

**Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Curso de Engenharia de Software**

**PROPOSTA DE UM CONJUNTO DE COMPETÊNCIAS
PARA UM *PRODUCT OWNER***

**Autor: Vinicius Maia de Brito
Orientador: Msc. George Marsicano Corrêa**

**Brasília, DF
2014**



Vinicius Maia de Brito

PROPOSTA DE UM CONJUNTO DE COMPETÊNCIAS PARA UM *PRODUCT OWNER*

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Orientador: Msc. George Marsicano Corrêa

Co-Orientador: Dr. Wander Cleber Maria Pereira da Silva

**Brasília, DF
2014**

CIP – Catalogação Internacional da Publicação*

Brito, Vinicius Maia de.

Proposta de um Conjunto de Competências Para um *Product Owner* / Vinicius Maia de Brito (em ordem normal). Brasília: UnB, 2014. 103 p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília
Faculdade do Gama, Brasília, 2014. Orientação: George
Marsicano Corrêa.

1. *Product Owner*. 2. Competência. 3. *Scrum* I. Corrêa, George
Marsicano. II. Msc.

CDU Classificação

- A ficha catalográfica oficial deverá ser solicitada à Biblioteca pelo aluno após a apresentação.



PROPOSTA DE UM CONJUNTO DE COMPETÊNCIAS PARA UM *PRODUCT OWNER*

Vinicius Maia de Brito

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software da Faculdade UnB Gama - FGA, da Universidade de Brasília, em 24/06/2014, apresentada e aprovada pela banca examinadora abaixo assinada:

Prof. Msc. George Marsicano Corrêa, UnB/ FGA
Orientador

Prof. Dr. Wander C. M. Pereira da Silva, UnB/ FGA
Co-Orientador

Profa. Dra. Rejane Maria da Costa Figueiredo, UnB/ FGA
Membro Convidado

Profa. Msc. Elaine Venson, UnB/ FGA
Membro Convidado

Brasília, DF
2014

AGRADECIMENTOS

Agradeço primeiro a Deus por me sustentar durante toda a minha vida, durante o curso, sobretudo nas horas mais difíceis, por me manter perseverante, por me ajudar a vencer os momentos mais complicados, apesar das minhas constantes falhas. Agradeço, também, a minha família que sempre me apoiou e meu deus todas as condições para que eu aprendesse o máximo durante o curso de Engenharia e o aproveitasse da melhor forma possível. Agradeço aos meus amigos, sobretudo minha namorada, que sempre esteve ao meu lado, me ajudando em todos os sentidos, em todos os momentos.

Agradeço meus professores por me proporcionarem momentos únicos durante minha vida acadêmica e colaborarem no meu desenvolvimento como pessoa e profissional, em especial, aos professores Dr. Edson Alves da Costa Júnior e Roque Martins, por serem dois dos maiores responsáveis por eu chegar a esse momento.

Agradeço, ao meu orientador, professor Msc. George Marsicano Corrêa, e ao meu co-orientador, professor Dr. Wander Cleber M. P. da Silva, que colaboraram no desenvolvimento deste trabalho e souberam conduzir-me da melhor forma possível desde o início da sua construção.

RESUMO

A importância da área de TI cresce cada vez mais tanto no contexto público quanto no privado. A partir da divulgação do Manifesto Ágil em 2001, juntamente com esse crescimento, há a disseminação da utilização de metodologias ágeis em ambos os contextos, sobretudo do *framework Scrum*, que traz nova logística, atividades e papéis para o desenvolvimento de um software. Um desses novos papéis é o *Product Owner* (PO), que possui diversas responsabilidades. Porém, para que ele consiga executar suas atribuições com êxito, é necessário que a pessoa que desempenha esse papel possua competências específicas. Este Trabalho de Conclusão de Curso possui como objetivo propor e validar um conjunto de competências iniciais para um *Product Owner* que esteja alinhado com as responsabilidades que este papel possui e as atividades que desempenha de modo a potencializar os fatores de sucesso e minimizar os fatores de fracasso de um projeto referentes ao PO. Um questionário foi elaborado para efetuar a validação das competências e produziu como resultado um ranqueamento das competências com base na importância dada pelos respondentes a cada uma delas.

Palavras-chave: *Product Owner*. Competência. *Scrum*.

ABSTRACT

The importance of IT grows increasingly both in the public and private context. From the release of the Agile Manifesto in 2001, along with this growth, occurs the spread of the use of agile methodologies in both contexts, especially the Scrum framework, which brings new logistics, activities and roles for the software development. One of these roles is the Product Owner (PO), who has many responsibilities. However, so the PO can execute his/her tasks successfully it is necessary that the person who performs this role has specific competences. The objective of this Work of Course's Conclusion is to propose and validate a set of preliminary skills for a Product Owner that is aligned with the responsibilities that this role has and the activities it performs, in order to maximize the success factors and minimize the failure ones of a project related to the PO. A questionnaire was designed to perform the validation of the skills and produced results in a ranking of competences based on the importance given by the respondents for each of them.

Keywords: Product Owner. Competences. Scrum.

LISTA DE FIGURAS

Figura 1. Metodologia de pesquisa (O autor)	17
Figura 2. Scrum (SUTHERLAND, 2011)	21
Figura 3. Sprint (Mountain Goat Software, 2005)	22
Figura 4. Competências como Fonte de Valor para o Indivíduo e para a Organização (FLEURY, 2001)	33
Figura 5. Estágios principais de um Survey (SCHUMAN & KALTON, 1985)	41
Figura 6. Escala utilizada no questionário (O autor)	42
Figura 7. Curva da distribuição normal (MORETTIN, 2009)	59
Figura 8. Distribuição dos respondentes por contexto: graus 1 e 2 de importância – CT10 (O autor) .	62
Figura 9. Distribuição dos respondentes por contexto: graus 4 e 5 de importância – CT10 (O autor) .	63
Figura 10. Distribuição dos respondentes por contexto: graus 1 e 2 de importância – CT11 (O autor).64	
Figura 11. Distribuição dos respondentes por contexto: graus 4 e 5 de importância – CT11 (O autor).65	
Figura 12. Distribuição dos respondentes por contexto: graus 1 e 2 de importância – CT12 (O autor).66	
Figura 13. Distribuição dos respondentes por contexto: graus 4 e 5 de importância – CT12 (O autor).67	
Figura 14. Distribuição dos respondentes por contexto: graus 1 e 2 de importância – CT13 (O autor).68	
Figura 15. Distribuição dos respondentes por contexto: graus 4 e 5 de importância – CT13 (O autor).69	

LISTA DE GRÁFICOS

Gráfico 1. Trabalhos coletados por fonte (O autor).....	36
Gráfico 2. Etapa de Extração dos documentos (O autor)	37
Gráfico 3. Caracterização do <i>Product Owner</i> para os artigos aceitos (O autor)	38
Gráfico 4. Possuir conhecimento das tecnologias no desenvolvimento de software (O autor).....	61
Gráfico 5. Possuir conhecimento do impacto de teste funcional no projeto (O autor).....	63
Gráfico 6. Possuir conhecimento do impacto da refatoração no projeto (O autor).....	65
Gráfico 7. Possuir conhecimento dos ritos e cerimônias (O autor).....	67
Gráfico 8. Resultado da CT1 (O autor)	108
Gráfico 9. Resultado da CT2 (O autor)	108
Gráfico 10. Resultado da CT3 (O autor)	108
Gráfico 11. Resultado da CT4 (O autor)	109
Gráfico 12. Resultado da CT5 (O autor)	109
Gráfico 13. Resultado da CT6 (O autor)	109
Gráfico 14. Resultado da CT7 (O autor)	110
Gráfico 15. Resultado da CT8 (O autor)	110
Gráfico 16. Resultado da CT9 (O autor)	110
Gráfico 17. Resultado da CT10 (O autor)	111
Gráfico 18. Resultado da CT11 (O autor)	111
Gráfico 19. Resultado da CT12 (O autor)	111
Gráfico 20. Resultado da CT13 (O autor)	112
Gráfico 21. Resultado da CT14 (O autor)	112
Gráfico 22. Resultado da CT15 (O autor)	112
Gráfico 23. Resultado da CT16 (O autor)	113
Gráfico 24. Resultado da CC1 (O autor).....	113
Gráfico 25. Resultado da CC2 (O autor).....	113
Gráfico 26. Resultado da CC3 (O autor).....	114
Gráfico 27. Resultado da CC4 (O autor).....	114
Gráfico 28. Resultado da CC5 (O autor).....	114
Gráfico 29. Resultado da CC6 (O autor).....	115
Gráfico 30. Sexo do respondente (O autor)	115
Gráfico 31. Atuação como PO (O autor)	115
Gráfico 32. Caráter da experiência como PO (O autor).....	116
Gráfico 33. Outros papéis desempenhados (O autor)	116
Gráfico 34. Contexto da experiência com <i>Scrum</i> (O autor)	116
Gráfico 35. Área de atuação do respondente (O autor).....	117
Gráfico 36. Continuidade na utilização do <i>Scrum</i> (O autor)	117

LISTA DE QUADROS

Quadro 1. Resumo do papel do <i>Product Owner</i> (O autor)	30
Quadro 2. Resumo de competências (adaptado de Leme (2006)).....	34
Quadro 3. Resumo dos atributos encontrados nos artigos (O autor)	39
Quadro 4. Variação da confiabilidade pelo valor de Alfa (adaptado de Freitas (2005))	49
Quadro 5. Caracterização dos respondentes (O autor).....	51
Quadro 6. Ranking das competências (O autor).....	70
Quadro 7. Fundamentação dos atributos por artigo (O autor)	83
Quadro 8. Competências para um <i>Product Owner</i> (O autor)	91
Quadro 9. Ranking das competências técnicas (O autor)	118
Quadro 10. Ranking das competências comportamentais (O autor).....	119

SUMÁRIO

AGRADECIMENTOS	5
RESUMO	6
ABSTRACT	7
LISTA DE FIGURAS	8
LISTA DE GRÁFICOS	9
LISTA DE QUADROS	10
SUMÁRIO	11
1. INTRODUÇÃO	13
1.1. CONTEXTO	13
1.1.1. Metodologias ágeis na Engenharia de Software	13
1.1.2. Riscos na contratação de desenvolvimento ágil	14
1.1.2.1. Riscos relacionados a pessoas	14
1.2. PROBLEMA	16
1.3. OBJETIVOS	16
1.3.1. Objetivo geral.....	16
1.3.2. Objetivos específicos.....	16
1.4. JUSTIFICATIVAS.....	17
1.5. METODOLOGIA DE PESQUISA.....	17
1.6. ORGANIZAÇÃO DO TRABALHO.....	17
2. REFERENCIAL TEÓRICO	19
2.1. METODOLOGIAS ÁGEIS	19
2.1.1. Scrum	20
2.1.1.2. Sprints	21
2.1.1.2.1. <i>Daily scrum meeting</i>	23
2.1.1.2.2. <i>Planejamento da sprint</i>	23
2.1.1.2.3. <i>Sprint review meeting</i>	24
2.1.1.2.4. <i>Retrospectiva da sprint</i>	24
2.1.1.3. Papéis	25
2.1.1.3.1. <i>ScrumMaster</i>	25
2.1.1.3.2. <i>Stakeholders</i>	25
2.1.1.3.3. <i>Equipe de desenvolvimento</i>	26
2.1.1.3.4. <i>Product Owner</i>	26
2.2. COMPETÊNCIA.....	32
3. METODOLOGIA DE PESQUISA	35
3.1. CLASSIFICAÇÃO DA PESQUISA.....	35
3.2. REVISÃO SISTEMÁTICA	36
3.3. INSTRUMENTO DE PESQUISA	40
3.3.1. Elaboração do instrumento de pesquisa	40
3.3.2. Validação do instrumento	42
3.3.2.1. Academia	43
3.3.2.2. Setor público	45
3.3.2.3. Institutos de pesquisa e desenvolvimento	45
3.3.3. Ajustes do instrumento de pesquisa.....	46
3.3.4. Aplicação do instrumento de pesquisa.....	48
3.3.5. Avaliação do questionário	49
3.3.5.1. Consistência do questionário.....	49
3.3.5.2. Representatividade dos resultados	49
4. APRESENTAÇÃO DOS RESULTADOS	51
4.1. CARACTERIZAÇÃO DA POPULAÇÃO RESPONDENTE.....	51
4.2. COMPETÊNCIAS TÉCNICAS	53
4.2.1. Dominar técnicas de planejamento de desenvolvimento ágil	53
4.2.2. Dominar o negócio em que o projeto está inserido.....	53
4.2.3. Dominar técnicas de priorização do product backlog.....	53
4.2.4. Dominar a técnica de escrita de user story	53
4.2.5. Dominar a técnica de validação de requisitos	54
4.2.6. Dominar as técnicas de monitoramento do ROI	54

4.2.7. Dominar as técnicas de comunicação.....	54
4.2.8. Possuir visão estratégica.....	54
4.2.9. Possuir visão sistêmica	54
4.2.10. Possuir conhecimento das tecnologias utilizadas no desenvolvimento do software	55
4.2.11. Possuir conhecimento do impacto de teste funcional no projeto	55
4.2.12. Possuir conhecimento do impacto da refatoração no projeto	55
4.2.13. Possuir conhecimento dos ritos e cerimônias realizadas no Scrum	55
4.2.14. Possuir o conhecimento dos valores ágeis	55
4.2.15. Possuir o conhecimento dos princípios ágeis	56
4.2.16. Dominar técnicas de negociação	56
4.3. COMPETÊNCIAS COMPORTAMENTAIS	56
4.3.1. Possuir a coragem para tomar uma decisão.....	56
4.3.2. Possuir a capacidade de admitir quando está errado	57
4.3.3. Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que se relaciona	57
4.3.4. Possuir a disposição e a habilidade em ajudar a equipe	57
4.3.5. Possuir a habilidade de se adequar a diferentes contextos.....	57
4.3.6. Possuir a habilidade de resolver conflitos	58
4.4. CONSIDERAÇÕES DOS RESPONDENTES.....	58
4.5. ANÁLISE DOS RESULTADOS.....	58
4.5.1. Critérios utilizados na seleção dos itens	58
4.5.2. Análise dos dados	60
4.5.2.1. Possuir conhecimento das tecnologias utilizadas no desenvolvimento de software.....	61
4.5.2.2. Possuir conhecimento do impacto de teste funcional no projeto	63
4.5.2.3. Possuir conhecimento do impacto da refatoração no projeto	65
4.5.2.4. Possuir conhecimento dos ritos e cerimônias realizadas no scrum	67
4.6. RANKING DAS COMPETÊNCIAS	69
5. CONCLUSÕES E TRABALHOS FUTUROS	72
5.1. CONCLUSÕES.....	72
5.2. TRABALHOS FUTUROS.....	73
APÊNDICE I – PROTOCOLO DA REVISÃO SISTEMÁTICA.....	80
APÊNDICE II – FUNDAMENTAÇÃO DOS ATRIBUTOS POR ARTIGO	83
APÊNDICE III – CONJUNTO INICIAL DE COMPETÊNCIAS	91
APÊNDICE IV – PRIMEIRA VERSÃO DO QUESTIONÁRIO	96
APÊNDICE V – SEGUNDA VERSÃO DO QUESTIONÁRIO	101
APÊNDICE VI – RESULTADOS DO QUESTIONÁRIO	108
APÊNDICE VII – RANKING DAS COMPETÊNCIAS TÉCNICAS.....	118
APÊNDICE VIII – RANKING DAS COMPETÊNCIAS COMPORTAMENTAIS	119

1. INTRODUÇÃO

1.1. CONTEXTO

1.1.1. Metodologias ágeis na Engenharia de Software

As áreas de TI exercem uma importância cada vez maior nas organizações públicas e privadas devido ao aumento da dependência, desde a disponibilização de serviços virtuais estatais à população, passando pelo monitoramento de informações, até a interligação de sistemas da administração pública.

Em 2001, houve a publicação do Manifesto Ágil, conjunto de princípios que fundamentam o desenvolvimento ágil de software, e, a partir daí, a criação de algumas metodologias ágeis, tanto de desenvolvimento quanto de gerenciamento de projetos. Essas metodologias baseadas no Manifesto, que possui valores distintos dos estabelecidos pelo processo unificado, ganharam popularidade no mercado internacional e nacional com o passar dos anos.

A popularidade das metodologias ágeis se alastrou primeiramente no contexto de instituições privadas, como apresenta James Shore (2008), exemplos de importantes empresas no âmbito internacional, como Google, Microsoft e Symantec. Após esse acontecimento, órgãos públicos passaram a se atentar às metodologias ágeis, como é o caso do Brasil, chegando utilizá-las em processos licitatórios, como transcrito no art. 6 do Acórdão 2314/2013 do Tribunal de Contas da União: “Essa realidade, somada às insatisfações frequentes das contratações de serviços geradas pelo uso do modelo corrente, tem levado algumas instituições públicas a acreditarem que podem obter melhores resultados com o uso das metodologias ágeis. (...) algumas dessas instituições começaram a realizar contratações para desenvolvimento de software” (AC 2314-33/2013).

Alguns exemplos de instituições, citadas no estudo publicado nesse Acórdão, que licitaram esse tipo de contrato foram o Banco Central (Bacen), Tribunal Superior do Trabalho (TST), Instituto do Patrimônio Histórico e Artístico Nacional (Iphan), Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep) e Supremo Tribunal Federal (STF).

1.1.2. Riscos na contratação de desenvolvimento ágil

A mudança de abordagem dos termos de referência voltados para a contratação de desenvolvimento de software pelos órgãos públicos traz consigo diversos riscos de diferentes naturezas: pessoas, processos, produtos, entre outros. Alguns desses riscos foram levantados no estudo realizado pela Secretaria de Fiscalização de Tecnologia da Informação (Sefti) do TCU, que “identificou instituições públicas que possuem contratos cujo objeto é de interesse desta fiscalização e visitou alguma delas”.

Nesse contexto, um “risco é um evento ou condição incerta que, se ocorrer, tem um efeito positivo ou negativo sobre ao menos um dos objetivos do projeto.” (PMI, 2004).

Apesar de alguns desses riscos advirem da “pouca experiência da instituição pública contratante na utilização de métodos ágeis” (AC 2314-33/2013), boa parte deles são inerentes da própria metodologia, como, por exemplo, o Risco 11, relativo a produto, presente no art. 282 do mesmo Acórdão: “alteração constante da lista de funcionalidades do produto”, que vai ao encontro com o princípio do Manifesto Ágil que diz que “mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento”.

1.1.2.1. Riscos relacionados a pessoas

No artigo “*Why Software Fails*”, Charette (2005) apresenta doze fatores para o fracasso de um projeto de software, sendo que oito deles se encontram na relação do negócio com o desenvolvimento. Dentre esses fatores, destacam-se:

- Metas do projeto irrealistas ou desarticuladas;
- Falta de comunicação entre os clientes, desenvolvedores e usuários;

Levando em consideração os valores ágeis que determinam que “indivíduos e interações mais que processos e ferramentas” e “colaboração com o cliente mais que negociação de contratos”, percebe-se que o relacionamento entre pessoas, tanto internamente na equipe, quanto externamente com o cliente, é primordial para assegurar que o desenvolvimento ágil aconteça.

Desse modo, há uma relevância inerente aos riscos relacionados a pessoas na contratação de desenvolvimento ágil por órgãos da administração pública.

Essa categoria de riscos foi abordada pelo estudo da Sefti e, por meio deste, foram identificados cinco deles, sendo dois materializados em contratos de instituições estudadas pela secretaria. Os riscos identificados foram:

- Falta de comprometimento ou colaboração insatisfatória do responsável indicado pela área de negócios (*Product Owner*) no desenvolvimento do software;
- Falta do conhecimento necessário do indicado pela área de negócios (*Product Owner*) para o desenvolvimento do software;
- Excessiva dependência da visão do indicado pela área de negócios (*Product Owner*);
- Equipe da empresa contratada não ter *expertise* em desenvolvimento de software com métodos ágeis;
- Dificuldade de comunicação entre a equipe de desenvolvimento da contratada com o indicado pela área de negócios (*Product Owner*).

Destaca-se que dos cinco riscos levantados pelo estudo, quatro estão atrelados a um mesmo personagem: o *Product Owner*. Esse papel específico faz parte de um *framework* denominado *Scrum*, cujas fases, logística, papéis e respectivas responsabilidades serão explicitadas na Seção (2) deste trabalho.

