



APLICAÇÃO DO SQFD EM UM SOFTWARE DE VENDAS NO CONTEXTO DE DESENVOLVIMENTO ÁGIL GERENCIADO PELO SCRUM

Larissa Ramalho Martins (larissaramalhom@gmail.com) - Escola de Engenharia, Universidade Federal de Minas Gerais

Júlio César Fonseca de Melo (juliocfmelo@gmail.com) - Escola de Engenharia, Universidade Federal de Minas Gerais

Raoni Barros Bagno (rbagno@dep.ufmg.br) - Escola de Engenharia, Universidade Federal de Minas Gerais

RESUMO

No Scrum o envolvimento do cliente no desenvolvimento do produto é feito pelo Dono do Produto, mas a avaliação e priorização de suas demandas não os envolve diretamente, possibilitando que a organização da Lista de Itens seja feita de forma tendenciosa. O Desdobramento da Função Qualidade (QFD) é um método que permite ao time de desenvolvimento especificar as necessidades e desejos do cliente, sendo o Software Quality Function Deployment (SQFD) uma vertente do QFD aplicada a softwares. Este trabalho consiste na aplicação do SQFD no desenvolvimento de um software, no qual buscou-se trazer a voz do cliente de forma mais ativa, em especial durante a definição e priorização da Lista de Itens. A metodologia que orientou o estudo é a Pesquisa-Ação. Dados foram obtidos com questionários e entrevistas semiestruturadas com clientes específicos. Dentre os principais resultados, destacam-se: adaptação das etapas do SQFD; estruturação das tabelas de Requisitos do Cliente, Qualidade Planejada, Voz do Engenheiro (a partir do método de descrição de Job Stories) e Matriz SQFD; identificação de funcionalidades novas ou a serem melhoradas; criação de uma Lista de Itens. Com a aplicação conjunta dos métodos, este estudo apoiou a empresa estudada a levantar necessidades dos clientes que antes não eram consideradas.

Palavras chave: Desdobramento da Função Qualidade (QFD); Software Quality Function Deployment (SQFD); Scrum; Desenvolvimento Ágil.

Área: Ferramentas e métodos de desenvolvimento de produtos e serviços.

1. INTRODUÇÃO

Em um ambiente complexo e sofrendo constantes mudanças, a agilidade de uma organização deixou de ser uma necessidade e passou a ser uma condição para entrar ou se manter no mercado (STOICA; MIRCEA; GHILIC- MICU, 2013). Com isso, várias empresas passaram a adotar metodologias de desenvolvimento ágil de softwares, como o *Scrum*. A ideia central do *Scrum* baseia-se no fato de que vários processos durante o desenvolvimento não são previsíveis (K. VLAANDEREN et al., 2011). Por se tratar não só de uma metodologia de desenvolvimento de softwares, mas, principalmente, de gestão do desenvolvimento (PERMANA, 2015), é possível aliar outros métodos para desenvolvimento de softwares ao *Scrum*. Além disso, no *Scrum* o Dono do Produto é o responsável por tomar decisões em

relação às prioridades da Lista de Itens durante todo o processo, consultando todos os *stakeholders* e a equipe, de forma a se certificar de que elas representam tanto o que as pessoas querem, quanto o que pode ser construído (SUTHERLAND, 2014). Com isso, ainda que as demandas sejam levantadas pelos clientes, a avaliação e priorização das mesmas não os envolve diretamente, possibilitando que a organização da lista de itens a serem desenvolvidos seja feita de forma tendenciosa.

O Desdobramento da Função Qualidade (QFD) é um método estruturado para desenvolvimento de produtos, que permite que o time de desenvolvimento especifique, de forma clara, as necessidades e desejos do cliente, e que avalie sistematicamente cada um dos elementos propostos para o produto em termos de seu impacto no atendimento dessas necessidades (CARRUTHERS, 1999). O *Software Quality Function Deployment* (SQFD) é o QFD aplicado ao desenvolvimento de softwares. Embora o SQFD possua os mesmos objetivos do QFD no que diz respeito a ouvir a voz do cliente, dentre outras diferenças, a principal está em sua aplicação, devido às características da indústria de software e do software em si. Com isso, o método permite uma comunicação sistemática e mais informal entre clientes e desenvolvedores (HERZWURM, 2003).

