São Paulo: Atlas, 2006.
- SCALVENZI, L., CAUCHICK MIGUEL, P. A. QFD aplicado ao desenvolvimento de software: priorização de requisitos do cliente em uma matriz de funções. *Exacta*, v. 14, n. 4, 2016
- ŞEN, C. G., BARAÇLI, H. Fuzzy quality function deployment based methodology for acquiring enterprise software selection requirements. *Expert Systems with Applications*, v. 37, n. 4, p. 3415-3426, 2010.
- SENER, Z., KARSAK, E. E. A decision model for setting target levels in software quality function deployment to respond to rapidly changing customer needs. *Concurrent Engineering*, v. 20, n. 1, p. 19-29, 2012.
- SHAH, S., GUO, H., XU, X. Development of a web-based quality function deployment system. *International Journal of Internet Manufacturing and Services*, v. 3, n. 1, p. 16-31, 2011.
- SINGH, M. et al. Selection of knowledge management tools using integrated AHP-QFD model. *International Journal of Management and Decision Making*, v. 12, n. 4, p. 376-392, 2013.
- SONDA, F. A., RIBEIRO, J. L. D., ECHEVESTE, M. E. A aplicação do QFD no desenvolvimento de software: um estudo de caso. *Produção*, v. 10, n. 1, p. 51-75, 2000

STOJCIC, N., HASHI, I., ORLIC, E. Creativity, innovation effectiveness and productive efficiency in the UK. *European Journal of Innovation Management*, 2018.

STUBBS, N., DIAZ, M. Impact of QFD utilization in the development of a nondestructive damage detection system for aerospace structures. *International Journal of Materials and Product Technology*, v. 9, n. 1-3, p. 3-22, 1994.

SUALI, A. J. et al. Developers' coordination issues and its impact on software quality: A systematic review. In: Science in Information Technology (ICSITech), 2017 3rd International Conference on. IEEE, 2017. p. 659-663.

SULLIVAN, L.P. Quality Function Deployment. *Quality Progress*, v. 19, n.6, p. 39-50, 1986.

SUN, Y., LIU, X. F. Business-oriented software process improvement based on CMMI using QFD. *Information and software technology*, v. 52, n. 1, p. 79-91, 2010.

TRANFIELD, D., DENYER, D., SMART, P. Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British journal of management*, v. 14, n. 3, p. 207-222, 2003.

TURRIONI, J. B., MELLO, C. H. P. Metodologia de pesquisa em engenharia de produção: estratégias, métodos e técnicas para condução de pesquisas quantitativas e qualitativas. Apostila do curso de Especialização em Qualidade e Produtividade. Universidade Federal de Itajubá, Itajubá, MG, 2012.

VAN DE VEN, A. H. The innovation journey: you can't control it, but you can learn to maneuver it. *Innovation*, v. 19, n. 1, p. 39-42, 2017

VINODH, S., KUMAR CHINTHA, S. Application of fuzzy QFD for enabling leanness in a manufacturing organisation. *International Journal of Production Research*, v. 49, n. 6, p. 1627-1644, 2011

WANG, H., XIE, M., GOH, T. N. A comparative study of the prioritization matrix method and the analytic hierarchy process technique in quality function deployment. *Total Quality Management*, v. 9, n. 6, p. 421-430, 1998.

WEST, M. A. Sparkling fountains or stagnant ponds: An integrative model of creativity and innovation implementation in work groups. *Applied psychology*, v. 51, n. 3, p. 355-387, 2002.

WOHLWEND, H., ROSENBAUM, S. Schlumberger's software improvement program. *IEEE Transactions on software engineering*, v. 20, n. 11, p. 833-839, 1994.

WOODMAN, R. W., SAWYER, J. E., GRIFFIN, R. W. Toward a theory of organizational creativity. *Academy of management review*, v. 18, n. 2, p. 293-321, 1993.

WU, X., LIAO, H. An approach to quality function deployment based on probabilistic linguistic term sets and ORESTE method for multi-expert multi-criteria decision making. *Information Fusion*, v. 43, p. 13-26, 2018.

XIONG, W., WANG, X-T., WU, Z-X. Study of a customer satisfaction-oriented model for outsourcing software quality management using Quality Function Deployment (QFD). In: Wireless Communications, Networking and Mobile Computing, 2008. *WiCOM'08. 4th International Conference on*. IEEE, 2008. p. 1-5

YADAV, O. P., BILEN, C., SOLANKI, R. Driving change in manufacturing quality systems by selecting best-fit off-the-shelf software platform. *International Journal of Process Management and Benchmarking*, v. 1, n. 2, p. 146-163, 2006.

YAMAKAWA, E. K., AOKI, A. R., MIGUEL, P. A. C. aplicação de fuzzy quality function deployment para seleção de indicadores de eficiência energética para utilização em um software de gestão de energia. *Ciência & Engenharia*, v. 23, n. 2, p. 21-31, 2014.