Caso esses riscos ligados ao *Product Owner* sejam materializados, como acontecem em algumas instituições públicas estudadas pelos servidores da Sefti do TCU, há um desencadeamento de problemas relacionados ao produto, como, por exemplo, a construção de um software que não atende às necessidades do cliente, caso haja falta de comprometimento, conhecimento necessário, dificuldade de comunicação, e/ou excessiva dependência do PO.

De acordo com Machado (2002) e Charette (2005), dentre outros fatores bastante comuns que podem acarretar na falha de projetos estão:

- Estimativas imprecisas dos recursos necessários;
- Definições ruins dos requisitos do sistema;
- Falta de monitoramento de riscos.

Esses fatores estão relacionados com as atividades que o *Product Owner* executa e com as responsabilidades que compete ao PO, dessa forma, um projeto pode fracassar caso o *Product Owner* não execute suas atividades e cumpra com suas responsabilidades com êxito.

1.2. PROBLEMA

As obras de Cohn (2009), Leffingwell (2011), Sutherland e Schwaber (2011) e, até mesmo, o Guia Scrum, também de Sutherland e Schwaber, apresentam a definição do papel de *Product Owner*, assim como suas responsabilidades, características e atributos, porém não explicitam as competências que uma pessoa deve possuir para desempenhar o papel com êxito.

Decorrente da observação dessa lacuna na literatura, com o levantamento de quatro riscos relacionados a esse papel pelo estudo da Secretaria de Fiscalização de TI do Tribunal de Contas da União, sendo que dois desses riscos se materializaram em projetos de instituições públicas, e a verificação feita nas obras de Charette (2005) e Machado (2002), no contexto apresentado, a questão de pesquisa deste trabalho é:

Quais as competências de um *Product Owner* para que ele desempenhe seu papel da melhor forma possível, cumprindo com suas responsabilidades, com a finalidade de mitigar os riscos relacionados a ele no desenvolvimento de software que utiliza metodologias ágeis?

1.3. OBJETIVOS

1.3.1. Objetivo geral

O principal objetivo desse trabalho é propor um conjunto inicial de competências para um *Product Owner*.

1.3.2. Objetivos específicos

- Caracterizar as metodologias ágeis utilizadas na Engenharia de Software, em especial, o *framework Scrum*;
- Caracterizar o papel de um *Product Owner* no desenvolvimento de software que utiliza o *framework Scrum*;
- Identificar características de um *Product Owner* nas principais referências encontradas na literatura;
- Mapear os relacionamentos, atividades e responsabilidades de um *Product Owner*;
- Caracterizar o termo “competência”.

1.4. JUSTIFICATIVAS

Com a verificação dos fatores que levam a um software fracassar, apresentados por Charette (2005) e os riscos levantados pelo estudo da Sefti do TCU, sendo alguns presentes nos estudos de caso de alguns órgãos, presentes no Acórdão 2314-33/2013, observou-se a importância deste trabalho no que se refere a estabelecer um conjunto de competências para um *Product Owner*, uma vez que não foi encontrado na literatura, no escopo em que foi feita a pesquisa, a definição para esse papel proveniente do *framework Scrum*.

Uma das principais contribuições deste trabalho é auxiliar na construção de um perfil para um *Product Owner*, guiando a pessoa que delegará esse papel e escolher o profissional que melhor se encaixa com base em suas competências.

1.5. METODOLOGIA DE PESQUISA

Este trabalho foi desenvolvido seguindo a metodologia apresentada na Figura (1).

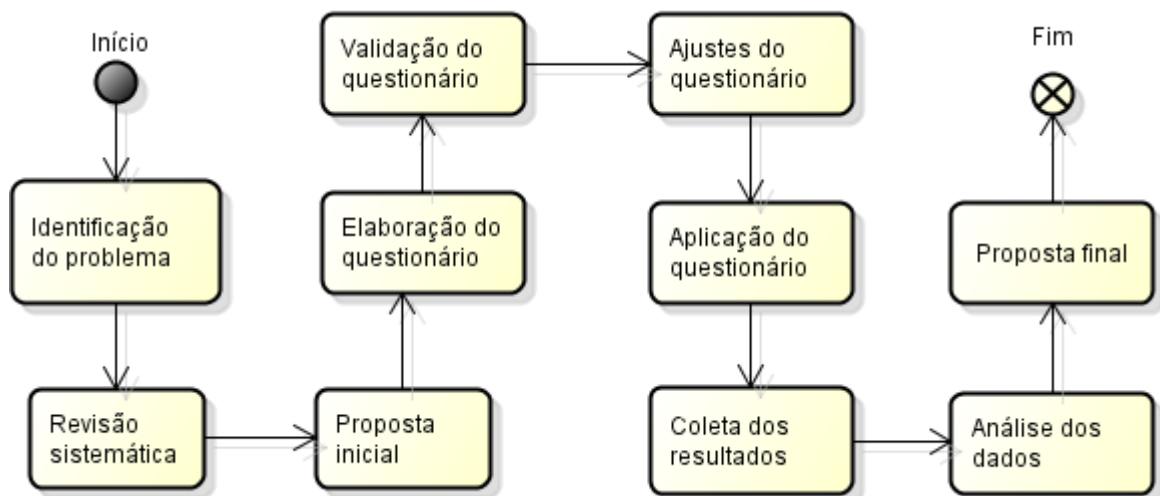


Figura 1. Metodologia de pesquisa (O autor).

Os detalhes quanto à definição e desenvolvimento da metodologia proposta estão descritos na Seção (3).

1.6. ORGANIZAÇÃO DO TRABALHO

Este documento está organizado em cinco capítulos. Neste primeiro capítulo está descrita a introdução do trabalho formado pelo contexto em que o trabalho está

inserido, o problema a ser resolvido, os objetivos vislumbrados, as justificativas e a metodologia de pesquisa adotada.

No segundo capítulo são apresentados os referenciais teóricos que dão embasamento ao trabalho. Nessa parte são referenciadas as metodologias ágeis de um modo geral, o *framework Scrum* com suas principais reuniões, seus papéis, sobretudo o *Product Owner* e, por fim, conceitos e definições sobre competência.

No terceiro capítulo há a disposição da proposta de um conjunto inicial de competências para o *Product Owner*, com os detalhes de como foi construído e de que forma se relaciona com as atividades e responsabilidades do PO.

O quarto capítulo contém os detalhes referentes ao instrumento de pesquisa construído para a avaliação do conjunto de competências proposto, os resultados obtidos com a aplicação deste instrumento e a análise dos dados.

O quinto e último capítulo apresentam as conclusões derivadas de todo o trabalho e os trabalhos futuros sugeridos para a continuação do estudo acerca do assunto abordado.

2. REFERENCIAL TEÓRICO

2.1. METODOLOGIAS ÁGEIS

O Guia das Áreas de Conhecimento de Engenharia de Software diz que, recentemente, metodologias ágeis têm promovido processos leves (SWEBOK, 2004). Os métodos ágeis chegaram à Engenharia de Software com o objetivo de solucionar as limitações dos modelos tradicionais de desenvolvimento de software, dadas as fragilidades conhecidas (NERUR, 2005).

Segundo Boehm (2002), *“organizações devem evoluir cuidadosamente para o melhor equilíbrio entre métodos ágeis e métodos tradicionais (plan-driven) que se alinha às suas situações”*. O ideal para uma instituição é possuir seu próprio processo, desenhado de acordo com as suas características para que, dessa forma, o processo não seja um gargalo no desenvolvimento. Porém, não é tão simples desenhar o próprio processo, pois demanda uma maturidade da organização como um todo. Nesse contexto, existem duas vertentes que servem de alicerce para os variados processos que surgiram: metodologias tradicionais e ágeis (BOEHM, 2002).

Esses métodos, independente da sua aplicação, gerencial ou técnica, se baseiam nos seguintes valores defendidos pelo Manifesto Ágil (BECK et al., 2001):

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

A partir desses valores foram derivados doze princípios do desenvolvimento ágil de software que expõem a importância das pessoas e a interação entre elas na metodologia, como, por exemplo (BECK et. al., 2001):

- Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

Esses princípios direcionam equipes que desenvolvem software com base nas metodologias oriundas desse Manifesto com o objetivo de satisfazer o cliente

com entregas rápidas e frequentes de software que possuam valor de negócio e que possa ser utilizado por ele para testes de aceitação e validação dos requisitos.

Segundo Cohn (2009), “*mudanças estão aparecendo mais rapidamente do que nunca*”. Com relação a elas, as metodologias ágeis vêm para suportar a volatilidade de tecnologias e do mercado, às vontades do consumidor e, conseqüentemente, dando ao cliente a possibilidade de ser competitivo no mercado, por possuírem maior *time to market* devido à maior produtividade de times ágeis que os possibilitam a produzir funcionalidades mais rapidamente e a probabilidade de liberar *releases* de maneira incremental.

Neste documento, será abordado um dos *frameworks* de desenvolvimento de projetos ágeis mais difundidos: o *Scrum*. Neste trabalho o objetivo é levantar as competências de um dos papéis presentes nele, o *Product Owner*. Serão elicitados os valores ágeis no detalhamento desse *framework*, apresentando a forma com que eles ajudam a equipe à alcançar a satisfação do cliente.

2.1.1. Scrum

Um processo que “*trava todas as suas práticas em um esqueleto de um processo iterativo, incremental*” (SCHWABER, 2004). “*O Scrum é um framework iterativo, incremental para desenvolvimento de projetos e produto ou aplicação.*” (SUTHERLAND, 2011).

Como explícito na Figura (2), o *Scrum* estrutura o desenvolvimento em pequenos ciclos de trabalho chamados de *sprints*, define papéis específicos, como o *Product Owner* e o *Scrum Master*, e artefatos a serem gerados, como o *Product Backlog*.

SCRUM

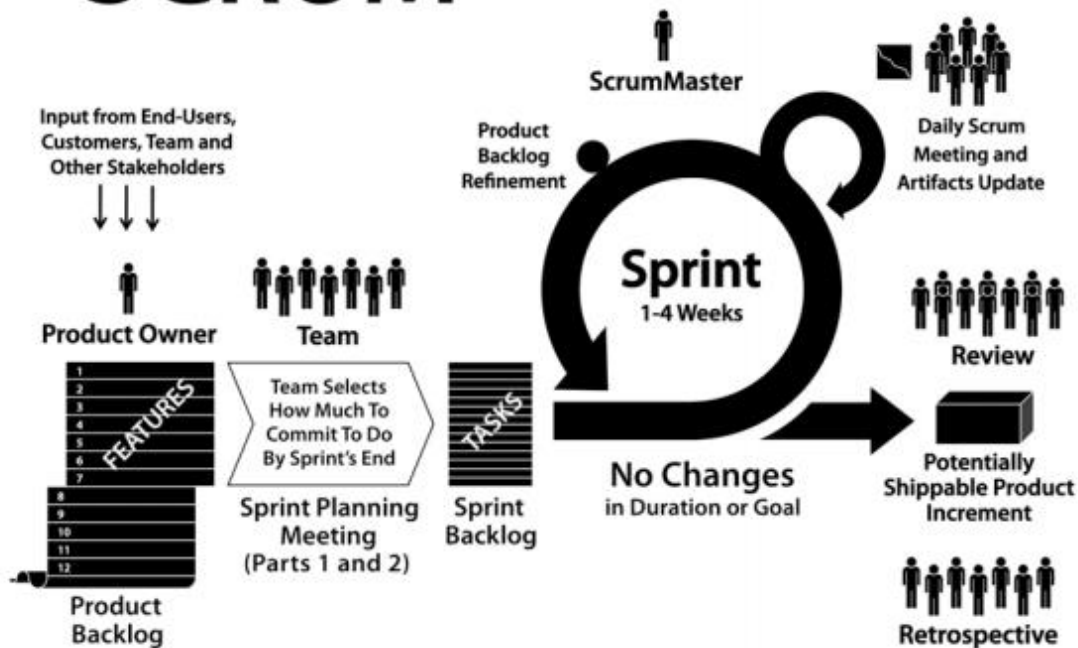


Figura 2. Scrum (SUTHERLAND, 2011).

2.1.1.2. Sprints

Seguindo os valores ágeis, o *Scrum* possui uma dinâmica que se baseia em pequenos ciclos, chamados de *sprints*. Esses ciclos devem durar curtos espaços de tempo, segundo Cohn (2009) e Sutherland (2011), 30 dias consecutivos, para que o planejamento seja feito para uma pequena fatia do projeto, sejam elicitados apenas os requisitos que devem ser implementados naquele período e, conseqüentemente, reduzindo a possibilidade de atrasos na entrega. Esse padrão adotado remete diretamente a princípios do Manifesto Ágil, uma vez que força entregas de software funcionando para o cliente constantemente com valor, o que leva a satisfação do mesmo (SCHWABER, 2004). Uma *sprint* pode ser apresentada de uma maneira simplista como na Figura (3).

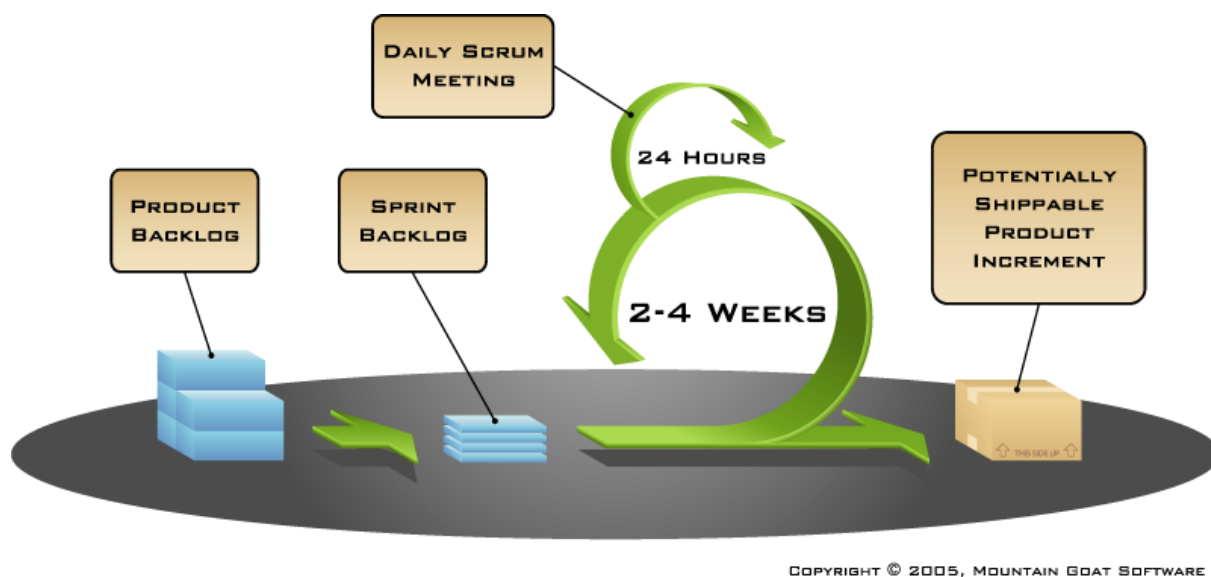


Figura 3. Sprint (Mountain Goat Software, 2005). Disponível em: <http://www.mountaingoatsoftware.com/>. Acesso em: 10 out. 2013.

O círculo inferior representa a iteração das atividades de desenvolvimento que ocorrem uma após a outra. Essa iteração possui como insumo as “*descrições de funcionalidades em alto nível (...) que serão refinadas com o progresso do projeto*” pelo *Product Owner*, chamados no Scrum de *user stories*, que são documentadas em uma lista, chamada de *backlog*, podendo ser subdivida em duas categorias (COHN, 2009):

- *Product Backlog*, que abrange todas as *user stories* do produto;
- *Sprint Backlog*, que engloba as *user stories* que devem ser implementadas na *sprint* em questão: essas são oriundas do *Product Backlog* e selecionadas a partir de uma priorização feita pelo *Product Owner*.

O *output* dessa iteração são as funcionalidades devidamente implementadas e testadas, empacotadas num artefato, chamado de *Release*, que é disponibilizado para o uso do cliente (SUTHERLAND, 2011). A fim de validar essas *user stories*, são criadas, juntamente com elas, os Testes de Aceitação para cada uma: esses são, basicamente, os critérios que, aplicados às funcionalidades que foram implementadas acerca daquela *user story*, indicam se ela foi contemplada por completo (COHN, 2004).

Já o círculo superior da Figura (3) representa a reunião diária, chamada de *Daily Scrum Meeting*, executada em pé todos os dias durante a iteração e antes de começar o expediente.

2.1.1.2.1. *Daily scrum meeting*

Durante a *sprint*, todos os dias acontecem reuniões denominadas *Daily Scrum Meeting*, ou até *Standup Meeting*, já que são executadas em pé. Antes do expediente começar, cada um dos membros da equipe se juntam para inspecionar as atividades dos outros e colaborar com ideias para otimizá-las. Essa reunião deve demorar cerca de quinze minutos, às vezes ultrapassando um pouco esse prazo, porém não podendo ultrapassar o tempo máximo de 30 minutos. Todos da equipe devem participar dessa reunião, em que o líder do time, chamado de *Scrum Master*, deve fazer as seguintes três perguntas a cada um (RISING, 2002) (SUTHERLAND, 2011):

- Das suas atividades, o que você fez desde a última reunião?
- Quais obstáculos você encontrou ao tentar completar essas atividades?
- O que você planeja fazer até a próxima reunião?

Contudo, para se iniciar uma *sprint*, deve-se possuir o *Product Backlog* com as *user stories* devidamente priorizadas pelo *Product Owner* de acordo com seu interesse de implementação. Com esse artefato em mãos, dá-se início a uma reunião de planejamento, chamado de *Sprint Planning Meeting* (SUTHERLAND, 2011).

2.1.1.2.2. *Planejamento da sprint*

Segundo Schwaber (2004), o *Sprint Planning Meeting* acontece sempre antes de se iniciar uma *sprint* e tem seu tempo máximo estipulado em 8 horas, dividido em duas partes de 4 horas cada. A primeira tem o objetivo de selecionar o *Product Backlog*, atualizando-o de acordo com a visão de prioridade do cliente e com o que já fora implementado; já a segunda fase é utilizada para preparar o *Sprint Backlog*, que devem conter requisitos selecionados a partir do *Product Backlog* a partir de uma priorização que depende da vontade do *Product Owner*, que normalmente escolhe aqueles requisitos que ele enxerga mais valor pro negócio. Essa priorização, no início do projeto, pode sofrer interferência do *Scrum Master*, se este que enxergar a necessidade de implementar *user stories* que exigem a construção da arquitetura completa do software, ou seja, que precise passar por todas as camadas.

Após a escolha das *user stories*, é feita uma quebra delas em tarefas que o time consiga estipular o tempo necessário para a completude de cada uma, reduzindo, assim, distorções nas estimativas de tempo para a execução delas. Para

cada tarefa um ou mais responsáveis pela sua execução que darão o *feedback* do andamento da mesma nas reuniões diárias (SUTHERLAND, 2011).

Quando o tempo da *sprint* se esgota, chega a hora do time entregar o produto do trabalho realizado ao *Product Owner*. Esse momento é chamado, no *Scrum*, de *Sprint Review Meeting* (SCHWABER, 2001).

2.1.1.2.3. *Sprint review meeting*

O fim de uma *sprint* é marcado por uma reunião, chamada de *Sprint Review Meeting*, composta por todos: *Product Owner*, *ScrumMaster*, time e *stakeholders*. Esse momento não pode ultrapassar quatro horas e tem como objetivo apresentar para o *Product Owner* e para os *stakeholders* apenas as funcionalidades que foram feitas completamente, mostrando quais *user stories* do *Product Backlog* estão implementadas com base nos Critérios de Avaliação de cada *user story* (RISING, 2002) Artefatos que não são funcionalidades só podem ser apresentados quando servem de suporte para o entendimento da demonstração da execução do software (COHN, 2009).