Nesse contexto, o escopo desse trabalho envolveu a aplicação do SQFD no desenvolvimento de um software de automação de vendas, em um contexto de desenvolvimento ágil juntamente com o *Scrum*. Com isso, o objetivo foi trazer o cliente e outros *stakeholders* para o desenvolvimento de forma mais ativa e participativa a partir de um projeto de pesquisa-ação, tendo sua base na análise do contexto da organização e no atual estágio de desenvolvimento do software. A partir dos resultados levantados com a aplicação do SQFD, foi criada uma Lista de Itens com os itens demandadas por todos os *stakeholders* da empresa, que serviram como insumo para os *Sprints* do *Scrum*.

2. REVISÃO BIBLIOGRÁFICA

Existem inúmeros modelos para o processo de desenvolvimento de um software, sendo cada um deles criado para atender a um determinado objetivo (STOICA; MIRCEA; GHILIC-MICU, 2013). Como o trabalho em questão trata-se da aplicação do *Scrum* e do SQFD, o foco foi no modelo Cascata, ou tradicional, e no modelo Incremental, o qual é a base para as metodologias de desenvolvimento ágil (STOICA; MIRCEA; GHILIC-MICU, 2013).

2.1 O *Scrum* no contexto das metodologias para o desenvolvimento de softwares

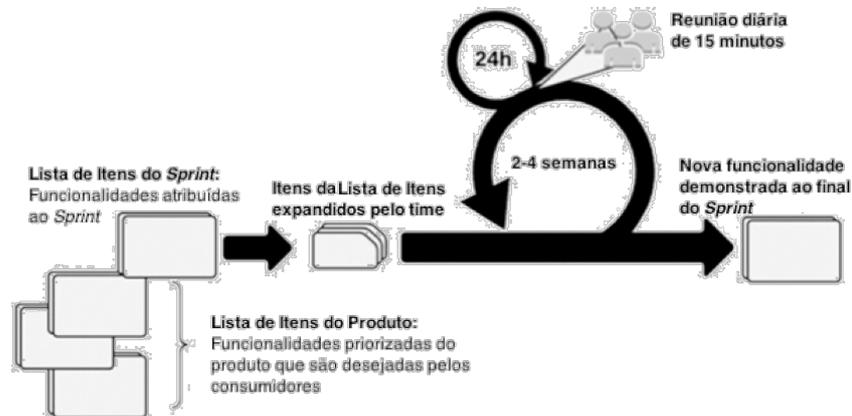
Segundo Aitken e Ilango (2013), o modelo em Cascata envolve completar sequencialmente cada uma das fases de desenvolvimento e, apenas quando cada fase está totalmente completa, partir para a fase seguinte. Já no modelo Incremental, os requisitos são divididos em subconjuntos (STOICA; MIRCEA; GHILIC-MICU, 2013). O modelo Incremental corresponde melhor ao mundo real do que o modelo em Cascata, visto que reconhece a necessidade de voltar e modificar resultados de fases anteriores, e que os desenvolvedores de software nunca conseguem, de fato, completar totalmente qualquer atividade (AITKEN; ILANGO, 2013). Esse modelo é a base para as metodologias de desenvolvimento ágil (STOICA; MIRCEA; GHILIC-MICU, 2013). Segundo Sommer *et al.* (2015), dentre as metodologias ágeis principais para o desenvolvimento de softwares, o *Scrum* é o único que é explicitamente intencionado para o gerenciamento de projetos em todo o processo de desenvolvimento de softwares.

Segundo Permana (2015), o *Scrum* foi desenvolvido com o objetivo de se tornar uma metodologia de gerenciamento e desenvolvimento que segue os princípios da metodologia ágil. Segundo Sommer *et al.* (2015), o *Scrum* pode ser executado como um processo de três estágios: (i) o desenvolvimento da “Lista de Itens”; (ii) o *Sprint* principal; (iii) e os *Sprints* diários. Trata-se de um modelo iterativo, no qual esses estágios repetem-se inúmeras vezes ao longo do processo de desenvolvimento do software.