ZHOU, J., SHALLEY, C. E. Research on employee creativity: A critical review and directions for future research. In: Research in personnel and human resources management. *Emerald Group Publishing Limited*, 2003. p. 165-217.

ZUPIC, I., ČATER, T. Bibliometric methods in management and organization. *Organizational Research Methods*, v. 18, n. 3, p. 429-472, 2015.

Apêndice A

Esta pesquisa procura levantar as preferências dos usuários de um software tem como objetivo desafiar as capacidades dos colaboradores das organizações parceiras do parque científico e tecnológico para que discutam ideias e trabalhem de forma colaborativa na composição e aprimoramento de ideias de produtos ou serviços, visando melhorias nos processos organizacionais.										
Obrigado pela colaboração.										
										😊
										😊
										😊
										😊
										😊
Pensando na utilização de um software de criatividade e inovação.										
Por favor, indique o grau de importância quanto à situação atual dos itens abaixo mencionados.										
										😊😊😊😊😊
1. Em relação a FUNCIONALIDADE, o quanto importante é:										
Funções que atendem minhas necessidades	1	2	3	4	5	6	7	8	9	10
Fazer o que foi proposto de forma correta;	1	2	3	4	5	6	7	8	9	10
Evitar acesso não autorizado.	1	2	3	4	5	6	7	8	9	10
2. Em relação a CONFIABILIDADE, o quanto importante é:										
Baixa frequência de falhas;	1	2	3	4	5	6	7	8	9	10
Continuar funcionando quando ocorre uma falha.;	1	2	3	4	5	6	7	8	9	10
Recuperar dados em caso de falha	1	2	3	4	5	6	7	8	9	10
3. Em relação a USABILIDADE, o quanto importante é:										
Fácil de entender as funcionalidades e aplicações;	1	2	3	4	5	6	7	8	9	10
Fácil de aprender a usar;	1	2	3	4	5	6	7	8	9	10
Auxiliar a operar e controlar.	1	2	3	4	5	6	7	8	9	10
4. Em relação a EFICIÊNCIA, o quanto importante é:										
Velocidade na execução e baixo tempo de resposta;	1	2	3	4	5	6	7	8	9	10
Baixo custo, tempo para completar tarefa e esforço.	1	2	3	4	5	6	7	8	9	10
5. Em relação a MANUTENÇÃO, o quanto importante é:										
Diagnóstico de deficiências ou causas de falhas no software;	1	2	3	4	5	6	7	8	9	10
Baixo risco de fazer alterações;	1	2	3	4	5	6	7	8	9	10
Fácil de testar quando realizar alterações.	1	2	3	4	5	6	7	8	9	10
6. Em relação a PORTABILIDADE o quanto importante é:										
Adaptável conforme minhas necessidades;	1	2	3	4	5	6	7	8	9	10
Fácil de instalar em vários ambientes;	1	2	3	4	5	6	7	8	9	10
Funcionar bem com outros softwares instalados.	1	2	3	4	5	6	7	8	9	10