Após a apresentação, os *stakeholders*, juntamente com o *Product Owner*, devem expor suas impressões sobre o produto e informar, caso haja, mudanças desejadas e classificá-las em níveis de prioridade. Caso vislumbrem alguma nova funcionalidade que não faz parte do *Product Backlog*, eles podem pedir a inclusão da mesma, indicando a prioridade (SUTHERLAND, 2011)

Com essa reunião encerrada, a *sprint* é fechada. Se houver alguma *user story* que deveria ter sido implementada e, por algum motivo, não foi totalmente completa, esta deve participar da reunião de planejamento da próxima *sprint* (SCHWABER, 2004). Porém, antes que se inicie uma nova *sprint*, ocorre uma reunião de retrospectiva para que se discuta o que funcionou e o que deixou a desejar na *sprint* recém-encerrada.

2.1.1.2.4. *Retrospectiva da sprint*

Após o encerramento de uma *sprint*, acontece uma reunião chamada de Retrospectiva da Sprint, ou *Sprint Retrospective Meeting*, que tem o objetivo de reunir o time, o *ScrumMaster* e o *Product Owner*, se ele quiser participar, para que eles discutam o que ocorreu bem durante a *sprint* e o que pode ser melhorado para a próxima *sprint* (RISING, 2002). Nessa fase, o *Scrum Master* tem o papel de

facilitador, para que a equipe como um todo consiga chegar a mudanças que impactem positivamente na execução da próxima *sprint* (SCHWABER, 2004).

2.1.1.3. Papéis

Segundo Cohn (2009), times e organizações resistem ao *Scrum* devido a diversos motivos diferentes, sendo um deles a confusão feita acerca dos papéis que existem num projeto que utiliza essa metodologia. Esses papéis são: *ScrumMaster*, *Product Owner*, time e *stakeholders*.

2.1.1.3.1. *Scrum master*

O *Scrum Master* é o líder do time: é a pessoa que tem a responsabilidade de não deixar com que o processo atrapalhe a produtividade da equipe (SCHWABER, 2004). Entretanto, existe uma peculiaridade acerca dessa liderança: a autoridade dada ao *Scrum Master* é estritamente em cima do processo e não sobre pessoas, ou seja, ele possui o poder de marcar as reuniões, de mediar discussões, entre outros, porém, não tem o direito de demitir alguém, por exemplo (COHN, 2009). Para auxiliar o *Scrum Master* na tomada de decisões no projeto, existem alguns artefatos que, devidamente atualizados, dão a exata noção do andamento e do custo da *sprint* e do projeto como um todo, como por exemplo, o gráfico *Burndown* e o *Agile EVM*.

2.1.1.3.2. *Stakeholders*

Os *stakeholders* são todos aqueles afetados pelo projeto que não fazem parte da equipe de desenvolvimento. Leffingwell (2011) divide os *stakeholders* em duas categorias: *stakeholders* de sistema e de projeto.

O primeiro tipo engloba qualquer um que usa diretamente o sistema, trabalha com os resultados daqueles que usam o sistema e/ou será impactado com o desenvolvimento e a operação do sistema. Dessa forma, esses *stakeholders* podem ser exemplificados por usuários do software (LEFFINGWELL, 2011).

Já os *stakeholders* de projeto são aqueles que possuem interesse direto no orçamento e no cronograma, em entender como o produto é desenvolvido e estão envolvidos na comercialização, instalação ou manutenção do software (LEFFINGWELL, 2011).

2.1.1.3.3. Equipe de desenvolvimento

A equipe de desenvolvimento do *Scrum* é formada por todos aqueles que trabalham na construção das funcionalidades do software de fato, que produzirão o código-fonte que formará o produto vislumbrado pelo cliente. Segundo Cohn (2009), “os programadores programam, testam, analisam, desenham e fazem qualquer coisa necessária para ajudar a equipe”. Schwaber (2001) diz que, no contexto do *Scrum*, esse time deve ser auto-gerenciável e auto-organizável.

2.1.1.3.4. Product Owner

Para Cohn (2009), o *Product Owner* (PO) é “a pessoa que certifica que o time está mirando o alvo certo”, apontando o time para a direção correta, com a ajuda do *ScrumMaster*, que organiza-o a fim de alcançar o objetivo da forma mais eficiente possível (COHN, 2009).

Schwaber e Sutherland (2011) enfatizam, no Guia *Scrum*, que o PO “é uma pessoa, não um comitê” que é “a única responsável por gerenciar o *Product Backlog*” e “deve representar os desejos de um comitê no *Product Backlog*”. Esse comitê é formado pelos *stakeholders*.

“Várias das responsabilidades do *Product Owner* envolvem o estabelecimento e a comunicação da visão do produto”. Essa necessidade de saber construir a dita “visão do produto” é essencial para a pessoa que assumir esse papel do *Scrum*, uma vez que “os melhores times são aqueles cujas paixões foram inflamadas através de um compartilhamento convincente da visão do *Product Owner*”, ou seja, a visão que o PO possui do produto precisa ser clara para todo e qualquer envolvido, caso contrário, várias outras responsabilidades desse personagem estará comprometida (COHN, 2009) (JUDY et al, 2008).

Com o intuito de concretizar o fato de que o time está no caminho certo, é de incumbência do *Product Owner* levantar os requisitos em conjunto com os *stakeholders* numa espécie de *brainstorming*. Nas metodologias ágeis os requisitos “são descobertos e detalhados durante todo o projeto”, não limitando “a descoberta do produto aos estágios iniciais de desenvolvimento” (PICHLER, 2011).

Com os requisitos em mãos, o PO deve definir as funcionalidades do produto, escrevendo *user stories* que apresentem as necessidades dos *stakeholders* na forma de funcionalidades, traduzi-las numa lista de prioridade, que é o *Product*

Backlog, atualizá-la e refiná-la continuamente, assegurando que ela está clara para todos (SUTHERLAND, 2011) (COHN, 2009).

Cohn (2009) destaca que, para a atualização da prioridade das *user stories*, além da visão, que “*mostra o que o produto pode tornar-se*” o PO deve levar em consideração aspectos restritivos da realidade, ou seja, a “visão do produto” deve possuir limites, que devem ser estabelecidos pelo *Product Owner*. Esses limites definidos determinarão o sucesso do produto e podem servir, também, como fator motivador, caso o PO saiba manipulá-los de modo a incentivar o time a solucionar um problema difícil, e “*não fixar inúmeros limites de modo a tornar o desafio impossível, desmotivando o time*”. Nesse contexto, “*o trabalho do Product Owner é criar limites para que o time pense dentro deles*” (COHN, 2009).

Segundo Sutherland (2011), Cohn (2009) e Leffingwell (2011), decorrente da responsabilidade de maximizar o retorno sobre investimento (ROI) do produto, o PO deve fazer a priorização das *user stories* de acordo com a visão de maior valor de negócio (*highest-business-value*), alinhado com objetivos estratégicos e atacando riscos. Ele deve, também, “*liderar o esforço do desenvolvimento para criar um produto que gere os benefícios almejados*” (PICHLER, 2010).

De acordo com Sutherland (2011), no caso de um desenvolvimento de aplicação interna, o PO não possui a responsabilidade sobre o ROI no sentido de produto comercial, somente na vertente da escolha das *user stories* e a priorização. Normalmente, nesse contexto, o *Product Owner* e o usuário costumam ser a mesma pessoa.

Cohn (2009) trata em sua obra do que ele chama de “*Atributos de um bom Product Owner*”. O autor apresenta brevemente cinco atributos que ele julga necessários para um bom PO, sendo que “Ter capacidade de decisão” e “Possuir autoridade” foram agrupados devido à correlação apresentada entre eles, cuja abordagem conjunta fora verificada em uma seção do livro “*Agile Software Requirements*”, do autor Dean Leffingwell (2011). Os atributos são:

- Possuir disponibilidade:

“*Product Owners sobrecarregados rapidamente se tornam gargalos e limitam o progresso do projeto*” (PICHLER, 2010). A indisponibilidade de um PO afeta todo o time, uma vez que ela implica na ruptura do ritmo do projeto, por exemplo, no caso da necessidade de uma resposta rápida que só o PO pode dar ao time, ou da falta

dessa pessoa em uma das reuniões. Portanto, além de prover a manutenção do ritmo das *sprints*, a disponibilidade do PO demonstra, também, o comprometimento dele em fazer o melhor produto possível (COHN, 2009).

- Possuir *business-savvy*:

Em algumas organizações, existe a divisão do papel de PO para mais de uma pessoa, criando, por exemplo, um papel de *Product Manager*, que fica responsável por gerenciar o *marketing* e a visão do produto, deixando para o *Product Owner* apenas as tarefas de guiar as *sprints* e as reuniões com o time. Essa prática, além de acarretar uma errônea definição e priorização de funcionalidades por falta de conhecimento de negócio, gera falta de autoridade, cria barreiras, atrasos e desperdícios (PICHLER, 2010).

É essencial para um *Product Owner* conhecer profundamente do negócio (*business-savvy*) em que o produto está inserido, questões de mercado, consumidores e usuários, pois esse conhecimento embasará decisões acerca do que deve ou não estar no produto e a prioridade das funcionalidades, uma vez que dá ao PO discernimento de quais *user stories* entregam maior valor para o usuário final (COHN, 2009) (LEFFINGWELL, 2011).

- Ser uma pessoa comunicativa:

No Manifesto Ágil, é dito que “o método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara” (BECK et. al., 2001). Nesse contexto, o papel do *Product Owner*, devido às suas responsabilidades, sobretudo às reuniões que participa, demanda que a pessoa que o assume deve ser comunicativa, uma vez que ela se relaciona com diferentes tipos de pessoas (*stakeholders*): usuários, clientes, parceiros e, naturalmente, o time.

“*Product Owners* habilidosos deverão possuir a capacidade de transmitir a mesma informação para cada interlocutor diferente”, sabendo manipular a mensagem para que seja entendida da melhor maneira possível (COHN, 2009). Leffingwell (2011) afirma que essa habilidade é necessária “pois o *Product Owner* traduz os objetivos de usuário e de negócio para o nível de detalhe adequado para a implementação”.

Segundo Cohn (2009), da mesma forma que transmitir informações, o PO deve saber ouvir muito bem as opiniões de todos, uma vez que, especialmente os

membros do time, após entenderem sobre a visão do produto e o mercado em que ele se insere, podem dar sugestões valiosas acerca do produto.

- Ter capacidade de decisão e possuir autoridade:

“Quando membros do time vão ao encontro de um *Product Owner* com um problema, eles querem uma solução” (COHN, 2009). Diante disso, “no ritmo do desenvolvimento ágil, nenhuma decisão (mesmo se a decisão for claramente “sem decisões por agora”) é pior que qualquer outra” (LEFFINGWELL, 2011).

De acordo com Leffingwell (2011), essa capacidade de decisão “requer autoridade, coragem e habilidade de admitir quando alguém está errado”. Uma vez que o *framework Scrum*, com sua dinâmica e responsabilidades bem definidas, “exerce pressão nos times para produzir funcionalidade o mais rápido possível”, a falta de um *Product Owner* que não decide é frustrante (COHN, 2009).

Pichler (2010) diz que um *Product Owner* sem autoridade, ou seja, aquele que “tem que consultar seu chefe para qualquer grande decisão”, não desempenha um trabalho efetivo, uma vez que, por consequência, não consegue, por exemplo, alinhar o *ScrumMaster* e os *stakeholders* ou retirar *user stories* do pacote de entrega da *sprint* por conta própria.

Dean Leffingwell (2001) completa os atributos definidos por Cohn (2009) com mais dois, apresentados em uma seção de seu livro similar com a da obra de Cohn. Estes são:

- Possuir fundamentos técnicos:

Uma vez que a evolução efetiva do escopo requer uma constante avaliação técnica, funcional, além dos valores na visão do usuário, o *Product Owner* necessita, para avaliar tecnicamente e funcionalmente o código-fonte do produto, conhecimento tecnológico. Além disso, no planejamento da *sprint*, pode ser necessário levar em consideração refatoração e/ou correção de defeitos no lugar de implementar novas funcionalidades; essa decisão, que deve ser tomada pelo PO, também necessita de um embasamento técnico (LEFFINGWELL, 2011).

A falta de conhecimento técnico pode acarretar em um dos “Problemas Comuns” citados por Cohn (2009): quando o *Product Owner* exerce uma pressão desnecessária em cima do time. Devido à pressão que o PO está sempre submetido, consequente da responsabilidade de devolver resultados financeiros

para a companhia, a pessoa que possui esse papel tem a tendência de forçar a implementação de um número de funcionalidades cada vez maior por *sprint*.

- Ser digno de confiança:

Segundo Leffingwell (2011), este é o atributo mais importante para um *Product Owner*, pois permeia todas as responsabilidades do papel. A confiança está muito atrelada com o sucesso do projeto, pois tanto “*o time deve confiar no Product Owner para fazer as chamadas rígidas na ordenação do escopo e para defender seus interesses na qualidade e na funcionalidade*”, quanto os *stakeholders* devem “*acreditar que o Product Owner apresenta fielmente suas prioridades acerca das funcionalidades ao time*”.

Todos esses atributos, nada mais são do que características que a pessoa que irá desempenhar o papel de *Product Owner* necessita ter para conseguir contribuir de forma efetiva com suas responsabilidades, como está apresentado no Quadro (1). Essas características estão diretamente relacionadas com competências que uma pessoa deve possuir para que esses atributos sejam cumpridos. Para que seja possível a definição das competências de um *Product Owner*, a próxima seção auxilia a compreensão sobre o que significa o termo “competência”.

Quadro 1. Resumo do papel de *Product Owner* (O autor).

Atividade	Relacionamentos			Responsabilidades do PO
	<i>ScrumMaster</i>	Time	<i>Stakeholders</i>	
<i>Sprint Planning Meeting</i>	Sim	Sim	Não	<ul style="list-style-type: none"> • Atualizar o <i>Product Backlog</i>; • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>; • Elucidar as dúvidas referentes às <i>user stories</i>.
<i>Sprint Review Meeting</i>	Sim	Sim	Sim	<ul style="list-style-type: none"> • Verificar, com base nos critérios de aceitação, se as <i>user stories</i> foram

Atividade	Relacionamentos			Responsabilidades do PO
	ScrumMaster	Time	Stakeholders	
				<p>implementadas corretamente;</p> <ul style="list-style-type: none"> • Expor sua impressão sobre o produto; • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>.
<i>Sprint Retrospective Meeting</i> (opcional)	Sim	Sim	Não	<ul style="list-style-type: none"> • Discutir melhorias quanto às suas atribuições, juntamente com o time, de modo a otimizar o desempenho da equipe como um todo.
<i>Sprint</i>	Sim	Sim	Sim	<ul style="list-style-type: none"> • Monitorar o ROI (retorno sobre investimento) do produto; • Estar disponível para elucidar dúvidas quanto às <i>user stories</i>; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>.
Levantar requisitos	Não	Não	Sim	<ul style="list-style-type: none"> • Extrair os requisitos do software a partir das necessidades dos <i>stakeholders</i>.

Atividade	Relacionamentos			Responsabilidades do PO
	ScrumMaster	Time	Stakeholders	
Escrever <i>user stories</i>	Não	Não	Sim	<ul style="list-style-type: none"> • Reproduzir os requisitos levantados com os <i>stakeholders</i> no formato de <i>user stories</i>. • Construir critérios de avaliação para a validação das <i>user stories</i>.
Priorizar <i>Product Backlog</i>	Não	Sim	Sim	<ul style="list-style-type: none"> • Levar em consideração o ROI (retorno sobre investimento) do produto.

2.2. COMPETÊNCIA

Segundo Júnior (2001), a origem da palavra competência tem origem no latim, *competere*, cuja etimologia é composta pelas palavras oriundas do latim *com*, que significa *conjunto* e *petere*, que quer dizer *esforço* (CZELUSNIAK, 2007).

Existe uma dificuldade inerente à definição do que é competência que se dá justamente à “*diversidade das interpretações do termo ao longo dos últimos trinta anos*” (DUTRA, et. al., 2000). Segundo Kilimnik e Sant’anna (2006), essas visões estão divididas em duas vertentes predominantes: a perspectiva anglo-saxônica, que define competências tomando como referência o mercado de trabalho e levando em consideração fatores ou aspectos ligados a descritores de desempenho requeridos pelas organizações; e a francesa, que enaltece o vínculo existente entre trabalho e educação, estabelecendo as competências como uma resultante de processos sistemáticos de aprendizagem.

De acordo com Fleury & Fleury (2001), McClelland deu início ao debate sobre competência nos Estados Unidos com a publicação do artigo “*Testing for Competence rather than Intelligence*”. Ainda segundo estes autores, os franceses começaram a debater a respeito de competência também na década de 70,

motivados pelo questionamento do conceito de qualificação técnica e do processo de formação profissional.

Um dos conceitos de competência mais aceitos é a proposta por Parry (1996), que o define como um “*cluster de conhecimentos, habilidades e atitudes relacionados que afetam a maior parte de um papel profissional ou responsabilidade, que se correlaciona com a performance desse papel ou responsabilidade, que possa ser medido contra parâmetros “bem aceitos” e que pode ser melhorado através de treinamento*” (PARRY 1996 apud DUTRA 2002).

Segundo Fleury (2001), a competência é “*um saber agir responsável e reconhecido, que implica mobilizar, integrar, transferir conhecimentos, recursos e habilidades, que agreguem valor econômico à organização e valor social ao indivíduo*”. Este conceito está ilustrado pela Figura (4).



Figura 4. Competências como Fonte de Valor para o Indivíduo e para a Organização (FLEURY, 2001).

Cardoso (2004) define competência como a capacidade de um indivíduo executar uma determinada tarefa, em uma determinada circunstância, uma vez que isso depende de sua história, sua vontade, do ambiente organizacional percebido e das características específicas da situação em que a tarefa será executada. “*Com isso, competência está fortemente relacionada ao conhecimento do indivíduo, mas também suas experiências e atitudes perante uma dada situação*” (BORSATTO, et. al., 2006).

Nesse mesmo contexto e corroborando com as ideias de Fleury (2002), Borsatto (2006) e Cardoso (2004), Durand (1998) e Leme (2006) constroem o conceito de competência apoiado em três pilares:

- Conhecimentos, que são as informações já assimiladas pela pessoa: o **Saber**;
- Habilidades, que são a capacidade de aplicar o conhecimento adquirido: o **Saber Fazer**;
- Atitudes para converter conhecimentos e habilidades em ações que agregam valor social à pessoa: o **Querer Fazer**.

A partir dessas definições de **Conhecimentos**, **Habilidades** e **Atitudes** (CHA), Leme (2006) separa competências em dois grupos:

- **Competências Técnicas**: “*É tudo o que o profissional precisa ter para desempenhar seu papel, e são representadas pelo C e pelo H do CHA, o ‘Saber’ e o ‘Saber Fazer’*” (LEME, 2006);
- **Competências Comportamentais**: “*São o diferencial competitivo de cada profissional e têm impacto em seus resultados. Aqui, são expressas pelo A do CHA, o ‘Querer Fazer’*” (LEME, 2006).

Esta separação está representada pelo Quadro (2).

Quadro 2. Resumo de competência (adaptado de Leme (2006)).

Conhecimento	Saber	Competência Técnica
Habilidade	Saber Fazer	
Atitude	Querer Fazer	Competência Comportamental

O autor destaca que a “*separação do CH e do A para definir as competências técnica e comportamental é uma separação didática*”, uma vez que ambos os tipos de competências estão apoiados nos três pilares, como Durand (1998) e Parry (1996) enfatizam.

No âmbito deste trabalho, serão utilizadas as definições de conhecimento, habilidade e atitude apresentados por Durand (1998) e Leme (2006) em conjunto com a separação de competências proposta por Leme, sendo parte do cerne do trabalho a definição de **competências**, tanto **técnicas** quanto **comportamentais** para um *Product Owner*.

3. METODOLOGIA DE PESQUISA

3.1. CLASSIFICAÇÃO DA PESQUISA

A escolha da metodologia de pesquisa utilizada neste trabalho se baseou a partir da definição dos objetivos identificados para solucionar a lacuna vislumbrada. A classificação da metodologia é dada em três categorias. São elas:

- Quanto à natureza:

A pesquisa aplicada mostrou-se mais adequada para este trabalho, uma vez que ela tem como objetivo gerar conhecimentos para a aplicação prática voltada para problemas específicos.

- Quanto aos objetivos:

A pesquisa exploratória encaixou-se melhor à realidade do trabalho, visto que, para a produção do conjunto de competências de um *Product Owner*, foi necessário fazer um levantamento bibliográfico acerca de tudo o que envolve esse papel e sobre competências de modo geral.