Para Permana (2015), além do Mestre *Scrum*, cuja função principal é remover obstáculos, existem outros dois papéis fundamentais no *Scrum*: o Dono do Produto e o Time de Desenvolvimento. O primeiro é o responsável por determinar as especificações ou o negócio de aplicações de software a serem desenvolvidos. Já o time é um grupo de pessoas que executa um projeto. Os itens da Lista são estimados por meio da definição de pontos ou dias ideais. Para estimar os itens, Sutherland (2014) sugere um método chamado “Pôquer do Planejamento”. Nesse método, todos os envolvidos na reunião de planejamento fazem estimativas do esforço necessário para concluir cada item, além do impacto de cada um. Para isso, cada pessoa dá uma nota segundo a sequência de Fibonacci.

A Figura 1 resume a sequência de etapas envolvidas no *Scrum*. A Lista de Itens do Produto contém todas as funcionalidades, que são atribuídas aos *Sprints* conforme a priorização de cada uma. Esses itens são trabalhados pelo time durante o *Sprint*, que pode ter diferentes durações. Durante o *Sprint*, ocorrem as reuniões diárias de 15 minutos, e ao final do *Sprint* obtém-se as novas funcionalidades, incrementos ou bugs solucionados, conforme planejado na Lista de Itens do *Sprint*.

Figura 1 – Modelo do processo do Scrum. Fonte: Adaptado de Sommer et al. (2015)



2.2 O Desdobramento da Função Qualidade (QFD) e o *Software Quality Function Deployment* (SQFD)

Segundo Eriksson e Mcfadden (1993), o QFD consiste em uma série de atividades apoiadas por várias tabelas e matrizes. Sua ideia básica é traduzir os requisitos do cliente nos requisitos técnicos apropriados para cada estágio de produção e de desenvolvimento do produto. Cheng e Melo Filho (2007) definem o QFD como uma forma de comunicar sistematicamente informação relacionada com a qualidade e de explicitar ordenadamente trabalho relacionado com a obtenção da qualidade.

O *Software Quality Function Deployment* (SQFD) é uma adaptação do QFD para a aplicação no desenvolvimento de softwares. Para Barnett e Raja (1995), no SQFD, a ligação entre os

requisitos de alto nível do cliente e os requisitos específicos do sistema é feita de maneira formal e coerente. Existem diferentes abordagens para o SQFD, mas parece não existir uma prática única e bem estabelecida para a aplicação do QFD em softwares (KOSKI, 2003). No âmbito deste trabalho, utilizou-se o modelo proposto por Herzwurm e Schockert (PriFo SQFD), pelo foco apenas na primeira fase do QFD (i.e. Planejamento do Produto) (HERZWURM; SCHOCKERT, 2003). Segundo Herzwurm, Schockert e Tauterat (2015), o PriFo SQFD trata-se de um modelo de SQFD “Focado” (*Focused Software QFD Model*), ou seja, considera que a principal importância do QFD no desenvolvimento de um software é na definição de objetivos de desenvolvimento, no sentido das características funcionais e de qualidade baseadas nos mais importantes requisitos do cliente.

3. METODOLOGIA

O método de pesquisa adotado nesse trabalho foi o Pesquisa-Ação, voltado para a aplicação do SQFD em um software recém lançado de uma empresa de tecnologia, que utiliza o *Scrum* como metodologia de gerenciamento do desenvolvimento. Como o trabalho envolveu a aplicação de uma metodologia (SQFD) que demandou um processo constante de planejamento, ação e avaliação dos resultados (COUGHLAN, P.; COUGHLAN, D., 2002), essa estratégia metodológica foi considerada a mais adequada. O software estudado automatiza o envio de e-mails para potenciais clientes baseado em uma metodologia de vendas chamada *Outbound Marketing*.

Considerando-se os objetivos desse trabalho, as etapas propostas pelo PriFo SQFD foram adaptadas, unindo conceitos do *Scrum* e do QFD. O primeiro passo foi o Pré-Planejamento do SQFD. Esse passo envolveu a definição dos objetivos do projeto e do seu escopo. Além disso, nessa etapa foi feita a identificação dos grupos de clientes, a atribuição da importância deles e a seleção de representantes de cada grupo. Sendo assim, existem três principais tipos de clientes: (i) pessoas que possuem o processo de vendas estruturado; (ii) pessoas que possuem o processo de vendas parcialmente estruturado; e (iii) pessoas que não possuem processo de vendas estruturado.