Apêndice B

Matriz de qualidade		NBR ISO/IEC 9126-1	Desdobramento das características da qualidade - Indicadores de desempenho													MI EI IDI*					
Características da qualidade de software	Subcaracterísticas da qualidade de software	Desdobramento das subcaracterísticas de qualidade - (Voz do cliente)	Gran de importância	Desdobramento das características da qualidade - Indicadores de desempenho													MI	EI	IDI*		
				Quantidade de ideias criativas e inovações geradas	Nível de controle de erros	Nível de segurança	Número de erros não detectados automaticamente (qtd. de erros / mês)	Delatagem de atualização do sistema (dias)	Quantidade de falhas monitoradas pelo software	nº de vezes que o software para de funcionar	Gran de satisfação do cliente.	Tempo de aprendizado (dias)	Tempo médio p/ entrada de dados (segundos)	Quantidade de horas de treinamento (horas)	Tempo de resposta (segundos)	Quantidade de falhas expostas pelo software				Tempo médio p/ o soft. realizar o atendimento (horas)	Quantidade de modificações de configuração
FUNCIONALIDADE (Atender as necessidades do cliente)	Adequação	Funções que atendem mínimas necessidades	1	9			3	1	3	1	3	1	1						0.5	1.0	0.71
	Acurácia	Fazer o que foi proposto de forma correta	3	9	3	9	9	3	3	1	1	1	1						1.0	1.0	3.00
	Segurança de acesso	Evitar acesso não autorizado	2	3	3	9	9	9	1	1	1	1	1						0.5	1.0	1.41
	Maturidade	Baixa frequência de falhas	4	1			3	3	9	1	1	1	1						1.5	1.5	6.00
CONFIBILIDADE (Manter nível de desempenho estabelecido)	Tolerância a falhas	Continuar funcionando quando ocorre uma falha	2	1			3	9	9	3	1	1							1.5	1.5	3.00
	Recuperabilidade	Recuperar dados em caso de falha	3	1	3	1	9	3	9	3	1	1							1.5	0.5	2.60
	Inteligibilidade	Fácil de entender as funcionalidades e aplicações	5	3			1				9	9	1						1.5	0.5	4.33
	Apreensibilidade	Fácil de aprender a usar	4	3			1				9	9	1						1.5	0.5	3.46
EFICIÊNCIA (Nível de desempenho)	Operacionalidade	Auxiliar a operar e controlar	4	3			1				9	9	1						1.5	0.5	3.46
	Comportamento do tempo	Velocidade na execução e baixo tempo de resposta	1	1			1	1	1	3	1	9	9						1.5	0.5	1.30
	Utilização de recursos	Baixo custo, tempo para completar tarefa e esforço	4	1			1	1	1	3	1	9	9						1.5	0.5	5.20
	Analisabilidade	Diagnóstico de deficiências ou causas de falhas no software	1	9			3	9	3	1	1	1							1.5	0.5	0.87
MANUTENIBILIDADE (Ser modificado, melhorado, corrigido)	Estabilidade	Baixo risco de fazer alterações	3				3	1	3	1	1								1.5	0.5	2.60
	Testabilidade	Fácil de testar quando realizar alterações	3				3	1	3	1	1								1.5	0.5	2.60
	Adaptabilidade	Adaptável conforme necessidades	5	3			3			3	1								1.5	0.5	4.33
	Capacidade para ser instalado	Fácil de instalar em vários ambientes	1	3			3			3	1								1.5	0.5	0.87
PORTABILIDADE (Transferência de sistemas)	Coexistência	Funcionar bem com outros softwares instalados	1	1			3			3	1								1.5	0.5	0.87
				10.07	11.39	2.95	9.26	15.66	11.54	8.04	14.41	10.34	7.48	10.34	6.36	3.90	8.88	17.00	0.87	2.81	
		Análise Competitiva	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
		Dificuldade de atuação	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	IQJ*	7.12	8.05	2.09	6.55	11.07	8.16	5.69	10.19	7.31	5.29	7.31	6.28	12.02	0.61	1.99					

Apêndice C

Matriz das partes	Desdobramento das características da qualidade																				
Módulos do software	Quantidade de ideias criativas e inovações geradas	Nível de controle de erros	Nível de segurança	Número de erros não detectados automaticamente (qtde. de erros / mês)	Defasagem de atualização do sistema (dias)	Quantidade de falhas monitoradas pelo software	nº de vezes que o software para de funcionar	Grau de satisfação do cliente.	Tempo de aprendizado (dias)	Tempo médio p/ entrada de dados (segundos)	Quantidade de horas de treinamento (horas)	Tempo de resposta (segundos)	Quantidade de falhas expostas pelo software	Tempo médio p/ o soft. realizar o atendimento (horas)	Quantidade de modificações de configuração	Quantidade de usuários que não conseguiram instalar	Quantidade de problemas decorrentes a outros softwares	iPi	Fi	Ti	IPi*
	Importância das CQs	7,12	8,05	2,09	6,55	11,07	8,16	5,69	10,19	7,31	5,29	7,31	4,50	2,76	6,28	12,02	0,61	1,99			
Colaboração	9				1		1	3	3	3	1	3						17,0	1,0	1,0	17,01
Storytelling	9				1		1	3	3	9	3	3						21,6	0,5	2,0	21,64
Caixa de Ferramentas	3				1		1	9	1	9	1	3						20,6	2,0	1,0	29,07
Retenção do Conhecimento.	9				1		1	9	1			3						19,3	1,0	2,0	27,34

Apêndice D

Matriz das características das partes do produto	Desdobramento das características das partes do produto											
Módulos do software	Importância das partes	Menu de operações	Aba de elaboração de ideias	Validação inicial de ideias	Divulgação de ideias	Inscrição de colaboradores	Colaboração de ideias	Agregar perfis semelhantes (Persona)	Opiniões específicas sobre a ideia (point of view)	Aprovação final		
	Colaboração	17,01	9	9	3	3	1	3				
Storytelling	21,64	9	9	3	9	1	3					
Caixa de Ferramentas	29,07	9	3	1	3	9	9	9	9			
Retenção do Conhecimento	27,34	9	3	9	1	1						9
Importância das CPs	85,56	51,71	39,11	36,04	32,76	37,76	26,16	26,16	24,61			