- Quanto aos procedimentos:

A pesquisa bibliográfica foi escolhida pela necessidade do levantamento de referências teóricas válidas já analisadas e publicadas que dessem embasamento teórico para o documento. Fontes sem tratamento analítico não foram consideradas.

Essa pesquisa foi dividida nas seguintes fases:

- Planejamento da pesquisa:

No início, com a escolha do tema abordado, foi definida a questão de pesquisa, que, conseqüentemente, explicitaram os objetivos, geral e específicos.

- Coleta de dados:

Com a questão de pesquisa e o *modus operandi* definidos, deu-se início à coleta de dados, em que são colocados em prática os procedimentos de pesquisa e as técnicas de coleta. Nesse momento, a revisão bibliográfica, apoiada por uma revisão sistemática, foi executada, a fim de investigar o estado da arte em que o trabalho se insere e construir uma base teórica para o desenvolvimento do mesmo. Quanto à revisão sistemática, seus detalhes estão dispostos na Seção (3.2) e o

Protocolo de Revisão Sistemática elaborado está presente no Apêndice (I) deste documento.

3.2. REVISÃO SISTEMÁTICA

Para o auxílio na execução da revisão sistemática e no refinamento das fontes, foram utilizados dois *softwares* gratuitos: Zotero e StArt. O primeiro tem como função gerenciar as referências encontradas pelo pesquisador, permite que o usuário colete-as através de um *plug-in* de *browser*, cite e compartilhe as fontes de pesquisa utilizadas. Já a segunda aplicação, cujo nome completo é *State of the Art through Systematic Review*, apesar do nome em inglês, foi desenvolvida pelo Laboratório de Pesquisa em Engenharia de Software (LAPES) do Departamento de Computação da Universidade Federal de São Carlos (DC/UFSCar) e tem como objetivo, como o próprio nome sugere, auxiliar o pesquisador na revisão sistemática do estado da arte, dando suporte à técnica e provendo gráficos interativos que dão ao usuário uma visibilidade tanto quantitativa quanto qualitativa das referências analisadas durante a revisão.

Com a execução das *strings*, presentes no protocolo de pesquisa, foram retornados 40 trabalhos, sendo 16 da *ACM Digital Library* e 24 provenientes da base de dados *IEEEExplore*, como está apresentado no Gráfico (1).

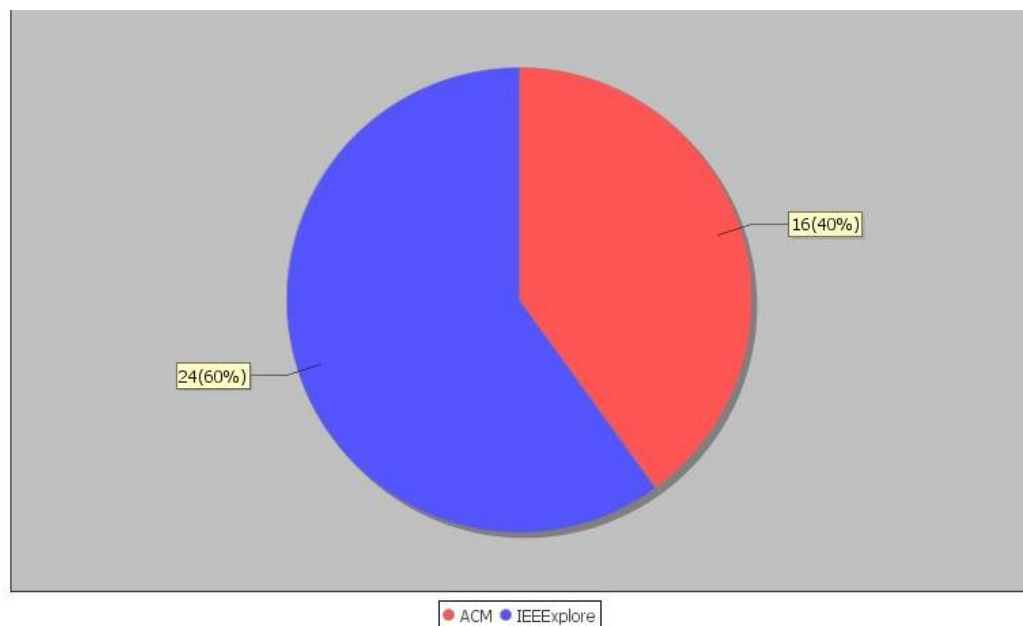


Gráfico 1. Trabalhos coletados por fonte (O autor).

O refinamento de utilização dessas referências foi dividido em duas etapas:

- Seleção:

A fase de Seleção teve como objetivo excluir os documentos que não possuem relação com metodologias ágeis a partir do título e dos *abstracts*, e identificar duplicação nas publicações encontradas;

- Extração:

Já na parte de Extração, os trabalhos que foram aceitos na Seleção são lidos na íntegra, para que sejam selecionados apenas aqueles que possuem todos os critérios de inclusão definidos no Protocolo de Revisão Sistemática, presente no Apêndice (I) deste documento.

Na etapa de Seleção, foram identificadas duas duplicações e 9 trabalhos foram descartados por possuir algum critério de exclusão, sendo aceitos 29 documentos para serem examinados na fase de Extração. Nesse momento, 5 documentos foram rejeitados por apresentarem algum critério de exclusão, sendo um em comum: “Não apresenta características/atributos de um *Product Owner*”, como está apresentado no Gráfico (2).

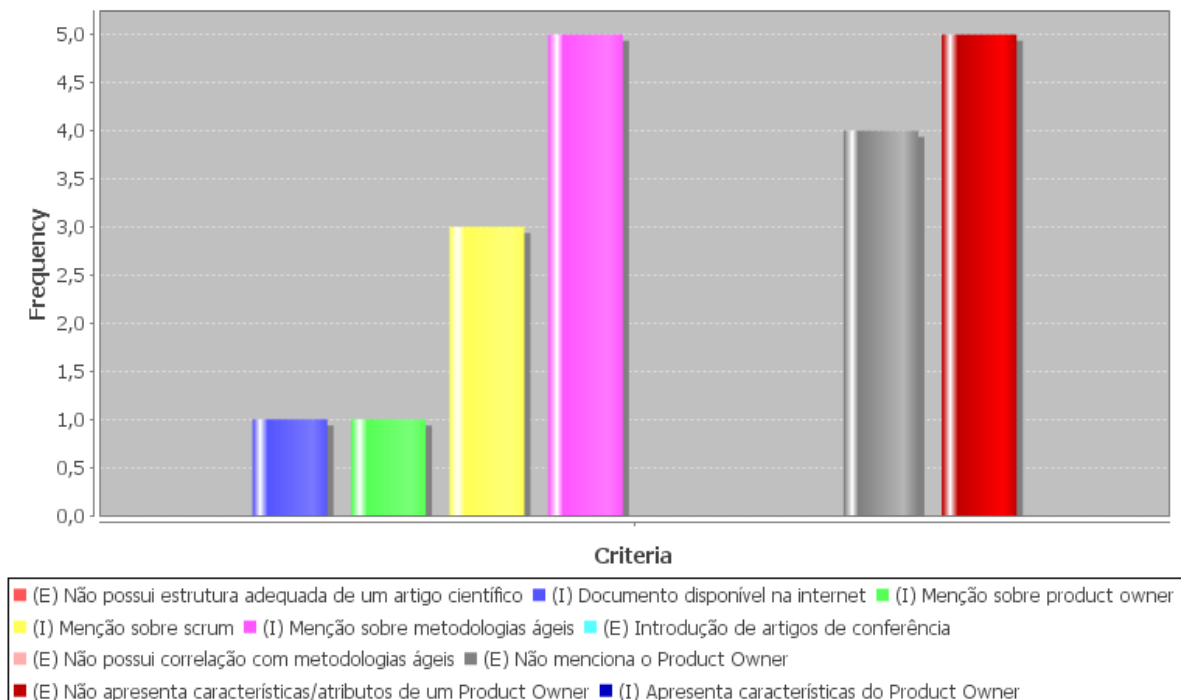


Gráfico 2. Etapa de Extração dos documentos (O autor).

Finalmente, 24 artigos foram caracterizados como aceitos por possuírem todos os critérios de inclusão.

- Redação dos resultados:

Após a aceitação dos 24 artigos na fase de Extração, estes foram classificados quanto à quais atributos de um *Product Owner* eles fazem menção; classificação, esta, ilustrada pelo Gráfico (3). Esses atributos são definidos na Seção (2.1.1.3.4), a qual caracteriza o papel do PO. Essa classificação favoreceu a definição das competências do PO e, conseqüentemente, serviu de insumo para a construção do instrumento de pesquisa que ocorreu na segunda etapa deste trabalho.



Gráfico 3. Caracterização do *Product Owner* nos artigos aceitos (O autor).

Com o estudo desses artigos e a classificação proposta no Gráfico (3), foi produzido o Quadro (8), presente no Apêndice (II), que indica o nível em que o documento aborda os atributos. Esse detalhamento busca a visualização da lacuna em que este trabalho se propõe a preencher no estado da arte, está definido da seguinte maneira:

- Bem fundamentado: cita o atributo, explica-o e o evidencia em estudo de caso;
- Fundamentado: cita o atributo e explica-o;
- Pouco fundamentado: apenas cita o atributo.

Com essa classificação quanto à fundamentação foi gerado o Quadro (3), que relata o número percentual da aparição de cada atributo dentre os 24 artigos selecionados e a porcentagem acerca dos níveis de fundamentação que eles foram

tratados. Essas informações fornecem uma visão consolidada da importância que os autores dos artigos deram a cada atributo e o grau de detalhe que cada um foi tratado.

Quadro 3. Resumo dos atributos encontrados nos artigos (O autor).

Atributo	Aparição em artigos	Bem fundamentado	Fundamentado	Pouco fundamentado
Comunicativo	67%	37%	44%	19%
Autoridade	54%	38%	31%	31%
Capacidade de decisão	58%	62%	23%	15%
Business-savvy	63%	67%	20%	13%
Disponibilidade	38%	33%	33%	33%
Escrever <i>user stories</i>	58%	79%	21%	0%
Levantar requisitos	63%	53%	47%	0%
Priorizar <i>backlog</i>	83%	70%	30%	0%
Validar produto	67%	65%	35%	0%
Monitorar ROI	17%	75%	25%	0%
Planejar <i>sprint</i>	13%	100%	0%	0%
Definir métricas	4%	100%	0%	0%
Validar protótipo	4%	100%	0%	0%
Traçar estratégia de marketing e venda	4%	0%	100%	0%
Manter relações de negócio	4%	0%	100%	0%
Gerenciar projeto	4%	0%	100%	0%
Computar <i>velocity</i>	4%	0%	0%	100%
Manter visão do projeto	4%	100%	0%	0%

Atributo	Aparição em artigos	Bem fundamentado	Fundamentado	Pouco fundamentado
Definir aspectos técnicos e arquiteturais do software	8%	50%	50%	0%

A partir dos resultados obtidos com a coleta de dados, foi desenvolvido o Referencial Teórico deste documento, apresentado na Seção (2).

Utilizando como base o conceito de competência definido nesse Referencial e contemplando os atributos encontrados na literatura e esmiuçados na Seção (2.1.1.3.4), as atividades que um PO participa no decorrer de um projeto e as responsabilidades deste personagem, foi elaborado um conjunto inicial de **competências para um *Product Owner***. Este conjunto permeia a formação básica deste papel e está disposto no Quadro (9), o qual está presente no Apêndice (III).

Com o intuito de avaliar a importância desse conjunto inicial de competências, foi desenvolvido um instrumento de pesquisa, cujos maiores detalhes da elaboração, análise, aplicação e resultados estão na Seção (3.3).

3.3. INSTRUMENTO DE PESQUISA

3.3.1. Elaboração do instrumento de pesquisa

O objetivo desta pesquisa é **validar a proposta de um conjunto de competências técnicas e comportamentais para um *Product Owner***, que foi elaborada após revisão sistemática e bibliográfica.

Com este intuito foi elaborado um questionário, pois, segundo Hartmut Günther (2003), é o “*instrumento principal para o levantamento de dados por amostragem*”. Dessa forma, um questionário é o instrumento de pesquisa que melhor se adequou ao contexto e ao objetivo do trabalho (GHÜNTHER, 2003).

Para construir um questionário, devem-se levar em consideração os objetivos da pesquisa, que, por sua vez, determinarão os conceitos a serem investigados. Esses conceitos derivarão os itens, que são os conteúdos das perguntas Os

principais estágios de um questionário estão presentes na Figura (5) (SCHUMAN & KALTON,1985).



Figura 5. Estágios principais de um Survey (Adaptado de Schuman & Kalton (1985)).

No caso deste trabalho, os conceitos investigados foram os graus de importância que cada competência proposta possui para pessoas de diferentes experiências com metodologias ágeis e de distintas áreas de atuação.

Uma vez que o questionário foi construído na ferramenta Google Drive, a fim de facilitar a aplicação e a coleta de dados, bem como aumentar o alcance da pesquisa, os itens que compuseram o questionário são as competências propriamente ditas, presentes no Quadro (9), de tal forma a facilitar o entendimento e diminuir o tempo gasto do respondente (DILLMAN, 1978).

Além disso, os objetivos pautam a população alvo, que podem variar desde toda a população de um estado até, por exemplo, profissionais que atuam na iniciativa privada. Essa caracterização foi feita a partir de itens presentes em uma seção do instrumento de pesquisa denominado “Dados do Entrevistado”, o qual está explicitado na primeira versão do questionário, presente no Apêndice (IV) deste documento.

No contexto em que foi construído o questionário, a população alvo é qualquer pessoa que tenha qualquer experiência com métodos ágeis, especialmente com *Scrum*: alunos, professores ou profissionais de uma equipe de desenvolvimento, independente do ambiente em que atuam, se público ou privado. Essa preferência quanto à experiência do participante com *Scrum* se deu pelo fato deste framework trazer o papel do *Product Owner*.

Como o objetivo da pesquisa é mensurar a importância de cada competência, foi utilizada a Escala *Likert*, pois a utilizando, no âmbito deste trabalho, mede-se o grau de discordância ou concordância com declarações relativas à importância de cada competência por parte do respondente (BAKER, 2005).

No questionário em questão foram utilizadas cinco alternativas para cada item, uma vez que um número ímpar de alternativas dá a oportunidade de o respondente marcar um meio termo, se assim julgar (GHÜNTHER, 2003).

A escala está representada por meio da Figura (6), possui dois extremos, sendo um negativo, chamado “Nada importante”, e um positivo, denominado “Extremamente importante” e cinco números acima das alternativas. Esses números auxiliam o respondente a se situar, pois, por exemplo, quanto menor o número e, conseqüentemente, mais próximo do “Nada importante”, menor o grau de importância que ele está classificando aquela competência.

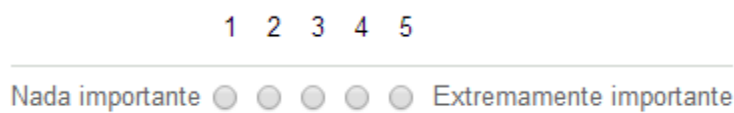


Figura 6. Escala utilizada no questionário (O autor).

Dessa forma, foi gerada uma primeira versão do questionário, presente no Apêndice (IV), o qual foi submetida à validação de um grupo de especialistas em metodologias ágeis, sobretudo o *framework Scrum*, sendo quatro professores, dois servidores do setor público federal e dois profissionais de empresas particulares.

3.3.2. Validação do instrumento

Antes de aplicar o instrumento de pesquisa, foi realizada uma validação do mesmo por especialistas (academia, iniciativa privada e setor público), a qual está apresentada a seguir. Tal validação teve como objetivo fazer uma verificação crítica acerca da semântica do conteúdo apresentado, da estrutura do documento e do conteúdo, de fato.

Além disso, julgou-se importante receber sugestões de novas competências de pessoas que possuem experiência no desenvolvimento de software utilizando *Scrum*, uma vez que elas complementam a revisão sistemática realizada.

Após a validação realizada, foi produzida uma segunda versão do questionário, presente no Apêndice (V) deste documento, com as sugestões e comentários indicados e detalhados nas Seções (3.3.2.1), (3.3.2.2) e (3.3.2.3).

3.3.2.1. Academia

Quatro professores da Faculdade Gama - Universidade de Brasília (FGA/UnB), especialistas em metodologias ágeis, foram entrevistados com a finalidade da análise do questionário.

Os entrevistados apoiaram a aplicação do instrumento de pesquisa, uma vez que um conjunto de competências validado pode servir de referência para a identificação de uma pessoa em uma organização para exercer o papel do PO.

Alguns comentários foram tecidos por parte dos especialistas, sendo eles:

- Recomendação que o questionário fosse divulgado de uma forma que abrangesse a maioria das regiões brasileiras, pois, caso ficasse restrito ao contexto de Brasília, os resultados obtidos seriam diferentes e deturpados, uma vez que, em sua opinião, a maioria dos profissionais que atuam com metodologia ágil na capital federal não a executa como está descrito na literatura.
- Inclusão de uma breve descrição em todas as competências, a fim de reduzir possíveis falhas de entendimento por parte dos respondentes;
- Inclusão de uma numeração dos itens do questionário, com o intuito de facilitar a referência destes em possíveis sugestões dos respondentes;
- Inclusão de algumas das técnicas de planejamento ágil, para situar o respondente na descrição da competência técnica referente ao domínio destas técnicas supracitadas;
- Reformulação das competências técnicas “Domínio do negócio em que o projeto está inserido” e “Possuir visão estratégica”, incluindo em ambas o nível, estratégico ou operacional, a que elas se referem;
- Reformulação da competência técnica “Dominar técnicas de levantamento de requisitos ágeis”, pois, segundo os especialistas, o levantamento de requisitos independe da metodologia utilizada;

- Reformulação da competência técnica “Dominar a técnica de escrita de *user story*”, caso fosse julgado necessário levar em consideração outras técnicas, como, por exemplo, BDD (*Behavior Driven Development*);
- Reformulação da competência técnica “Dominar a técnica da definição dos critérios de aceitação”, levando em consideração que critérios de aceitação fazem parte da validação de requisitos, a qual abrange mais técnicas que apenas a supracitada;
- Melhor definição de qual sentido é utilizado o monitoramento do ROI na competência técnica “Dominar as técnicas de monitoramento do ROI do produto”;
- Explicitação das técnicas de comunicação na competência técnica “Dominar as técnicas de comunicação”;
- Reformulação das competências técnicas “Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento de software”, “Possuir conhecimento técnico de teste funcional” e “Possuir conhecimento técnico de refatoração”, pois, segundo os especialistas, o *Product Owner* não necessita ter conhecimento técnico sobre tecnologias utilizadas, teste funcional ou refatoração, mas acerca do impacto delas no projeto;
- Inclusão de competências técnicas relacionadas às necessidades do PO em ter o conhecimento dos valores ágeis e dos princípios ágeis;
- Inclusão de uma competência técnica relacionada à negociação;
- Adição do fato de o PO possuir a habilidade em ajudar a equipe na competência comportamental “Possuir a disposição para ajudar a equipe”;
- Inclusão de uma competência relacionada à capacidade do PO de se adequar a diferentes contextos, tanto em projetos distintos, quanto dentro das camadas estratégica e operacional de um mesmo projeto;
- Inclusão de uma competência comportamental referente à resolução de conflitos.

Após a avaliação feita no contexto acadêmico, o instrumento de pesquisa foi validado por especialistas que atuam no setor público, cujos detalhes estão apresentados na Seção (3.3.2.2).

3.3.2.2. Setor público

Da mesma forma que os representantes da academia fizeram, dois servidores do setor público federal, atuantes no desenvolvimento de software atualmente utilizando o *framework Scrum*, avaliaram o questionário.

Um deles, atualmente, exerce a função de Secretário de TI no Supremo Tribunal Federal (STF); o outro, por sua vez, atua como analista de informática – desenvolvimento de sistemas na Procuradoria-Geral da República (PGR). Ambos foram favoráveis à aplicação do questionário, pois consideraram o objetivo e o resultado da pesquisa relevantes e de grande valia para o momento atual, o qual marca uma tentativa de utilização de metodologias ágeis em órgãos da Administração Pública Federal.