Em seguida, na segunda etapa, foi feita a coleta, tradução e tratamento das necessidades dos clientes. Para isso, um questionário foi conduzido com 15 representantes dos três principais grupos de clientes, o qual envolveu perguntas a respeito do processo de vendas do respondente, como: “Como é feita a prospecção de clientes?”; “O processo é automatizado?”; e “O que esperam de ferramentas do tipo?”. Além disso, realizou-se duas entrevistas etnográficas (uma com uma pessoa que já utilizou a ferramenta estudada e outra com uma pessoa que nunca a utilizou). A escolha desse tipo de entrevista foi feita pois, segundo Kumar (2013), discutir experiências em seu próprio contexto pode ajudar a memória das pessoas, além de deixá-las mais confortáveis do que em um ambiente artificial ou pouco familiar.

O terceiro passo envolveu a criação da Tabela de Requisitos do Cliente e Qualidade Planejada (Figura 2). Essa tabela foi criada a partir dos Requisitos do Cliente, e recebeu contribuições de representantes de todos os grupos de clientes, que responderam um questionário para a elaboração da Qualidade Planejada. O principal objetivo da Tabela de Requisitos do Cliente e Qualidade Planejada foi confirmar junto aos clientes que exigências são mais importantes, e verificar como eles percebem ou avaliam os produtos atuais que estão no mercado (CHENG; MELO FILHO, 2007).

Na quarta etapa, elaborou-se a Tabela Voz do Engenheiro (Figura 3). Essa tabela contém as potenciais funções do software, e é elaborada pela equipe SQFD, em especial

desenvolvedores (HELPERICH; HERZWURM; SCHOCKERT, 2006). Como o principal objetivo desse trabalho era gerar uma Lista de Itens útil e já priorizada, Nessa tabela foram descritas as principais funcionalidades do software, e não suas principais funções.

Para fazer o levantamento dessas funcionalidades, foi utilizado o método de descrição de *job stories*. Christensen *et al.* (2007) sugere que as *job stories* sejam descritas na lógica: “Quando [inserir situação], quero [inserir motivação], para que [inserir resultado esperado]”. A lista de funcionalidades a partir dos Requisitos do Cliente seria viável mas, por conta do contato prévio de um dos autores com o software, esse processo poderia ser tendencioso. Sendo assim, a técnica dos *jobs-to-be-done* considera aspectos como a situação, desejos e objetivos do usuário (e não apenas algum aspecto que convém ao pesquisador), de forma que várias funcionalidades podem emergir de uma mesma *job story*.

Para a elaboração da Tabela Voz do Engenheiro foram utilizadas não só as novas funcionalidades como também as já existentes, levantadas a partir das *job stories* criadas com base nos Requisitos do Cliente. Isso foi feito para que, ao elaborar a Matriz SQFD (Figura 4), fosse avaliada não só a necessidade de criar novas funcionalidades, como também a necessidade de melhorar determinadas funcionalidades já existentes no software.

Em seguida, na quinta etapa, foi elaborada a Matriz SQFD (Figura 4), que teve como insumo a Tabela Requisitos do Cliente e Qualidade Planejada e a Tabela Voz do Engenheiro. Com ela, foi possível identificar relações de causa-efeito entre a Voz do Cliente e a Voz do Engenheiro (funcionalidades do software). Além disso, obteve-se a Qualidade Projetada para todas as funcionalidades do software da Tabela Voz do Engenheiro. Na lógica do PriFo SQFD, essa matriz deve ser criada a partir da correlação entre as Funções do Produto e Requisitos do Cliente (HERZWURM; SCHOCKERT; MELLIS, 2000). Como mencionado anteriormente, no lugar de funções do produto foram inseridas as funcionalidades do produto, levantadas na tabela da Voz do Engenheiro. Isso foi feito pois, ao final do processo, desejava-se obter uma lista de itens com as funcionalidades do produto priorizadas segundo o *Scrum*.

Com base na Matriz SQFD, na sexta etapa, organizou-se as funcionalidades do produto em ordem decrescente segundo o peso relativo de cada uma. Feito isso, atribuiu-se um peso de 1,5 a todas funcionalidades que não existiam no software, para que as funcionalidades inexistentes recebessem maior prioridade. Dessa forma, o peso final de cada funcionalidade foi recalculado (Figura 5). Para definir os pesos segundo o Pôquer de Planejamento do *Scrum*, foram definidos os pesos 1, 3, 5, 8 ou 13 para cada um dos itens, sendo que os itens que possuíam peso final acima de 4% receberam o impacto 13, os itens entre 3% e 4% receberam o impacto 8, os entre 2% e 3% receberam o impacto 5, os itens entre 1% e 2% receberam impacto 3, e os menores que 1% receberam impacto 1. Portanto, na sétima etapa, foi criada uma Lista de Itens, priorizada com o valor de Impacto (Figura 5), de forma que a priorização dos itens (das funcionalidades) partisse de uma adaptação dos valores encontrados com o SQFD para os valores usados no Pôquer de Planejamento do *Scrum*.