Apêndice E

Matriz dos processos	Desdobramento das características de qualidade																				
Processos do software	Quantidade de ideias criativas e inovações geradas	Nível de controle de erros	Nível de segurança	Número de erros não detectados automaticamente (qtde. de erros / mês)	Defasagem de atualização do sistema (dias)	Quantidade de falhas monitoradas pelo software	nº de vezes que o software para de funcionar	Grau de satisfação do cliente.	Tempo de aprendizado (dias)	Tempo médio p/ entrada de dados (segundos)	Quantidade de horas de treinamento (horas)	Tempo de resposta (segundos)	Quantidade de falhas expostas pelo software	Tempo médio p/ o soft. realizar o atendimento (horas)	Quantidade de modificações de configuração	Quantidade de usuários que não conseguiram instalar	Quantidade de problemas decorrentes a outros softwares	iPi	Fi	Ti	IPi*
	Importância das CQs	7,12	8,05	2,09	6,55	11,07	8,16	5,69	10,19	7,31	5,29	7,31	4,50	2,76	6,28	12,02	0,61	1,99			
Análise dos atributos	1	9	9	9		9	9	9	1			9	9					40,6	1,0	1,0	40,58
Projeto do software	3	9	9	9	3	9	9	9	3	3		9	9			9		65,6	1,0	1,0	65,58
Desenvolvimento	9	9	9	9	3	9	9	9	3	3		9	9	9	9	9		65,8	1,0	1,0	65,81
Documentação	3	1	1	1	1	1	1	3	3				1			3		12,0	1,0	1,0	12,01
Implantação	1	1	1	1		1	1	3	1				1					7,8	1,0	1,0	7,83
Treinamento para empresas	9	1	1	1		1	1	3	9	3	9		1	1		3		28,4	1,0	1,0	28,36
Suporte/Manutenção	1	1	1	1		1	1	3					1	9		3		12,9	1,0	1,0	12,93
Evolução	9	9	9	9	9	9	9	9	3	3		9	9		9			75,9	1,0	2,0	107,40

Apêndice F

Matriz dos recursos humanos		Recursos humanos						
Processos do software	Importância dos processos	Coordenador de projeto	Analista	Programador	Assesor de custo	Analista de suporte	Instrutor	Atendente
Projeto do software	65,58	3	9	9	9			
Desenvolvimento	65,81	3	9	9	1			
Documentação	12,01	3			1	1	1	
Implantação	7,83	3			1			1
Treinamento para empresas	28,36	3			9	9	9	1
Suporte/Manutenção	12,93	3		3	9	9	3	9
Evolução	107,40	3		9	9			
Importância dos recursos	102,2	102,2	154,8	218,8	205,5	38,4	30,6	15,3
Quantidade		1,00	1,00	1,00	1,00	1,00	1,00	1,00
Salário		R\$ 3.000,00	R\$ 2.000,00	R\$ 2.500,00	R\$ 1.500,00	R\$ 1.500,00	R\$ 1.500,00	R\$ 1.500,00
Encargos		R\$ 2.100,00	R\$ 1.400,00	R\$ 1.750,00	R\$ 1.050,00	R\$ 1.050,00	R\$ 1.050,00	R\$ 1.050,00
% tempo dedicado		20%	20%	20%	20%	20%	20%	20%
Custo mensal		R\$ 1.020,00	R\$ 680,00	R\$ 850,00	R\$ 510,00	R\$ 510,00	R\$ 510,00	R\$ 510,00

Apêndice G

Matriz da infraestrutura		Recursos de infraestrutura						
Processos do software	Importância dos processos	Banco de dados	Sistema operacional	Ferramentas de análise	Ferramentas de programação	Software para suporte	Computador / Impressoras	Software para apresentação
Projeto do software	65,58	9	9	9	9	9		
Desenvolvimento	65,81	3	9		9	3		
Documentação	12,01						9	9
Implantação	7,83						9	9
Treinamento para empresas	28,36							9
Suporte/Manutenção	12,93		1		3	9		
Evolução	107,40	1	3	3	9			
Importância dos recursos	89,5	89,5	155,8	127,8	218,8	90,4	17,9	43,4
Quantidade		1	1	10	5	1	5	2
Custo do Equipamento		R\$ 200,00	R\$ 2.000,00	R\$ 200,00	R\$ 500,00	R\$ 300,00	R\$ 2.000,00	R\$ 1.000,00
Tempo de Amortização (anos)		3	5	5	5	5	5	3
Custo de Operação e Manutenção (mês)		R\$ 5,00	R\$ 50,00	R\$ 50,00	R\$ 200,00	R\$ 100,00	R\$ 200,00	R\$ 200,00
Tempo de Uso (%)		30%	30%	50%	20%	30%	30%	30%
Custo Mensal		R\$ 3,17	R\$ 25,00	R\$ 41,67	R\$ 48,33	R\$ 31,50	R\$ 110,00	R\$ 76,67