Os entrevistados aprovaram a estrutura, o conteúdo e a semântica do questionário e fizeram as seguintes observações:

- Conhecer os ritos e cerimônias realizados durante um projeto ágil, especialmente quanto à finalidade e aos objetivos deles;
- Possuir a proatividade e a atitude necessárias para auxiliar a equipe do projeto na remoção de obstáculos ligados ao negócio (ou seja, que não tenham relação com questões técnicas de TI).

Por fim, a validação do questionário foi realizada por profissionais de institutos de pesquisa e desenvolvimento. O resultado dessa validação está presente na Seção (3.3.2.3).

3.3.2.3. Institutos de pesquisa e desenvolvimento

A terceira abordagem nas validações foi realizada com dois profissionais de institutos de pesquisa e desenvolvimento que, atualmente, estão inseridos no contexto de desenvolvimento de software utilizando *Scrum*. Esses institutos são caracterizados como Oscip (Organização da Sociedade Civil de Interesse Público), ou seja, são organizações criadas por pessoas jurídicas de direito privado sem fins lucrativos que podem relacionar-se por meio de parceria com o Estado.

Um dos avaliadores trabalha no Instituto de Pesquisas Eldorado e exerce a função de analista de software – desenvolvedor *mobile*; enquanto o outro especialista a avaliar o instrumento atua como desenvolvedor de software no Instituto Sapiencia. A análise de ambos foi positiva quanto ao entendimento, ao conteúdo e à estrutura e aos objetivos da aplicação do instrumento.

Algumas observações foram feitas, tais como:

- Julgamento do item relacionado ao sexo do respondente como desnecessário;
- Sugestão da alteração das alternativas “Time” e “Stakeholder” do item referente aos outros papéis desempenhados, exceto PO, para, respectivamente, “Time de desenvolvimento” e “Interessado”, e a exclusão da alternativa “Patrocinador”, uma vez que, em sua opinião, faz parte dos “Interessados”, e “Estudante”, já que não existe no *Scrum*;
- Sugestão da alteração do item “Qual(is) a(s) aplicação(ões) dessa experiência?” para “Qual(is) a(s) natureza(s) da empresa dessa experiência?”;
- Sugestão de que o item “Atualmente, qual sua área de atuação no desenvolvimento de software?” poderia ser melhor formulado, pois, em sua opinião, pode induzir o respondente a pensar nas áreas da engenharia de software, como, por exemplo, teste, desenvolvimento e requisitos;
- Julgamento da competência “Conhecer e aplicar adequadamente técnicas de levantamento de requisitos ágeis” como inválida, pois entende que técnicas de requisitos são independentes da metodologia;
- Sugestão da alteração da competência “Conhecer e aplicar adequadamente a técnica de escrita de *user story*” para “Conhecer e aplicar adequadamente a técnica para definição de requisitos” com a prerrogativa de que *user story* é apenas uma das formas para se definir requisitos;
- Sugestão da inclusão de uma competência, a qual ele não soube dizer se técnica ou comportamental, que seria relacionada à negociação de prazo e mudança de escopo em um projeto.

O resultado quanto aos ajustes feitos no instrumento de pesquisa influenciados pelas observações de todos os especialistas está detalhado na Seção (3.3.3).

3.3.3. Ajustes do instrumento de pesquisa

Após a etapa de validações, as observações dos especialistas foram analisadas e, aquelas que foram consideradas importantes para a melhoria da qualidade do instrumento, foram postas em prática. Sendo elas:

- Inclusão de uma breve descrição em todas as competências e exemplos naquelas que se julgou necessário;
- Numeração dos itens do questionário;
- Retirada da competência técnica “Dominar técnicas de levantamento de requisitos ágeis”;
- Reformulação da competência técnica “Dominar a técnica da definição dos critérios de aceitação” para “Dominar a técnica de validação de requisitos” (CT5);
- Retirada do termo “técnico” da competência técnica “Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento de software” (CT10);
- Reformulação da competência técnica “Possuir conhecimento técnico de teste funcional” para “Possuir conhecimento do impacto de teste funcional no projeto” (CT11);
- Reformulação da competência técnica “Possuir conhecimento técnico de refatoração” para “Possuir conhecimento do impacto de refatoração no projeto” (CT12);
- Inclusão da competência técnica “Possuir conhecimento dos ritos e cerimônias realizadas no *Scrum*” (CT13);
- Inclusão da competência técnica “Possuir conhecimento dos valores ágeis” (CT14);
- Inclusão da competência técnica “Possuir conhecimento dos princípios ágeis” (CT15);
- Inclusão da competência técnica “Dominar técnicas de negociação” (CT16);
- Reformulação da competência comportamental “Possuir a disposição para ajudar a equipe” para “Possuir a disposição e habilidade em ajudar a equipe” (CC4);
- Inclusão da competência comportamental “Possuir a habilidade de se adequar a diferentes contextos” (CC5);
- Inclusão da competência comportamental “Possuir a habilidade de resolver conflitos” (CC6);

- Inclusão do período de tempo “em anos ou meses” dos itens presentes na seção “Dados do Entrevistado” referentes à experiência do respondente utilizando o *Scrum*, ao exercício do papel do PO e ao desempenho de outros papéis;
- Inclusão do item “Essa experiência como PO foi positiva?” na seção “Dados do Entrevistado”;
- Reformulação do item “Qual(is) a(s) aplicação(ões) dessa experiência com *Scrum*” para “Em qual(is) contexto(s) essa experiência com *Scrum* foi aplicada” da seção “Dados do Entrevistado”;
- Alteração das alternativas “Acadêmica” para “Área acadêmica” presentes em dois itens da seção “Dados do Entrevistado”;
- Criação da seção “Sugestões, críticas e avaliações” com um campo para que o respondente, caso quisesse, incluísse sugestões de novas competências ou críticas sobre o instrumento de pesquisa, e um campo para que o participante inserisse um e-mail para contato, caso quisesse receber o resultado da aplicação do questionário.

Terminada a fase de ajustes do instrumento, este foi construído em uma plataforma *web* e divulgado. Os detalhes quanto à aplicação do questionário estão presentes na Seção (3.3.4).

3.3.4. Aplicação do instrumento de pesquisa

Após a etapa de ajustes do instrumento, este foi construído em cima da plataforma gratuita Google Docs e disponibilizado através da seguinte URL: https://docs.google.com/forms/d/1KGKFRONai02r2QAMjsTC4TQCKV3zZGVwUA625y8LKJ0/viewform?usp=send_form.

A divulgação foi feita por meio de comunidades em redes sociais referentes a desenvolvimento de software, listas de e-mail de comunidades ágeis e para e-mails de setores de empresas que utilizam *Scrum*.

O instrumento de pesquisa ficou disponível para respostas do dia 15/04/2014 até o dia 09/05/2014, totalizando 24 dias de disponibilidade. Ao final foram obtidas 68 respostas, apresentadas na Seção (4).

3.3.5. Avaliação do questionário

3.3.5.1. Consistência do questionário

De acordo com Ledesma et. al. (2002), o método mais utilizado para medir o grau de consistência interna de um questionário, é o cálculo do Alfa de Cronbach. Esse recurso mede a confiabilidade do instrumento de pesquisa em função do número dos itens e a proporção de variância total e a covariância entre os itens.

O cálculo é feito através de uma expressão matemática, apresentada em um artigo por Lee J. Cronbach em 1951, e, de acordo com Freitas (2005), o grau de fidedignidade varia de acordo com o valor de Alfa, sendo essa variação explícita no Quadro (4).

Quadro 4. Variação da confiabilidade pelo valor de Alfa (adaptado de Freitas (2005)).

Confiabilidade	Valor de Alfa
Muito baixa	$\alpha \leq 0,30$
Baixa	$0,30 < \alpha \leq 0,60$
Moderada	$0,60 < \alpha \leq 0,75$
Alta	$0,75 < \alpha \leq 0,90$
Muito alta	$\alpha > 0,90$

O cálculo do Alfa de Cronbach do instrumento de pesquisa deste trabalho foi executado pela ferramenta IBM SPSS Statistics – versão 22 e apontou como resultado o valor de **0,842**. Utilizando a classificação proposta por Freitas (2005), a confiabilidade do questionário em questão foi considerada **alta**.

3.3.5.2. Representatividade dos resultados

No contexto deste trabalho, a aplicação do questionário foi feita por amostragem, uma vez que não foi encontrada a quantidade de profissionais que trabalham com metodologias ágeis no Brasil e existe uma dificuldade inerente de se obter a opinião de toda esta população.

Segundo Hair (*apud* SIMON, 2004), de acordo com a escala selecionada, a quantidade absoluta de respondentes deve ser no mínimo de quatro a cinco vezes o número de questões, para que seja realizada uma análise fatorial dos dados. No caso deste trabalho, para que fosse válido realizar a análise em questão, a

quantidade mínima de participantes deveria ser de, ao menos, 88 pessoas, visto que, o instrumento construído possui 22 itens.

Essa quantidade não foi alcançada, uma vez que o número total de respondentes foi de 68. Devido à quantidade insuficiente de participantes, esta pesquisa não apresentará informações acerca da execução de uma análise fatorial.

4. APRESENTAÇÃO DOS RESULTADOS

4.1. CARACTERIZAÇÃO DA POPULAÇÃO RESPONDENTE

A fim de caracterizar a população respondente, foi elaborada uma seção no instrumento denominada “Dados do Entrevistado”, contendo os seguintes itens:

- Sexo;
- Experiência utilizando o *Scrum*, com o objetivo de extrair a quantidade de tempo, em meses ou anos, que o respondente participou de projetos utilizando o *framework*;
- Atuação como PO, com a finalidade de saber se o participante já atuou como *Product Owner* em algum projeto;
- Experiência como PO, com o intuito de se obter a quantidade de tempo, em meses ou anos, que o respondente atuou como PO;
- Caráter da experiência como PO, para saber qual a avaliação que o participante faz da sua experiência no papel de PO.

Uma fragilidade do instrumento de pesquisa observada foi o fato de não ter sido fixado uma unidade de tempo para os itens referentes à experiência do participante quanto ao uso do *Scrum*, ao exercício do papel de PO e de outros papéis. Esse detalhe fez com que alguns respondentes deixassem de indicar a unidade em sua resposta, dificultando a análise dos dados. Os valores que não possuem unidade estão definidos como “Dado sem unidade” no Quadro (5).

Dessa forma, um resumo dos resultados dos principais itens da seção supracitada está apresentado no Quadro (5).

Quadro 5. Caracterização dos respondentes (O autor).

Item	Alternativas	Resultados	
		Absoluto	Percentual
Sexo	Masculino	57	84%
	Feminino	11	16%
Experiência utilizando <i>Scrum</i>	Nenhuma	2	3%
	1 a 6 meses	8	12%

Item	Alternativas	Resultados	
		Absoluto	Percentual
	7 a 12 meses	9	13%
	1 a 3 anos	27	40%
	3 a 5 anos	9	13%
	Mais de 5 anos	5	7%
	Dado sem unidade	8	12%
	Atuação como PO	Sim	20
Não		48	71%
Experiência como PO	Nenhuma	47	69%
	1 a 6 meses	6	9%
	7 a 12 meses	4	7%
	1 a 3 anos	9	13%
	Mais de 3 anos	1	1%
	Dado sem unidade	1	1%
Caráter da experiência como PO	Não se aplica	46	68%
	Positiva	21	31%
	Negativa	1	1%

Verificou-se uma inconsistência na resposta de um participante: o resultado do item “Atuação como PO” foi “Não”, enquanto o item “Caráter da experiência como PO” também teve como resposta “Não”, sendo que, neste item, a resposta esperada seria “Não se aplica”, pois não existe possibilidade de se avaliar o caráter de uma atuação que não se concretizou. Para fins de análise, esta resposta foi alterada pela esperada.

Outra inconsistência foi encontrada no item correspondente à experiência com *Scrum*: cinco respondentes indicaram não possuir experiência no desenvolvimento de software utilizando o *framework*, porém indicaram possuir experiência em outros papéis diferentes do *Product Owner*. Nestes casos, foram descartadas as respostas para o item referente à experiência em outros papéis do *Scrum*.

Os demais resultados referentes à caracterização dos respondentes estão presentes no Apêndice (VI).

4.2. COMPETÊNCIAS TÉCNICAS

Na seção do questionário referente às Competências Técnicas, todas aquelas presentes no Quadro (9), as quais possuem “CT” no “ID”, juntamente com as propostas pelos especialistas, foram transcritas para avaliação juntamente com uma breve descrição para reduzir possíveis dificuldades ou erros de entendimento. Os dados detalhados estão presentes no Apêndice (VI).

A avaliação da importância de todas as competências foi feita com base em uma Escala *Likert*, a qual está descrita na Seção (3.3.1), Figura (6).

4.2.1. Dominar técnicas de planejamento de desenvolvimento ágil

No resultado da avaliação dessa competência, que é relacionada ao domínio de técnicas como *planning poker*, *velocity* e *burndown*, os dados se concentraram nos valores de 3 a 5 na escala: apenas 16% considerou-a “Nada” ou “Pouco importante”.

4.2.2. Dominar o negócio em que o projeto está inserido

Os dados resultantes da avaliação dessa competência, a qual se refere ao PO ter um conhecimento consistente do domínio do negócio no contexto operacional, convergiram quase que totalmente para o maior valor da escala (81%).

4.2.3. Dominar técnicas de priorização do product backlog

Essa competência está relacionada com a capacidade do PO definir quais user stories são mais importantes para o projeto e teve o resultado da votação concentrado nos valores de 3 a 5 da escala, uma vez que apenas 3% definiu-a como “Pouco importante”.

4.2.4. Dominar a técnica de escrita de user story

A competência que está ligada à capacidade do PO de extrair os requisitos do cliente e transformá-los em *user stories* teve como resultado uma massiva escolha dos valores 3 ao 5 das alternativas, sendo que apenas 13% esteve dentro dos outros valores.

4.2.5. Dominar a técnica de validação de requisitos

Esse item está relacionado à capacidade de validar as funcionalidades implementadas, a qual pode ser feita por meio de critérios de aceitação, que, por sua vez, devem ser testáveis. Apenas 8% dos respondentes avaliaram-na como “Pouco” ou “Nada importante”.

4.2.6. Dominar as técnicas de monitoramento do ROI

Esta competência, relacionada à capacidade de garantir que o investimento feito no projeto seja consumido conforme o planejado, e que os mesmos trarão os benefícios esperados à organização, obteve como resultado 93% dos votos concentrados nos valores de 3 a 5 da escala proposta.

4.2.7. Dominar as técnicas de comunicação

Uma vez que o *Product Owner* tem a necessidade de se comunicar com pessoas de camadas diferentes (estratégica, tática e operacional), foi proposta esta competência.

De acordo com as respostas computadas, apenas 3% dos participantes consideraram-na “Pouco importante”, sendo que o restante votou de “Importante” até “Extremamente Importante”.

4.2.8. Possuir visão estratégica

Esta competência está intimamente ligada ao domínio que o *Product Owner* deve ter do negócio no nível estratégico. Dessa forma, 99% dos participantes consideraram-na, no mínimo, “Importante”.

4.2.9. Possuir visão sistêmica

A necessidade de o PO possuir a compreensão do contexto de negócio em que o projeto está envolvido, suas partes (estratégica, tática e operacional) e a relação existente entre elas deu origem a esta competência. Ela foi avaliada por 96% como, minimamente, “Importante”.

4.2.10. Possuir conhecimento das tecnologias utilizadas no desenvolvimento do software

Esta competência se refere à necessidade do PO, por exemplo, priorizar certas *user stories* do *Product Backlog* na primeira *sprint*, uma vez que estas forçam a implementação da arquitetura do software.

As avaliações quanto à importância dessa competência ficaram bem distribuídas, sendo que a alternativa mais escolhida foi “Importante”, com 26%.

4.2.11. Possuir conhecimento do impacto de teste funcional no projeto

A necessidade de, por exemplo, priorizar a implementação de testes funcionais em vez de novas funcionalidades, em um momento em que a cobertura de testes do código é considerada baixa, se relaciona com esta competência.

Os dados resultantes das respostas obtidas concentraram-se nos valores 3 e 4 da escala, totalizando uma soma de 70% do total.

4.2.12. Possuir conhecimento do impacto da refatoração no projeto

Esta competência está relacionada com a necessidade de, por exemplo, priorizar a refatoração do código em vez de implementar novas funcionalidades, a fim de prezar pela qualidade do código.

As respostas ficaram divididas quase que uniformemente entre os valores 2, 3 e 4, restando 22% para os extremos.

4.2.13. Possuir conhecimento dos ritos e cerimônias realizadas no Scrum

Pelo fato de o PO participar de ritos e cerimônias presentes no *Scrum*, como o *Sprint Planning Meeting* e *Sprint Daily Meeting*, esta competência foi proposta.

De acordo com as respostas computadas, 86% delas foram distribuídas entre “Extremamente Importante”, “Muito Importante” e “Importante”.

4.2.14. Possuir o conhecimento dos valores ágeis

A necessidade de o PO saber que, por exemplo, para um projeto ágil, é “mais importante a colaboração do cliente que negociação de contratos” e que “indivíduos

e interação entre eles é mais importante que processos e ferramentas”, gerou esta competência.

Esta competência foi avaliada por 90% dos respondentes como “Importante”, “Muito Importante” ou “Extremamente Importante”.

4.2.15. Possuir o conhecimento dos princípios ágeis

Esta competência refere-se à necessidade de o PO saber que, por exemplo, “conhecedores do negócio e desenvolvedores devem trabalhar juntos diariamente durante o projeto”.

O resultado das respostas indicam que a maioria dos participantes, 94%, julgaram a competência como “Importante”, “Muito Importante” ou “Extremamente Importante”.

4.2.16. Dominar técnicas de negociação

A necessidade de o PO negociar, por exemplo, variação de prazo e escopo com o cliente e com o time, serviu de insumo para a criação desta competência.

O resultado das respostas dos participantes indicou que a maioria deles (81%) julgou a competência como “Muito Importante” ou “Extremamente Importante”.

4.3. COMPETÊNCIAS COMPORTAMENTAIS

De modo semelhante como fora feito com as Competências Técnicas, todas aquelas Competências Comportamentais presentes no Quadro (9), as quais possuem “CC” no “ID”, juntamente com as competências sugeridas pelos especialistas foram transcritas para avaliação no questionário em sua seção própria, juntamente com uma breve descrição para reduzir possíveis dificuldades ou erros de entendimento.

4.3.1. Possuir a coragem para tomar uma decisão

Esta competência refere-se ao *Product Owner* ser capaz e não hesitar em tomar uma decisão. Os resultados indicam que todos os participantes julgaram o item como, no mínimo, “Importante”.

4.3.2. Possuir a capacidade de admitir quando está errado

O PO deve ser capaz e ter a humildade para reconhecer perante as pessoas envolvidas no projeto, que errou. Os dados de resposta desta competência indicam que 94% dos participantes julgaram-na “Extremamente Importante” ou “Muito Importante”.

4.3.3. Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que se relaciona

A fim de fazer referência à necessidade de o PO ser sincero e responsável com as pessoas e com os compromissos assumidos, esta competência foi criada. De todos os participantes, 72% consideraram o item “Extremamente Importante”.

4.3.4. Possuir a disposição e a habilidade em ajudar a equipe

Em determinadas ocasiões, o PO precisa sanar dúvidas da equipe relacionadas ao negócio, objetivando esclarecer *user stories* e seus critérios de aceitação. Com o objetivo de contemplar essa necessidade, a competência em questão foi elaborada.

As respostas dos participantes convergiram para os valores 4 e 5 da escala proposta, sendo que apenas 6% foram destinadas para o 3.

4.3.5. Possuir a habilidade de se adequar a diferentes contextos

O trânsito constante do PO entre os níveis estratégico e operacional, chegando até ao time de desenvolvimento, induziu a construção desta competência.

Dos dados resultantes das respostas, 88% convergiram para os dois maiores valores da escala de importância proposta.

4.3.6. Possuir a habilidade de resolver conflitos

A necessidade do PO em resolver, por exemplo, conflitos interpessoais e/ou conflitos de estratégia que possam vir a prejudicar o andamento e o sucesso do projeto deu origem a esta competência.