4. RESULTADOS

A partir das respostas dos questionários conduzidos na segunda etapa do trabalho foi possível compreender com maior riqueza de detalhes o que as pessoas esperam de uma ferramenta de automação de e-mails e o que elas gostariam que todos softwares do tipo tivessem. Além disso, as entrevistas etnográficas podem ser consideradas uma estratégia relevante para a

pesquisa, pela possibilidade de levantar observações bem diferentes das levantadas no questionário, pois as pessoas mencionaram qualidades atrativas e óbvias e não se restringiram às qualidades lineares como a maioria fez ao responder o questionário.

Na terceira etapa foi criada a Tabela de Requisitos do Cliente e Qualidade Planejada, a qual está representada na Figura 2. Para a avaliação de desempenho foram analisados o software estudado e um de seus concorrentes, o Mixmax, o qual é atualmente o concorrente mais conhecido no Brasil.

Figura 2. Tabela de Requisitos do Cliente e Qualidade Planejada. Fonte: Elaborada pelos autores

Tabela Requisitos do cliente (último nível)	Qualidade Planejada							
	Grau de importância	Avaliação de desempenho		Planejamento			Peso	
		Software estudado	Mixmax	Plano de Qualidade	Índice de melhoria	Argumento de venda	Peso absoluto	Peso relativo
Organizar e-mails na ordem de envio	4	5	5	5	1.0	1.0	4.4	2.3%
Gerenciar tarefas além de emails (ligações por exemplo)	3	2	1	5	3.1	1.5	15.9	8.5%
Enviar fluxos de cadência automatizados	5	5	4	5	1.1	1.0	5.2	2.8%
Definir quando cada email será enviado	4	4	4	5	1.2	1.0	5.0	2.7%
Enviar e-mails com tom pessoal	4	5	4	4	0.9	1.0	3.7	2.0%
Visualizar como os e-mails serão enviados antes de enviá-los	4	2	4	3	1.3	1.0	4.6	2.4%
Enviar emails do meu próprio email	5	5						
Não mandar e-mails repetidos para uma mesma pessoa								

A quarta etapa consistiu na elaboração da Tabela Voz do Engenheiro. Para fazer o levantamento dessas funcionalidades, foi utilizado o método de descrição de *job stories*. Essa técnica forçou uma reflexão acerca de todos os requisitos do cliente, funcionalidades atuais do produto e possíveis funcionalidades, ainda não existentes, de forma a compreender os motivos e desejos por trás de cada requisito do cliente. Além disso, ela foi importante para verificar se aqueles requisitos realmente faziam sentido e eram, de fato, aplicáveis. Algumas *job stories* deram origem a mais de uma funcionalidade, visto que todas elas resolveriam o *job* descrito. Para isso, cada *job story* foi analisada individualmente pela pesquisadora, e todas as funcionalidades que poderiam resolver aquele *job* foram levantadas, existentes ou não no produto.

Figura 3. Tabela de Requisitos do Cliente, *job stories* e Voz do Engenheiro. Fonte: Elaborada pelos autores

Requisitos do Cliente	Job stories	Voz do Engenheiro
Organizar e-mails na ordem de envio	Quando crio meu fluxo de cadência, quero definir em qual ordem os e-mails serão enviados, para garantir que meu lead os receberá na ordem correta do fluxo	Organizar e-mails e outras ações de um fluxo de acordo com sua ordem de envio Permitir a edição de fluxos já existentes
Gerenciar tarefas além de emails (ligações por exemplo)	Quando crio meu fluxo de cadência, quero inserir não só e-mails como também ligações e contatos via redes sociais, para que seja possível gerenciar todas as atividades do fluxo e eu não precise usar outras ferramentas ao mesmo tempo para isso.	Permitir o agendamento de ações (emails, ligações, contatos em redes sociais, etc) dentro de um fluxo Permitir o gerenciamento de ações, como ligações e mensagens em redes sociais, dentro de um fluxo
Enviar fluxos de cadência	Quando crio um fluxo de cadência, quero que ele seja disparado automaticamente conforme o planejamento.	

Na quinta etapa, foi elaborada a Matriz SQFD (Figura 4), que teve como insumo a Tabela Requisitos do Cliente e Qualidade Planejada e a Tabela Voz do Engenheiro. Diferentemente

Figura 5. Lista de itens priorizada segundo o Scrum. Fonte: Elaborada pelos autores

FUNCIONALIDADE	PESO FINAL	IMPACTO SCRUM
Permitir a definição de ações de um fluxo com base nas atitudes tomadas por cada lead	4.74%	13
Permitir a definição de ações com base na execução ou não da ação anterior do fluxo	4.58%	13
Inteligência artificial: Sugestões sobre o que funciona bem e não funciona bem nos fluxos e e-mails, no que diz respeito ao horário de envio dos e-mails, grupos de contatos, conteúdos dos e-mails, etc	4.58%	13
Envio automático de e-mails dentro de um fluxo de cadência, conforme programado por mim	4.42%	13
Inserir contatos automaticamente em um novo fluxo com base na resposta positiva ou negativa	4.10%	13
Permitir a edição de fluxos já existentes	3.68%	8
Inserir comentários nos contatos, a respeito das respostas recebidas	3.58%	8
Uso de variável para sugerir diferentes horários de reunião	3.55%	8
Permitir o gerenciamento de ações, como ligações e mensagens em redes sociais, dentro de um fluxo	3.48%	8
Organizar e-mails e outras ações de um fluxo de acordo com sua ordem de envio	3.48%	8
Categorias para organização dos fluxos	3.15%	
Filtros por data, categorias e responsáveis para os fluxos, para os contatos da base e para os templates		

5. CONCLUSÃO

O trabalho atendeu aos objetivos propostos na medida que, com a aplicação do SQFD no software avaliado, foi possível trazer o cliente de forma ativa para o levantamento e priorização das funcionalidades da Lista de Itens. Além disso, desejos e necessidades dos clientes que antes não eram levados em conta de maneira direta na empresa foram considerados.

Como o principal objetivo desse trabalho era gerar uma Lista de Itens aplicável e priorizada, na Tabela Voz do Engenheiro foram descritas as principais funcionalidades do software, e não suas principais funções, tornando a técnica de *jobs-to-be-done* essencial para o sucesso dessa etapa: ela forçou a pesquisadora a descrever as funcionalidades com os “olhos de um iniciante”, considerando aspectos como a situação, desejo e objetivos do usuário (e não apenas algum aspecto que lhe convém). Com isso, várias funcionalidades emergiram de uma mesma *job story*, dentre elas funcionalidades que a equipe de produto nunca havia considerado anteriormente. Essa técnica provocou a reflexão acerca de todos os requisitos do cliente, funcionalidades atuais do produto e outras novas vislumbradas, de forma a compreender os motivos e desejos por trás de cada requisito do cliente. Ela também foi importante para verificar se aqueles requisitos realmente faziam sentido e se eram, de fato, aplicáveis. Além disso, a definição dos pesos da Lista de Itens utilizando a sequência de Fibonacci do Pôquer de Planejamento permitiu que esses itens pudessem ser adicionados à Lista de Itens da empresa estudada, para posterior definição dos valores de Esforço pela equipe de desenvolvimento.

Como limitações do estudo, aponta-se o processo de definição das correlações da matriz SQFD. Como muitos requisitos do cliente e funcionalidades foram levantados, o estabelecimento das correlações levou alguns dias para ser finalizado, o que dificultou a

definição de padrões de comparação entre uma correlação e outra. Uma solução seria dividir esse processo em grupos com base nos tipos de funcionalidades, para torna-lo mais curto, produtivo e assertivo. Além disso, como todos os respondentes conheciam apenas uma ferramenta concorrente, o estabelecimento da Qualidade Planejada acabou se tornando limitado. Dessa forma, mais entrevistas poderiam ter sido conduzidas com pessoas de outras partes do Brasil ou até mesmo de outros países, de forma a levantar informações a respeito de outros concorrentes.