De acordo com as respostas obtidas, 94% delas estão concentradas nos valores 3, 4 e 5, ou seja, do “Importante” ao “Extremamente Importante”.

4.4. CONSIDERAÇÕES DOS RESPONDENTES

Ao final do questionário existia uma seção denominada “Sugestões, críticas e avaliações”, composta por dois itens: o primeiro dava a oportunidade de o respondente sugerir novas competências ou sugestões sobre o trabalho; já a segunda dava a opção de o respondente deixar o e-mail para receber o resultado da pesquisa.

Um respondente sugeriu a inclusão do conceito de PO no cabeçalho do questionário, com o intuito de lembrar àqueles que trabalharam há mais tempo com *Scrum* e não estão mais em contato com o *framework*.

Outra sugestão recebida foi a de medir a importância das competências de forma relativa, dando ao respondente a opção de escolher, entre duas competências, qual a mais importante.

Dois comentários positivos sobre o trabalho foram recebidos como resposta deste campo, parabenizando a organização do questionário e a construção do mesmo.

Um participante fez a observação de que talvez a pesquisa não estivesse bem elaborada e, como consequência disso, as respostas quanto à importância das competências seriam, em sua maioria, “Extremamente Importante”; esse fato, em sua opinião, não ajudaria no trabalho.

4.5. ANÁLISE DOS RESULTADOS

4.5.1. Critérios utilizados na seleção dos itens

Antes da análise dos dados obtidos através da pesquisa, houve a aplicação de critérios para que houvesse uma seleção dos itens cujos resultados não apresentaram um estabelecimento da opinião da comunidade participante.

A partir dos resultados dos itens do questionário foram plotados histogramas, que são representações gráficas nas quais os dados são agrupados em classes de valores e apresentados na forma de retângulos, para cada item. A base de cada retângulo, presente no eixo x, corresponde a um valor da escala definida para o grau de importância da competência; enquanto a altura dos retângulos é definida pela quantidade (frequência) de respostas (MORETTIN, 2009).

Com os gráficos gerados, presentes no Apêndice (VI), foi observada a curva da função de densidade da distribuição normal para cada item, uma vez que essa curva apresenta o comportamento dos dados ao longo da escala (MORETTIN, 2009).

Diante disso, os itens 10, 11, 12 e 13 da seção de Competências Técnicas, que apresentaram sua curva semelhante à curva da distribuição normal padrão, a qual está representada pela Figura (7), foram selecionados para análise, pois o comportamento da curva destes indica que a quantidade das respostas referentes aos graus “Nada importante” e “Pouco importante” está equivalente com a quantidade de respostas voltadas para os graus “Extremamente importante” e “Muito importante”.

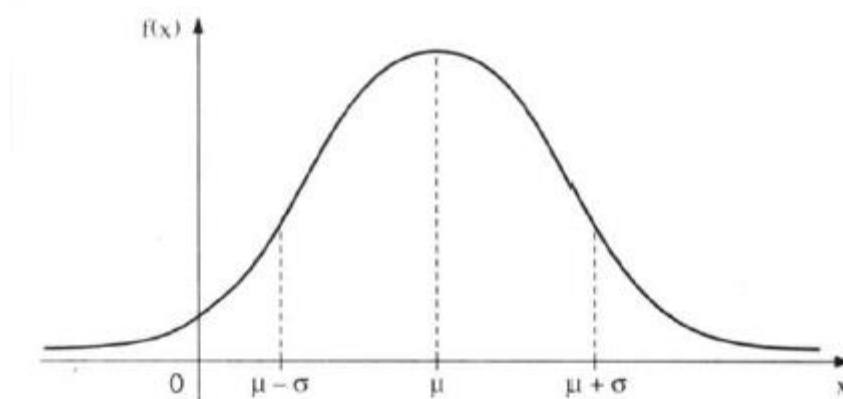


Figura 7. Curva da distribuição normal (MORETTIN, 2009).

Os demais itens, cujas curvas de densidade da distribuição normal apresentaram maior altura à direita do gráfico, ou seja, mais afastada do eixo “y”, indicam que as competências referentes a eles tiveram seus conceitos considerados já estabelecidos pela comunidade, uma vez que houve uma convergência das respostas dos respondentes para os altos graus de importância, e, portanto, não carecem de uma análise detalhada.

Caso houvesse algum item cuja curva apresentasse maior altura à esquerda, ou seja, mais próxima do eixo vertical, ele não seria selecionado para a análise, pois haveria uma concentração de respostas nos baixos graus de importância, inferindo que essa competência não possui importância para a comunidade.

4.5.2. Análise dos dados

Objetivando analisar as características dos respondentes que julgaram as competências selecionadas como “Nada Importante” (grau 1) ou “Pouco Importante” (grau 2), os dados das respostas dos mesmos foram agrupados. Da mesma forma, foram aglomerados os resultados que se referem aos graus 4 (“Muito Importante”) e 5 (“Extremamente Importante”) da escala. Os valores referentes ao grau 3, ou seja, os respondentes que consideraram a competência como “Importante”, não foram consideradas, pois teve-se como objetivo traçar os perfis dos respondentes reunidos nas extremidades da escala.

A principal característica dos respondentes escolhida para fazer parte da análise do perfil dos respondentes foi o contexto da experiência com *Scrum* que o respondente passou. Os demais itens presentes no questionário referentes à caracterização do respondente não foram utilizados, pois não trouxeram nenhum resultado relevante na análise feita. Para cada competência selecionada foram elaboradas duas figuras contendo os números absolutos de respondentes para cada contexto: iniciativa privada, setor público e área acadêmica. A primeira figura representa a distribuição dos respondentes que julgaram a competência “Nada Importante” ou “Pouco Importante”, ou seja, indicaram os graus de importância 1 e 2, e, a segunda, daqueles que julgaram o item “Muito Importante” ou “Extremamente Importante”.

O item relacionado à este contexto de experiência com o *framework*, presente na seção “Dados do Entrevistado” do questionário, permitia a múltipla escolha, desta forma, os valores presentes nas faixas de intersecção das opções nas figuras supracitadas representam o número absoluto de respondentes que escolheram ambas as opções.

4.5.2.1. Possuir conhecimento das tecnologias utilizadas no desenvolvimento de software

Esse item, que se refere à Competência Técnica 10 (CT10), tem a distribuição das suas respostas apresentada no Gráfico (4).

Os valores apresentados no Gráfico (4) e na Figura (8) indicam que metade dos respondentes que consideraram a competência como “Nada Importante” ou “Pouco Importante” teve sua experiência com *Scrum* **apenas** no contexto do **setor público**. Em contrapartida, somente 4 participantes disseram já ter atuado na área acadêmica, porém, nenhum deles atuou apenas nesse contexto. Ainda acerca desta análise, 11 dos 24 respondentes, ou seja, cerca de 46% destes, já atuaram na iniciativa privada.

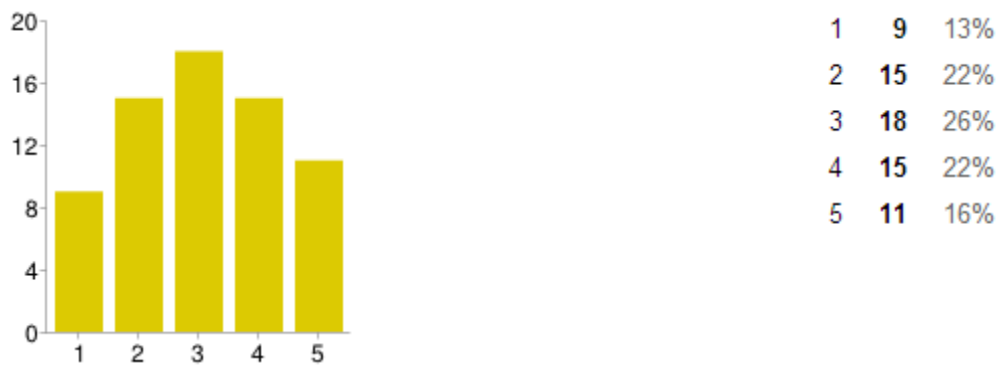


Gráfico 4. Possuir conhecimento das tecnologias utilizadas no desenvolvimento de software (O autor).

Com isso, infere-se que a o perfil do respondente que julgou a CT10 como “Nada Importante” ou “Pouco Importante” é de um profissional que possui experiência no **âmbito privado ou no setor público**.



Figura 8. Distribuição dos respondentes por contexto: graus 1 e 2 de importância - CT10 (O autor).

Dos participantes que classificaram a competência técnica em questão como “Muito Importante” ou “Extremamente Importante”, 17 (65%) têm em sua experiência um histórico na **iniciativa privada**, 11 (42%) já atuaram no setor público e 9 (35%) já estiveram na área acadêmica.

Dessa forma, os dados, representados pela Figura (9), indicam que houve um notório aumento dos votos dos profissionais que já atuaram na área acadêmica para os maiores níveis de importância para esta competência. Entretanto, observa-se que a maioria dos que a consideraram “Muito Importante” ou “Extremamente Importante” possui experiência na **iniciativa privada**.

Vale ressaltar que houve um empate no número de respondentes que já atuaram como PO em ambos os casos: 7. Isto leva ao indício que, neste caso, não há, de fato, um estabelecimento da importância desta competência para quem já exerceu o papel na comunidade pesquisada.



Figura 9. Distribuição dos respondentes por contexto: graus 4 e 5 de importância - CT10 (O autor).

4.5.2.2. Possuir conhecimento do impacto de teste funcional no projeto

Os dados resultantes das respostas obtidas deste item, o qual é relativo à Competência Técnica 11 (CT11), estão presentes no Gráfico (5).

Os dados presentes na Figura (10) apontam que, daqueles que consideraram a competência como “Nada Importante” ou “Pouco Importante”, apenas dois (15%) já tiveram experiência no âmbito acadêmico, enquanto 8 (cerca de 62%) deles já atuaram no contexto do setor público. Por outro lado, 7 afirmaram já ter atuado na iniciativa privada.



Gráfico 5. Possuir conhecimento do impacto de teste funcional no projeto (O autor).

Com isso, destaca-se que a o perfil do respondente que julgou a CT11 como “Nada Importante” ou “Pouco Importante” é de um profissional que possui experiência basicamente no **âmbito privado ou no setor público**.

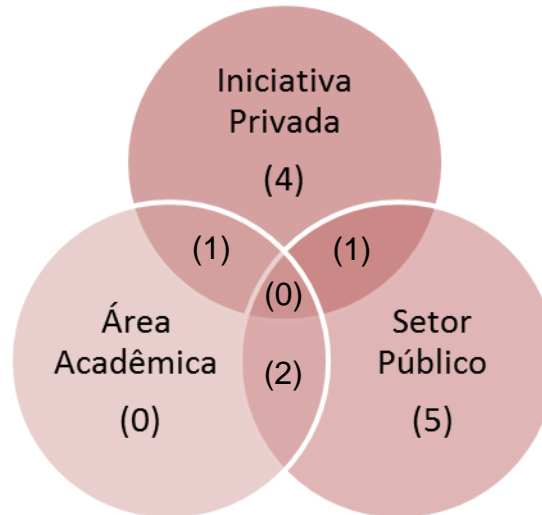


Figura 10. Distribuição dos respondentes por contexto. Graus 1 e 2 de importância - CT11 (O autor).

Quanto aos participantes que classificaram a competência técnica em questão como “Muito Importante” ou “Extremamente Importante”, a Figura (11) ilustra que 12 (38%) têm em sua experiência um histórico na área acadêmica, 16 (52%) já atuaram na iniciativa privada e 15 (48%) já estiveram no setor público.

Desse modo, destaca-se uma grande participação de profissionais da academia na classificação desta competência como, no mínimo, “Muito Importante”, quando relacionado com o caso anterior. Entretanto, a maior parte dos respondentes possui em seu perfil experiência no **setor público e na iniciativa privada**.

Comparando a quantidade de participantes que declararam já ter atuado como PO nas duas análises realizadas, houve uma diferença percentual de 6% e absoluta de 6 respondentes, sendo que a maior parte classificou a CT11 como “Extremamente Importante” ou “Muito Importante”. Este destaque se faz necessário, pois há de se ressaltar a opinião daqueles que exerceram o papel em questão, possibilitando fazer um contraste da visão “externa” do mesmo.



Figura 11. Distribuição dos respondentes por contexto. Graus 4 e 5 de importância - CT11 (O autor).

4.5.2.3. Possuir conhecimento do impacto da refatoração no projeto

As respostas referentes à Competência Técnica 12 (CT12) estão ilustradas no Gráfico (6).



Gráfico 6. Possuir conhecimento do impacto da refatoração no projeto (O autor).

As informações presentes na Figura (12) indicam que, daqueles que consideraram a competência como “Nada Importante” ou “Pouco Importante”, 14 (70%) já tiveram experiência no setor público, enquanto 9 (45%) deles já atuaram no contexto da iniciativa privada. Por outro lado, apenas 4 afirmaram já ter atuado na área acadêmica.

Com isso, pode-se traçar o perfil do respondente que avaliou esta competência com os graus supracitados como um profissional que basicamente teve experiência na **iniciativa privada ou setor público**.



Figura 12. Distribuição dos respondentes por contexto: graus 1 e 2 de importância - CT12 (O autor).

Os respondentes que julgaram a CT12 como “Muito Importante” ou “Extremamente Importante” se dividem da seguinte forma: 12 (48%) afirmaram já passar por experiência na iniciativa privada, 14 (56%) no setor público e 9 (36%) na área acadêmica. Estes dados estão presentes na Figura (13).

Dessa forma, destaca-se um aumento considerável de respondentes que já atuaram na academia, se comparado com o caso anterior. Entretanto, a maior parte dos respondentes que fizeram a classificação em questão atuaram na **iniciativa privada ou no setor público**.

Suscita-se a diferença percentual da quantidade de respondentes que já atuaram como PO: 15% consideraram-na “Pouco Importante” ou “Nada Importante” e 32% classificaram-na como “Muito Importante” ou “Extremamente Importante”. Estes dados revelam que dentre aqueles que já exerceram o papel em questão, a maior parte considera a competência com um alto grau de importância.



Figura 13. Distribuição dos respondentes por contexto. Graus 4 e 5 de importância - CT12 (O autor).

4.5.2.4. Possuir conhecimento dos ritos e cerimônias realizadas no scrum

De acordo com as respostas computadas para o item referente à Competência Técnica 13 (CT13), foi plotado o Gráfico (7).

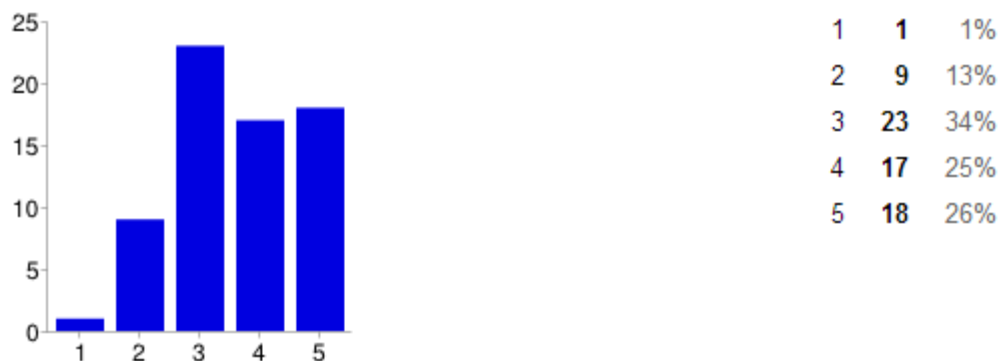


Gráfico 7. Possuir conhecimento dos ritos e cerimônias. (O autor)

A Figura (14) apresenta os seguintes dados sobre os respondentes que consideraram a Competência Técnica 13 (CT13) como “Nada Importante” ou “Pouco Importante”: 3 (30%) atuaram apenas na iniciativa privada, a mesma quantidade informaram que atuaram apenas no setor público e, na mesma proporção, estão aqueles que declararam atuar nos dois contextos. Um respondente, por sua vez, declarou ter atuado no setor público e na área acadêmica.

Com isso, verifica-se, para essa classificação de importância da CT13 um perfil de profissional que atuou na **iniciativa privada e/ou no setor público**.



Figura 14. Distribuição dos respondentes por contexto: graus 1 e 2 de importância - CT13 (O autor).

Por outro lado, os respondentes que julgaram esta competência como “Muito Importante” ou “Extremamente Importante” têm sua distribuição apresentada pela Figura (15) e se dividiram da seguinte maneira: 8 (cerca de 23%) afirmaram já ter atuado na área acadêmica, enquanto 19 (em torno de 54%) indicaram ter experiência no setor público, número que se repetiu, também, para aqueles que informaram ter experiência na iniciativa privada.

Fazendo uma comparação com o caso anterior, destaca-se um aumento considerável de respondentes com experiência na área acadêmica. Entretanto, o perfil do participante que classificou a competência como “Muito Importante” ou “Extremamente Importante” é àquele com experiência nos contextos do **setor público ou da iniciativa privada**.

Ressalta-se que, apesar da diferença absoluta de 10 respondentes que já atuaram como PO nos dois casos, a diferença percentual é de apenas 7%, sendo que a maior parte julgou a CT13 como “Extremamente Importante” ou “Muito Importante”. Este destaque se dá pela pouca diferença do número de respondentes que já exerceram o papel de PO no julgamento da importância da competência.

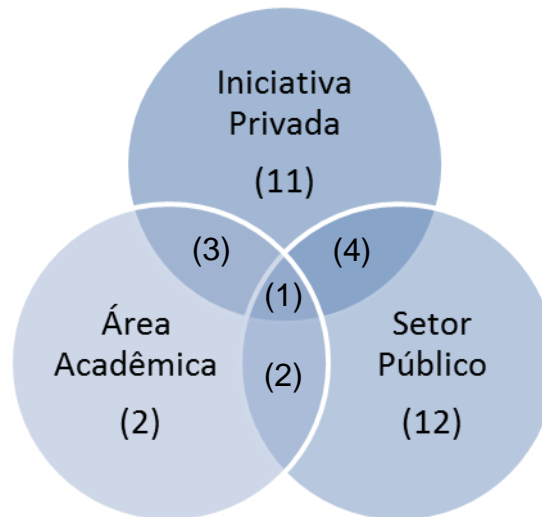


Figura 15. Distribuição dos respondentes por contexto. Graus 4 e 5 de importância - CT13 (O autor).

Após a análise dos dados, foi elaborado um ranking das competências, baseado nos resultados obtidos com a aplicação do questionário. Os detalhes quanto a este ordenamento estão apresentados na Seção (4.6).

4.6. RANKING DAS COMPETÊNCIAS

A última seção criada no instrumento de pesquisa, a qual foi denominada “Sugestões, críticas e avaliações”, possuía um item de campo de texto aberto para que os respondentes deixassem quaisquer observações que julgassem necessárias. Com isso, das considerações que foram computadas, duas sugeriam a comparação das competências quanto às suas importâncias.

Dessa forma, com o objetivo de propor as competências com maior grau de importância na visão da comunidade pesquisada, foi construído um ranking com todas as competências propostas.

O ordenamento das competências foi feito da seguinte forma: cada item do questionário se relacionava com uma competência proposta no Quadro (9); com base nas respostas obtidas, somou-se o a quantidade absoluta de respostas referentes aos valores 3, 4 e 5 da escala proposta, uma vez que estes valores indicam que a competência em questão é, no mínimo, “Importante”.

Nos casos em que ocorreu empate entre duas ou mais competências, o desempate foi feito com a comparação das quantidades de respostas referentes ao

maior valor da escala, dessa forma, contemplaria àquela que foi considerada por mais pessoas como “Extremamente Importante”.

O ranking contempla todas as competências, tanto técnicas quanto comportamentais, e está apresentado no Quadro (6), sendo que a posição da competência está representada pela coluna **R**. Foi construído, também da mesma maneira, um ranking para cada tipo de competência: o referente às competências técnicas está presente no Apêndice (VII) e, o relacionado às comportamentais, no Apêndice (VIII).