Por fim, com base nos resultados atingidos com o trabalho, surge como possibilidade de trabalhos futuros a união desses métodos de forma iterativa, ou seja, a aplicação do SQFD considerando-se Sprints de uma ou duas semanas, para a atualização constante da Lista de Itens. Ou seja, trazer a voz do cliente em tempo real para o desenvolvimento.

6. REFERÊNCIAS BIBLIOGRÁFICAS

AITKEN, A.; ILANGO, V. A Comparative Analysis of Traditional Software Engineering and Agile Software Development. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEMS ENGINEERING, 46, 2013, Hawaii. *Proceedings...* Hawaii: IEEE Computer Society, 2013, p. 4751-4760.

CARRUTHERS, D. **Software Quality Function Deployment**: a method to build better software. 1999. 86 f. Tese (Bachelor of Science with Honours – Computer Science) – Departamento de Ciência da Computação, Universidade Edith Cowan, Perth. 1999.

CHENG, L.C.; MELO FILHO, L. R. **QFD**: Desdobramento da função qualidade na gestão de desenvolvimento de produtos. 1a ed. São Paulo: Editora Blücher, 2007. 539 p.

CHRISTENSEN, C. M. *et al.* Finding the Right Job For Your Product. **MIT Sloan Management Review**, v. 48, n. 3, p. 38-47, 2007.

COUGHLAN, P.; COUGHLAN, D. Action Research for Operations Management. **International Journal of Operations and Production Management**, v. 22, n. 2, p. 220-240, 2002.

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.

HALLIGAN, Brian. **Inbound Marketing vs. Outbound Marketing**. 2010. Disponível em: <<https://blog.hubspot.com/blog/tabid/6307/bid/2989/Inbound-Marketing-vs-Outbound-Marketing.aspx>>. Acesso em: 17 Jun. 2017.

HELFERICH, A.; HERZWURM, G.; SCHOCKERT, S. The use of Quality Function Deployment (QFD) for customer-focused Product Development. In: BLECKER, T. *et al.* (Eds.). **Customer Interaction and Customer Integration**. Berlin: Gito GmbH, 2006, p. 217-231.

HERZWURM, G; SCHOCKERT, G; MELLIS, W. **Joint Requirements Engineering:QFD** for Rapid Customer Focused Software and Internet Development. 1. ed. Braunschweig – Wiesbaden: Vieweg – Gabler, 2000.

HERZWURM, G.; SCHOCKERT, S. The leading edge in QFD for software and electronic business. **International Journal of Quality & Reliability Management**, [S.I.], v. 20, n. 1, p. 36-55, 2003.

HERZWURM, G.; SCHOCKERT, S.; TAUTERAT, T. Quality Function Deployment in software development: State-of-the-art. In: INTERNATIONAL SYMPOSIUM ON QFD, 21, 2015, Hangzhou. *Anais...* Hangzhou: QFD Institute, 2015.

KOSKI, J. **Quality Function Deployment in Requirements Engineering: A Review and Case Studies**. 2003. 108 f. Tese (MBA em Software Product Business) - Executive School of Business, Helsinki University of Technology, Espoo, 2003.

KUMAR, V. **101 Design Methods**. Hoboken: John Wiley & Sons, Inc., 2013. 325 p.

PERMANA, P. A. G. *Scrum* Method Implementation in a Software Development Project Management. **International Journal of Advanced Computer Science and Applications**, Bali, v. 6, n. 9, p. 198-204, 2015.

BARNETT, W. D.; RAJA, M. K. Application of QFD to the software development process. **International Journal of Quality & Reliability Management**, [Arlington], v.12, n. 6, p. 24-42, 1995.

SOMMER, A. F. *et al.* Improved Product Development Performance through Agile/Stage-Gate Hybrids. **Research-Technology Management**, Arlington, p.34-44, 2015.

STOICA, M.; MIRCEA, M.; GHILIC-MICU, B. Software Development: Agile vs. Traditional. **Informatica Economică**, Romania, v. 17, n. 4, p. 64-74, 2013.

SUTHERLAND, J. **Scrum: A arte de fazer o dobro do trabalho em metade do tempo**. São Paulo: Leya, 2014. 180 p.

VLAANDEREN, K. *et al.* The agile requirements refinery: Applying *SCRUM* principles to software product management. **Information and Software Technology**, Utrecht, v. 53, n. 1, p. 58-70, 2011.