Quadro 6. Ranking das competências. (O autor)

R	ID	Competência	Resultado	
			Absoluto	Percentual
1º	CC3	Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que relaciona	68	100%
2º	CC1	Possuir a coragem para tomar uma decisão	68	100%
3º	CC2	Possuir a capacidade de admitir quando está errado	68	100%
4º	CC4	Possuir a disposição para ajudar a equipe	68	100%
5º	CT8	Possuir visão estratégica	67	99%
6º	CT2	Dominar o negócio em que o projeto está inserido	66	97%
7º	CT7	Dominar as técnicas de comunicação	66	97%
8º	CT3	Dominar técnicas de priorização do <i>Product Backlog</i>	66	97%
9º	CC5	Possuir a habilidade de se adequar a diferentes contextos	65	96%
10º	CT9	Possuir visão sistêmica	65	96%
11º	CC6	Possuir a habilidade de resolver conflitos	64	94%
12º	CT15	Possuir o conhecimento dos princípios ágeis	64	94%
13º	CT16	Dominar técnicas de negociação	63	93%

R	ID	Competência	Resultado	
			Absoluto	Percentual
14º	CT6	Dominar as técnicas de monitoramento do ROI (retorno sobre investimento) do produto	63	93%
15º	CT5	Dominar a técnica de validação de requisitos	62	92%
16º	CT14	Possuir o conhecimento dos valores ágeis	61	90%
17º	CT4	Dominar a técnica de escrita de <i>user story</i>	59	87%
18º	CT13	Possuir conhecimento dos ritos e cerimônias realizadas no <i>Scrum</i>	58	86%
19º	CT1	Dominar técnicas de planejamento de desenvolvimento ágil	57	84%
20º	CT11	Possuir conhecimento do impacto de teste funcional no projeto	55	81%
21º	CT12	Possuir conhecimento do impacto de refatoração no projeto	48	71%
22º	CT10	Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento do software	44	65%

Este ranking apresenta certa preferência por boa parte das Competências Comportamentais pelos respondentes. Apesar disso, há uma pequena diferença tanto absoluta quanto percentual entre as 20 primeiras competências, uma vez que apenas duas destoaram do conjunto.

5. CONCLUSÕES E TRABALHOS FUTUROS

5.1. CONCLUSÕES

Este Trabalho de Conclusão de Curso teve como objetivo propor um conjunto de competências para um *Product Owner*, que foi feito por meio de revisão bibliográfica e sistemática do estado da arte. Essa proposição se deu a partir do mapeamento das responsabilidades, atividades e relacionamentos do papel de PO no *Scrum*, bem como suas características, cerimônias e peculiaridades.

Além do conjunto de competências, composto por 22 competências, sendo 16 técnicas e 6 comportamentais, objetivou-se a validação deste por meio de um instrumento de pesquisa. Como resultado, 18 das competências apresentaram seus conceitos estabelecidos perante a amostra pesquisada, sendo que as outras 4 tiveram seus dados submetidos a uma análise mais profunda acerca do perfil dos respondentes.

A análise dos dados selecionados resultou na constatação de que as respostas referentes aos menores graus da escala, basicamente eram de pessoas que não possuíam experiência com *Scrum* na área acadêmica. Por outro lado, alguns dos respondentes que julgaram as competências como “Extremamente Importante” ou “Muito Importante” declararam ter experiência com o *framework* ágil na academia.

A partir da análise dos dados ressalta-se que, apesar de quase 20% da amostra pesquisada contemplar respondentes de outros estados, o maior foco dos resultados foi no contexto do Distrito Federal. Cabe destacar que, depois de obtidos os resultados da aplicação do questionário, os resultados desta pesquisa condizem com a revisão sistemática e bibliográfica realizada, o que indica uma consistência nos resultados da aplicação desses métodos.

Este trabalho se propôs a auxiliar na construção de um perfil para um *Product Owner*. Espera-se auxiliar a pessoa que delegará esse papel e escolher o profissional que melhor se encaixa com base no conjunto de competências aqui proposto e validado pela pesquisa.

Salienta-se que a revisão sistemática deste trabalho fez parte de um artigo, elaborado pelos mesmos autores deste documento, denominado “*A systematic review of the main Product Owner’s features*”, apresentado em maio de 2014 no 11th International Conference on Information Systems and Technology Management.

Ressalta-se que este trabalho faz parte de um projeto sobre competências, realizado na Universidade de Brasília – Campus Gama (UnB/FGA). Atualmente, este projeto possui outros dois trabalhos que se complementam com este, sendo que o segundo está em andamento:

- “*Proposta de um conjunto de competências para um time ágil*” (SOARES, G.H.F.);
- “*Proposta de um conjunto de competências para um ScrumMaster*” (QUERUBIM, T.S.).

Por fim, consideram-se, também, como pontos fortes do trabalho a proposta inicial de competências, uma vez que apenas quatro delas precisaram ter seus dados analisados neste trabalho e não houve novas sugestões; a consistência do questionário, confirmada na Seção (3.3.5.1); a validação do questionário feita por especialistas, deixando a versão final do questionário mais robusta.

5.2. TRABALHOS FUTUROS

Como o tempo de disponibilização e divulgação do instrumento de pesquisa teve de ser restrito devido ao contexto deste Trabalho de Conclusão de Curso, houve dificuldade na obtenção de respostas de um número representativo de pessoas como um todo, sobretudo de outros estados. Esse pequeno número obtido acarretou, entre outros fatores, na dificuldade de caracterização da amostra e indica que os resultados apresentados refletem a visão da população local.

Dessa forma, sugere-se como trabalho futuro a reaplicação do instrumento durante um maior espaço de tempo, com maior divulgação interestadual e com maior número de respondentes que já atuaram como PO.

O estudo do impacto das competências nas atividades do PO em um projeto é outra sugestão de trabalho futuro, pois é uma maneira de avaliar a importância de cada uma delas na prática.

Por fim, incentiva-se a construção de um instrumento para avaliar quais das competências propostas estão presentes em uma amostra de profissionais que

almejam exercer o papel de *Product Owner*, objetivando auxiliar a escolha do profissional mais adequado para a função.

Bibliografia

- BACA, D., & CARLSSON, B. **Agile development with security engineering activities**. Proceedings of the 2011 International Conference on Software and Systems Process, p.149-158, 2011.
- BASS, J. **Agile Method Tailoring in Distributed Enterprises: Product Owner Team**. IEEE 8th International Conference on Global Software Engineering, 2013.
- BAKER, M.J. **Administração de marketing**. Rio de Janeiro: Elsevier, 2005.
- BECK, K., BEEDLE, M. COCKBURN, A. et al. **Manifesto para o desenvolvimento ágil de software**. 2001. Disponível em: www.manifestoagil.com.br. Acessado em: 27/09/2013.
- BOEHM, B. **Get ready for agile methods, with care**. 2002.
- BORSATTO, G., SHIBATA, T., & SANTOS, E.L. **Aplicação da gestão de competências por processos: um estudo de caso em uma empresa petrolífera**. In: XXVI Encontro Nacional de Engenharia de Produção, 2006, Fortaleza. Anais.
- BRASIL**. Lei n.º 8666, de 21 de junho de 1993. Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, v.
- BRASIL**. INSTRUÇÃO NORMATIVA Nº 04, de 12 de novembro de 2010. Dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal. Diário Oficial [da] República Federativa do Brasil, Brasília, DF, v.
- BRASIL**. Tribunal de Contas da União. Conhecimento acerca da utilização de métodos ágeis nas contratações para desenvolvimento de software pela administração pública federal. Acórdão 2314-33/2013. Relator: José Múcio Monteiro. 28 de agosto de 2013. Ata nº 33/2013 – Plenário.
- BUDWIG, M., JEONG, S., & KELKAR, K. **When User Experience Met Agile: A Case Study**. Extendend Abstracts on Human Factors in Computing Systems. Boston, MA, USA, 2009.
- CARDOSO, V. C., LESSA, E., QUEIROZ, J.G. et al. **O Caso de Implantação da Gestão de Competências por Processos no Operador Nacional do Sistema Elétrico**. In: Congresso Anual da Sociedade Brasileira de Gestão do Conhecimento, 3, 2004, São Paulo. Anais.
- CHARETTE, R.N. **Why Software Fails**. IEEE Spectrum, Setembro 2005, 42.

- COHN, M. **Sprint**. Mountain Goat Software. 2005. Disponível em: <<http://www.mountaingoatsoftware.com/>>. Acesso em: 10 out. 2013.
- COHN, M. **Succeeding with Agile: Software Development with Scrum**. Addison-Wesley, 2009.
- COHN, M. **User Stories Applied: For Agile Software Development**. Addison-Wesley, 2004.
- CZELUSNIAK, D. J. **Proposta de sistema para apoio à gestão de competências baseada em sistemas agentes**. Ponta Grossa: [s.n.], 2007.
126 f.: il. ; 30 cm.
- DILLMAN, D.A. **Mail and telephone surveys: The total design method**. New York: John Wiley & Sons, 1978.
- DURAND, T. **Forms of Incompetence. Proceedings Fourth International Conference on Competence-Based Management**. Oslo: Norwegian School of Management, 1998.
- DUTRA, J., HIPOLITO, J., & SILVA, C. **Gestão de pessoas por competências: o caso de uma empresa do setor de telecomunicações**. Revista de Administração, p.161–176. 2000.
- FISHER, K., & BANKSTON, A. **From Cradle to Sprint: Creating a Full-Lifecycle Request Pipeline at Nationwide Insurance**. Agile Conference, 2009.
- FLEURY, M. T. L., & OLIVEIRA Jr., M. M. **Gestão Estratégica do Conhecimento**. Atlas, São Paulo, 2001.
- FLEURY, M. T. L. **Modelo de Gestão de Pessoas por Competências**. In: **Curso aberto em gestão de competências**, 1º, 1999, Anais. São Paulo: PROGEP FIA/FEA-USP, 1999.
- FREITAS, P.L.; & RODRIGUES, S.G. **A avaliação da confiabilidade de questionários: uma análise utilizando o coeficiente Alfa de Cronbach**. XII SIMPED, Bauru, 2005.
- GÜNTHER, H. **Como Elaborar um Questionário (Série: Planejamento de Pesquisa nas Ciências Sociais, N° 01)**. Brasília: UnB, Laboratório de Psicologia Ambiental, 2003.
- HODA, R., KRUCHTEN, P., NOBLE, J. et al. **Agility in context**. Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications, p. 74-88. New York, NY, USA, 2010.
- IEEE. **Guide to the Software Engineering Body of Knowledge (SWEBOK)**. 2004.
- JUDY, K. **Agile Values, Innovation and the Shortage of Women Software Development**. 45th Hawaii International Conference on System Sciences, 2012.
- JUDY, K., & Krumins-Beens, I. **Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership**. 41st Hawaii International Conference on System Sciences. 2008.

- KILIMNIK, Z.M., CASTILHO, I.V., & SANT'ANNA, A.S. **Carreiras em transformação e seus paradoxais reflexos nos indivíduos: Metáforas de carreira e de competências.** In: *Comportamento Organizacional e Gestão*, 2006, vol. 12, nº 2, p. 257-280.
- LEDESMA, R.; IBAÑES, G.M.; & MORA, P.V. **Análisis de consistencia interna mediante Alfa de Cronbach: un programa basado em gráficos dinámicos.** *Psico-USF*, v. 7, nº 2, p. 143-152, Jul./Dez. 2002.
- LEFFINGWELL, D. **Agile Software Requirements: Lean Requirements Practices for Teams, Programs and the Enterprise.** Addison-Wesley Professional, 2011.
- LEME, R. **Avaliação de desempenho com foco em competência: a base para a remuneração por competências.** Rio de Janeiro: Qualitymark, 2006.
- LORBER, A., & TIESZEN, S. **A Starting Point for Negotiations – Delivering with a Heterogeneous Team.** Agile Conference, 2012.
- LOWERY, M., & EVANS, M. **Scaling Product Ownership.** Agile Conference, 2007.
- MACHADO, C.A.F. **A-Risk: Um método para identificar e quantificar risco de prazo em projetos de desenvolvimento de software.** 2002. Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada. Curitiba, 2002.
- MAHNIC, V. **A Capstone Course on Agile Software Development Using Scrum.** *IEEE Transactions on Education*, v.55, nº 1, February, 2012.
- MAHNIC, V., & ROZANC, I. **Students' Perceptions of Scrum Practices.** MIPRO, 2012.
- MARCONI, M.A.; & LAKATOS, E.M. **Fundamentos de Metodologia Científica.** 5 ed., São Paulo: Atlas, 2003.
- MENCKE, R. **A Product Manager's Guide to Surviving the Big Bang Approach to Agile Transitions.** Agile Conference, 2008.
- MOE, N., DIGSOYR, T., & KVANGARDSNES, O. **Understanding Shared Leadership in Agile Development: A Case Study.** 42th Hawaii International Conference on System Sciences, 2009.
- MOON, A.M. **{“Come”} Together, Right Now – How the Songs of The Beatles Helped our Product Owners and Teams Live in Harmony.** Agile Conference, 2008.
- MORETTIN, P.A.; & BUSSAB, W.O. **Estatística Básica.** 5 ed. Ed. Saraiva, 2009.
- NERUR, S., MAHAPATRA, R., & MANGALARAJ, G. **Challenges of migrating to agile methodologies.** *Communications of the ACM*, 48(5), p. 72–78, 2005.
- NOTTONSON, K., & DELONG, K. **Crawl, Walk, Run: 4 Years of Agile Adoptin at {BabyCenter.com}.** Agile Conference, 2008.
- PAASIVAARA, M., HEIKKILA, V., & LASSENIUS, C. **Experiences in Scalling the Product Owner Role Large-Scale Globally Distributed Scrum.** *IEEE 7th International Conference On Global Software Engineering*, 2012.

- PAASIVAARA, M., LASSENIUNS, C., DAMIAN, D. et al. **Teacing Students Global Software Engineering, Skills using Distributed Scrum**. ICSE. San Francisco, CA, USA, 2013.
- PERKUSICH, M., ALMEIDA, H.O. de, & PERKUSICH, A. **A model to detect problems on scrum-based software development projetcs**. Proceedings of the 28th Annual ACM Symposium on Applied Compugint, p. 1037-1042, 2013.
- PERRY, T. **The Intermediate Customer Anti-Pattern**. Agile Conference, 2008.
- PICLHER, R. **Agile Product Management with Scrum: Creating Products that Customers Love**. Addison-Wesley, 2010.
- PMI. **Conjunto de Conhecimentos em Gerenciamento de Projetos (PMBok)**. Project Management Institute (PMI) – 3ª edição – 2004.
- QUAGLIA, E., & TOCANTINS, C. **Simulation Projects Management Using Scrum**. Winter Simulation Conference, 2011.
- RABECHINI JR., R.; & PESSOA, M. S. de P. **Um modelo estruturado de competências e maturidade em gerenciamento de projetos**. Prod., São Paulo , v. 15, n. 1, April, 2005.
- RACHEVA, Z., DANEVA, M., & SIKKEL, K. et al. **Business Value Is not only Dollars – Results from Case Study Research on Agile Software Projects**. Proceedings of the 11th International Conference on Product Focused Software Process Improvement, p.131-145. 2010.
- RISING, L. **Agile Meetings: Putting Frequent, Short Meetings to Work for Your Team**. In: Software Testing and Quality Engineering Magazine, 2002.
- SCHUMAN, H. & KALTON, G. Survey methods. Em LINDZEY, G., & ARONSON, E. **Handbook of social psychology**. 3.ed., vol. 1. New York: Random House, 1985.
- SCHWABER, K. **Agile Project Management with Scrum**. Microsoft Press, 2004.
- SCHWABER, K., & BEEDLE, M. **Agile Software Development with Scrum**. Prentice Hall, 2001.
- SHORE, J. **A Arte do Desenvolvimento Ágil**. Alta Books, 2008.
- SIMON, F.O. **Habilidades e Competências em Engenharia: Criação e Validação de um Instrumento**. Dissertação de Mestrado. Universidade Estadual de Campinas, Faculdade de Educação, 2004.
- SINGH, M. **U-SCRUM: An Agile Methodology for Promoting Usability**. Agile Conference, 2008.
- SOARES, G. H. F. S. **Proposta de um conjunto de competências para um time ágil**. Monografia (Graduação) – Universidade de Brasília Faculdade do Gama, Brasília, 2014.
- STE-COIX, A., & EASTON, A. **The Product Owner Team**. Agile Conference, 2008.

SUTHERLAND, J., & SCHWABER, K. **The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework**. SCRUM Training Institute, 2011.

ZIERIS, F., & SALINGER, S. **Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices**. IEEE 8th International Conference on Global Software Engineering, 2013.

APÊNDICE I – PROTOCOLO DA REVISÃO SISTEMÁTICA

1. FORMULAÇÃO DA QUESTÃO

1.1. FOCO DA QUESTÃO

Levantar os principais registros relacionados às características, responsabilidades e atributos de um *Product Owner*, buscando, a partir deles, identificar uma lacuna no estado da arte e atuar em cima dela.

1.2. QUALIDADE E AMPLITUDE DA QUESTÃO

1.2.1. Problema:

Em um desenvolvimento de software que utiliza o *framework Scrum*, existe a delegação de um papel a uma pessoa denominado *Product Owner*, cujos atributos e responsabilidades estão atrelados a competências ainda não explorados pela literatura. Essa lacuna traz riscos perante a contratação de desenvolvimento de software que utiliza metodologias ágeis.

1.2.2. Questão:

- Quais as competências de um *Product Owner*, para que ele desempenhe seu papel da melhor forma possível?

1.2.3. Palavras-chave e sinônimos

- *product owner, po;*
- *software agile development, scrum;*
- *competences;*
- *attributes, responsibilities.*

2. SELEÇÃO DAS FONTES

2.1. CRITÉRIOS DE SELEÇÃO DAS FONTES

- Disponibilidade para consulta na Internet;
- Disponibilidade para consulta na Biblioteca Central da Universidade de Brasília;
- Presente nas pesquisas utilizando as palavras-chave;

- Presente nas referências bibliográficas de fontes aproveitadas.

2.2. IDIOMA DAS FONTES

- Inglês.

2.3. IDENTIFICAÇÃO DAS FONTES

2.3.1. Base de dados IEEE e ACM

2.3.1.1. Strings de busca, utilizada na opção “Advanced Search” (IEEE)

- ("Document Title":product owner)
- (((agile) AND (scrum AND "product owner" OR po)))

2.3.1.2. Strings de busca, utilizada no método comum de pesquisa

- (((agile) AND (scrum AND "product owner" OR po)))

3. SELEÇÃO DOS DOCUMENTOS

3.1. DEFINIÇÃO DOS DOCUMENTOS

3.1.1. Critérios de inclusão

- O documento está disponível na internet;
- O documento possui menção ao *Product Owner* no título, *abstract* e/ou sumário;
- O documento possui menção ao *framework* Scrum;
- O conteúdo do documento possui correlação com metodologias ágeis para desenvolvimento de software;
- O conteúdo apresenta características/atributos do *Product Owner*.

3.1.2. Critérios de exclusão

- O documento é uma introdução de anal de conferência;
- O documento não possui estrutura adequada para um artigo científico;
- O documento não menciona o *Product Owner*;
- O documento não apresenta características/atributos de um *Product Owner*.

3.2. Procedimento para a seleção dos documentos

As *strings* de busca devem ser utilizadas nas fontes de pesquisa escolhidas. Com o resultado da pesquisa em mãos, devem ser lidos os títulos, *abstracts* e palavras-chave, a fim de aplicar os critérios de inclusão e exclusão definidos na Seção (3.1) deste Protocolo. Os artigos que não se encaixarem em nenhum critério de exclusão e possuírem os critérios de inclusão deverão ser lidos.

4. EXTRAÇÃO DE INFORMAÇÃO E SINTETIZAÇÃO DOS DADOS

Os documentos serão compreendidos, avaliados de acordo com seu grau de relevância e alinhamento junto às perguntas de pesquisa e sintetizados e, por fim, terem suas ideias condensadas.

APÊNDICE II – FUNDAMENTAÇÃO DOS ATRIBUTOS POR ARTIGO

Quadro 7. Fundamentação dos atributos por artigo (O autor).

Artigo	Atributo	Fundamentação
<i>A model to detect problems on scrum-based software development projects.</i>	Comunicativo	Fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>Business value is not only dollars; results from case study research on agile software projects.</i>	Comunicativo	Fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado
	Levantar requisitos	Fundamentado
<i>Simulation projects management using Scrum.</i>	Comunicativo	Pouco fundamentado
	Autoridade	Fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Fundamentado
	Priorizar <i>backlog</i>	Fundamentado
	Escrever <i>user stories</i>	Fundamentado
	Validar produto	Fundamentado
<i>Teaching students global</i>	Comunicativo	Bem fundamentado

Artigo	Atributo	Fundamentação
<i>software engineering skills using distributed Scrum.</i>	Autoridade	Bem fundamentado
	Disponibilidade	Bem fundamentado
	Levantar requisitos	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
<i>When user experience met agile: a case study.</i>	Comunicativo	Bem fundamentado
	Business-savvy	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Levantar requisitos	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
<i>A Capstone Course on Agile Software Development Using Scrum.</i>	Autoridade	Fundamentado
	Capacidade de decisão	Fundamentado
	Disponibilidade	Pouco fundamentado
	Escrever <i>user stories</i>	Fundamentado
	Levantar requisitos	Fundamentado
	Priorizar <i>backlog</i>	Fundamentado
	Validar produto	Fundamentado
	Computar <i>velocity</i>	Pouco fundamentado
<i>A Product Manager's Guide to Surviving the Big Bang Approach to Agile Transitions.</i>	Comunicativo	Fundamentado
	Capacidade de decisão	Fundamentado
	Business-savvy	Fundamentado

Artigo	Atributo	Fundamentação
	Definir métricas	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Planejar <i>sprint</i>	Bem fundamentado
	Levantar requisitos	Bem fundamentado
	Validar produto	Bem fundamentado
	Validar protótipos	Bem fundamentado
<i>A Starting Point for Negotiations & Delivering with a Heterogeneous Team.</i>	Autoridade	Pouco fundamentado
	Capacidade de decisão	Pouco fundamentado
	Business-savvy	Pouco fundamentado
	Planejar <i>sprint</i>	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>Agile Values, Innovation and the Shortage of Women Software Developers.</i>	Comunicativo	Pouco fundamentado
<i>Crawl, Walk, Run: 4 Years of Agile Adoption at {BabyCenter.com}.</i>	Comunicativo	Bem fundamentado
	Autoridade	Bem fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado

Artigo	Atributo	Fundamentação
	Disponibilidade	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Monitorar ROI	Bem fundamentado
	Validar produto	Bem fundamentado
	Levantar requisitos	Bem fundamentado
<i>Doing Scrum Rather Than Being Agile: A Case Study on Actual Nearshoring Practices.</i>	Comunicativo	Fundamentado
	Autoridade	Bem fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado
	Disponibilidade	Fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>From Cradle to Sprint: Creating a Full-Lifecycle Request Pipeline at Nationwide Insurance.</i>	Autoridade	Pouco fundamentado
	Capacidade de decisão	Pouco fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>Scaling Product Ownership.</i>	Autoridade	Bem fundamentado
	Business-savvy	Bem fundamentado
	Disponibilidade	Bem fundamentado

Artigo	Atributo	Fundamentação
	Priorizar <i>backlog</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>Students' perceptions of Scrum practices.</i>	Comunicativo	Bem fundamentado
	Autoridade	Fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado
	Disponibilidade	Fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>The Intermediate Customer Anti-Pattern.</i>	Autoridade	Pouco fundamentado
	Capacidade de decisão	Pouco fundamentado
	Disponibilidade	Pouco fundamentado
	Levantar requisitos	Fundamentado
<i>Understanding Sharing Leadership in Agile Development: A Case Study.</i>	Comunicativo	Bem fundamentado
	Autoridade	Bem fundamentado
	Disponibilidade	Fundamentado
	Gerenciar projeto	Fundamentado
	Priorizar <i>backlog</i>	Fundamentado
	Validar produto	Fundamentado
	Levantar requisitos	Fundamentado
<i>U-$\{$SCRUM:$\}$ An Agile</i>	Comunicativo	Pouco fundamentado

Artigo	Atributo	Fundamentação
<i>Methodology for Promoting Usability.</i>	Business-savvy	Pouco fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Validar produto	Bem fundamentado
	Traçar estratégia de marketing e venda	Fundamentado
	Validar produto	Fundamentado
	Manter relações de negócio	Fundamentado
<i>Agile Method Tailoring in Distributed Enterprises: Product Owner Teams.</i>	Comunicativo	Fundamentado
	Autoridade	Bem fundamentado
	Capacidade de decisão	Fundamentado
	Business-savvy	Bem fundamentado
	Levantar requisitos	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Validar produto	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado
	Definir aspectos técnicos e arquiteturais do software	Bem fundamentado
<i>Experiences in Scaling the Product Owner Role in Large-Scale Globally Distributed Scrum.</i>	Comunicativo	Bem fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado

Artigo	Atributo	Fundamentação
	Priorizar <i>backlog</i>	Bem fundamentado
	Monitorar ROI	Bem fundamentado
	Levantar requisitos	Bem fundamentado
	Planejar <i>sprint</i>	Bem fundamentado
	Validar produto	Bem fundamentado
<i>Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership.</i>	Comunicativo	Fundamentado
	Capacidade de decisão	Bem fundamentado
	Business-savvy	Bem fundamentado
	Priorizar <i>backlog</i>	Fundamentado
	Monitorar ROI	Fundamentado
	Levantar requisitos	Fundamentado
	Validar produto	Fundamentado
<i>The Product Owner Team.</i>	Comunicativo	Fundamentado
	Autoridade	Fundamentado
	Business-savvy	Fundamentado
	Disponibilidade	Pouco fundamentado
	Levantar requisitos	Fundamentado
	Definir aspectos técnicos e arquiteturais do software	Fundamentado
	Escrever <i>user stories</i>	Fundamentado
	Priorizar <i>backlog</i>	Fundamentado

Artigo	Atributo	Fundamentação
	Validar produto	Fundamentado
<i>Agility in context.</i>	Escrever <i>user stories</i>	Bem fundamentado
	Levantar requisitos	Fundamentado
	Priorizar <i>backlog</i>	Fundamentado
<i>Agile Development with Security Engineering Activities.</i>	Levantar requisitos	Bem fundamentado
<i>“Come Together, Right Now” – How the Songs of The Beatles Help our Product Owners and Teams Live in Harmony</i>	Levantar requisitos	Bem fundamentado
	Priorizar <i>backlog</i>	Bem fundamentado
	Monitorar ROI	Bem fundamentado
	Manter visão do projeto	Bem fundamentado
	Escrever <i>user stories</i>	Bem fundamentado

APÊNDICE III – CONJUNTO INICIAL DE COMPETÊNCIAS

As competências presentes no Quadro (9) possuem um identificador único cada (ID) denominado **CT**, oriundo de **Competência Técnica**, ou **CC**, proveniente de **Competência Comportamental**, que possui a finalidade de facilitar a referência textual.

Quadro 8. Competências para um *Product Owner* (O autor).

ID	Competência	Responsabilidade
CT1	Dominar técnicas de planejamento de desenvolvimento ágil.	<ul style="list-style-type: none"> • Atualizar o <i>Product Backlog</i>; • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>; • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>.
CT2	Dominar o negócio em que o projeto está inserido.	<ul style="list-style-type: none"> • Monitorar o ROI (retorno sobre investimento) do produto; • Extrair os requisitos do software a partir das necessidades dos <i>stakeholders</i>; • Expor sua impressão sobre o produto; • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>.
CT3	Dominar técnicas de priorização do <i>Product Backlog</i> .	<ul style="list-style-type: none"> • Atualizar o <i>Product Backlog</i>; • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>;

ID	Competência	Responsabilidade
		<ul style="list-style-type: none"> • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>; • Levar em consideração o ROI (retorno sobre investimento) do produto.
CT4	Dominar adequadamente a técnica de levantamento de requisitos ágeis.	<ul style="list-style-type: none"> • Extrair os requisitos do software a partir das necessidades dos <i>stakeholders</i>.
CT5	Dominar a técnica de escrita de <i>user story</i> .	<ul style="list-style-type: none"> • Reproduzir os requisitos levantados com os <i>stakeholders</i> no formato de <i>user stories</i>.
CT6	Dominar a técnica de definição de critérios de aceitação.	<ul style="list-style-type: none"> • Verificar, com base nos critérios de aceitação, se as <i>user stories</i> foram implementadas corretamente. • Construir critérios de avaliação para a validação das <i>user stories</i>.
CT7	Conhecer e aplicar adequadamente as técnicas de monitoramento do ROI (retorno sobre investimento) do produto.	<ul style="list-style-type: none"> • Levar em consideração o ROI (retorno sobre investimento) do produto; • Monitorar o ROI (retorno sobre investimento) do produto.
CT8	Dominar as técnicas de comunicação.	<ul style="list-style-type: none"> • Discutir melhorias quanto às suas atribuições, juntamente com o time, de modo a otimizar o desempenho da equipe como um todo; • Elucidar as dúvidas referentes às

ID	Competência	Responsabilidade
		<p><i>user stories</i>;</p> <ul style="list-style-type: none"> • Expor sua impressão sobre o produto; • Extrair os requisitos do software a partir das necessidades dos <i>stakeholders</i>; • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>.
CT9	Possuir visão estratégica	<ul style="list-style-type: none"> • Caso haja mudanças, priorizá-las juntamente com os <i>stakeholders</i>; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>; • Discutir melhorias quanto às suas atribuições, juntamente com o time, de modo a otimizar o desempenho da equipe como um todo; • Atualizar o <i>Product Backlog</i>; • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>; • Monitorar o ROI (retorno sobre investimento) do produto.
CT10	Possuir visão sistêmica.	<ul style="list-style-type: none"> • Discutir melhorias quanto às suas atribuições, juntamente com o time, de modo a otimizar o desempenho da equipe como um todo; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões

ID	Competência	Responsabilidade
		acerca do andamento da <i>sprint</i> .
CT11	Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento do software.	<ul style="list-style-type: none"> • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>.
CT12	Possuir conhecimento de teste funcional.	<ul style="list-style-type: none"> • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>.
CT13	Possuir conhecimento de refatoração.	<ul style="list-style-type: none"> • Priorizar <i>user stories</i> a fim de formar o <i>Sprint Backlog</i>; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>.
CC1	Possuir a coragem para tomar uma decisão.	<ul style="list-style-type: none"> • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>.
CC2	Possuir a capacidade de admitir quando alguém está errado.	<ul style="list-style-type: none"> • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>; • Discutir melhorias quanto às suas atribuições, juntamente com o time, de modo a otimizar o desempenho da equipe como um todo.
CC3	Possuir a capacidade de estabelecer uma relação de	<ul style="list-style-type: none"> • Ter atitudes condizentes com os valores que prega;

ID	Competência	Responsabilidade
	confiança com as pessoas que relaciona.	<ul style="list-style-type: none"> • Demonstrar comprometimento com o projeto, sendo solícito quando requisitado.
CC4	Possuir a disposição para ajudar a equipe.	<ul style="list-style-type: none"> • Demonstrar comprometimento com o projeto, sendo solícito quando requisitado; • Caso necessário, tomar decisões acerca da implementação de <i>user stories</i>; • Caso necessário, tomar decisões acerca do andamento da <i>sprint</i>; • Elucidar as dúvidas referentes às <i>user stories</i>; • Expor sua impressão sobre o produto.

APÊNDICE IV – PRIMEIRA VERSÃO DO QUESTIONÁRIO

Competências de um Product Owner

Este questionário faz parte de um Trabalho de Conclusão de Curso de graduação em Engenharia de Software da Universidade de Brasília (UnB), o qual se busca a proposição de um conjunto inicial de competências de um Product Owner (PO).

O objetivo desta pesquisa é validar a proposta de um conjunto de competências técnicas e comportamentais para um PO.

*Obrigatório

Competências Técnicas

Entende-se por Competências Técnicas "tudo o que o profissional precisa ter para desempenhar seu papel, (...) o 'Saber' e o 'Saber Fazer'" (LEME, 2006).

Por favor, a partir da definição apresentada, indique o grau de importância para cada competência técnica apresentada abaixo, sendo os graus:

- 1: Nada importante
- 2: Pouco importante
- 3: Mediamente importante
- 4: Muito importante
- 5: Extremamente importante

Idealmente um Product Owner deve possuir as seguintes competências técnicas:

Dominar técnicas de planejamento de desenvolvimento ágil *

1 2 3 4 5

Nada importante Extremamente importante

Dominar o negócio em que o projeto está inserido *

1 2 3 4 5

Nada importante Extremamente importante

Dominar técnicas de priorização do Product Backlog *

1 2 3 4 5

Nada importante Extremamente importante

Dominar técnicas de levantamento de requisitos ágeis *

1 2 3 4 5

Nada importante Extremamente importante**Dominar a técnica de escrita de user story ***

1 2 3 4 5

Nada importante Extremamente importante**Dominar a técnica da definição dos critérios de aceitação ***

1 2 3 4 5

Nada importante Extremamente importante**Dominar as técnicas de monitoramento do ROI (Return on Investment) do produto ***

1 2 3 4 5

Nada importante Extremamente importante**Dominar as técnicas de comunicação ***

1 2 3 4 5

Nada importante Extremamente importante**Possuir visão estratégica ***

1 2 3 4 5

Nada importante Extremamente importante**Possuir visão sistêmica ***

1 2 3 4 5

Nada importante Extremamente importante**Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento do software ***

1 2 3 4 5

Nada importante Extremamente importante

Possuir conhecimento técnico de teste funcional *

1 2 3 4 5

Nada importante Extremamente importante**Possuir conhecimento técnico de refatoração ***

1 2 3 4 5

Nada importante Extremamente importante**Competências Comportamentais**

Entende-se por Competências Comportamentais sendo "o diferencial competitivo de cada profissional e têm impacto em seus resultados, (...) 'Querer Fazer'" (LEME, 2006).

Por favor, a partir da definição apresentada, indique o grau de importância para cada competência comportamental apresentada abaixo, sendo os graus:

- 1: Nada importante
- 2: Pouco importante
- 3: Mediamente importante
- 4: Muito importante
- 5: Extremamente importante

Idealmente, um Product Owner deve possuir as seguintes competências comportamentais:

Possuir a coragem para tomar uma decisão *

1 2 3 4 5

Nada importante Extremamente importante**Possuir a capacidade de admitir quando está errado ***

1 2 3 4 5

Nada importante Extremamente importante**Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que se relaciona ***

1 2 3 4 5

Nada importante Extremamente importante

Possuir a disposição para ajudar a equipe *

1 2 3 4 5

Nada importante Extremamente importante

Dados do Entrevistado

Sexo *

- Masculino
 Feminino

Qual sua experiência (tempo) no desenvolvimento ágil de software utilizando o framework Scrum? *

Já exerceu o papel de PO? *

- Sim
 Não

Se sim, por quanto tempo? *

Qual(is) outro(s) papel(éis) desempenhado(s)? *

- ScrumMaster
 Time
 Stakeholder
 Patrocinador
 Estudante

Por quanto tempo exerceu? *

Qual(is) a(s) aplicação(ões) dessa experiência com Scrum? *

- Setor público
 Iniciativa privada
 Acadêmica

Atualmente, qual sua área de atuação no desenvolvimento de software? *

- Setor público
- Iniciativa privada
- Acadêmica

Nos dias de hoje, segue utilizando Scrum? *

- Sim
- Não

Qual sua localização (apenas UF)? *

APÊNDICE V – SEGUNDA VERSÃO DO QUESTIONÁRIO

Competências de um Product Owner

Este questionário faz parte de um trabalho em Engenharia de Software da Universidade de Brasília (UnB), o qual se busca a proposição de um conjunto inicial de competências de um Product Owner (PO).

O objetivo desta pesquisa é validar a proposta de um conjunto de competências técnicas e comportamentais para um PO.

*Obrigatório

Competências Técnicas

Entende-se por Competências Técnicas "tudo o que o profissional precisa ter para desempenhar seu papel, (...) o 'Saber' e o 'Saber Fazer'" (LEME, 2006).

Por favor, a partir da definição apresentada, indique o grau de importância para cada competência técnica apresentada abaixo, sendo os graus:

- 1: Nada importante
- 2: Pouco importante
- 3: Mediamente importante
- 4: Muito importante
- 5: Extremamente importante

Idealmente um Product Owner deve possuir as seguintes competências técnicas:

1 - Dominar técnicas de planejamento de desenvolvimento ágil *

Relacionada ao domínio, por exemplo, das técnicas de planning poker, velocity, burndown, etc.

1 2 3 4 5

Nada importante Extremamente importante

2 - Dominar o negócio em que o projeto está inserido *

Relacionada a ter um conhecimento consistente do domínio do negócio, no contexto operacional, e ser capaz de transmiti-lo ao time de desenvolvimento.

1 2 3 4 5

Nada importante Extremamente importante

3 - Dominar técnicas de priorização do Product Backlog *

Relacionada à capacidade de definir quais user stories (escopo) são mais importantes para o projeto.

1 2 3 4 5

Nada importante Extremamente importante

4 - Dominar a técnica de escrita de user story *

Relacionada à capacidade de extrair requisitos do cliente e transformá-los em user stories.

1 2 3 4 5

Nada importante Extremamente importante

5 - Dominar a técnica de validação de requisitos *

Relacionada à capacidade de validar as funcionalidades implementadas. Essa validação pode ser feita por meio de critérios de aceitação, os quais devem ser testáveis.

1 2 3 4 5

Nada importante Extremamente importante

6 - Dominar as técnicas de monitoramento do ROI (Return on Investment) do produto *

Relacionada à capacidade de garantir que o investimento feito no projeto seja consumido conforme o planejado, e que os mesmos trarão os benefícios esperados à organização.

1 2 3 4 5

Nada importante Extremamente importante

7 - Dominar as técnicas de comunicação *

Relacionada à necessidade de se comunicar com pessoas de camadas diferentes (estratégico, tático e operacional).

1 2 3 4 5

Nada importante Extremamente importante

8 - Possuir visão estratégica *

Relacionada ao domínio do negócio no nível estratégico.

1 2 3 4 5

Nada importante Extremamente importante

9 - Possuir visão sistêmica *

Relacionada a possuir a compreensão do contexto de negócio em que o projeto está envolvido, suas partes (estratégico, tático e operacional) e a relação existente entre elas.

1 2 3 4 5

Nada importante Extremamente importante

10 - Possuir conhecimento das tecnologias utilizadas no desenvolvimento do software *

Refere-se à necessidade de, por exemplo, priorizar certas user stories do Product Backlog na primeira sprint, uma vez que estas forçam a implementação da arquitetura.

1 2 3 4 5

Nada importante Extremamente importante

11 - Possuir conhecimento do impacto de teste funcional no projeto *

Refere-se à necessidade de, por exemplo, priorizar a implementação de testes em vez de novas funcionalidades, em um momento que a cobertura de testes do código é considerada baixa.

1 2 3 4 5

Nada importante Extremamente importante

12 - Possuir conhecimento do impacto da refatoração no projeto *

Refere-se à necessidade de, por exemplo, priorizar a refatoração do código em vez de implementar novas funcionalidades, a fim de prezar pela qualidade do código. Essa decisão pode ser baseada por meio de métricas.

1 2 3 4 5

Nada importante Extremamente importante

13 - Possuir conhecimento dos ritos e cerimônias realizadas no Scrum *

Refere-se à necessidade de conhecer, por exemplo, as características de Sprint Planning Meeting, Sprint Daily Meeting, etc.

1 2 3 4 5

Nada importante Extremamente importante

14 - Possuir o conhecimento dos valores ágeis *

Refere-se à necessidade de saber que, por exemplo, para um projeto ágil, a "colaboração do cliente mais que negociação de contrato" e "indivíduos e interação entre eles é mais que processos e ferramentas".

1 2 3 4 5

Nada importante Extremamente importante

15 - Possuir o conhecimento dos princípios ágeis *

Refere-se à necessidade de saber que, por exemplo, "conhecedores do negócio e desenvolvedores devem trabalhar juntos diariamente durante o projeto", "simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser", "o método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara", etc.

1 2 3 4 5

 Nada importante Extremamente importante

16 - Dominar técnicas de negociação *

Refere-se à necessidade de, por exemplo, negociar variação de prazo e escopo com o cliente e time de desenvolvimento.

1 2 3 4 5

 Nada importante Extremamente importante

Competências Comportamentais

Entende-se por Competências Comportamentais sendo "o diferencial competitivo de cada profissional e têm impacto em seus resultados, (...) 'Querer Fazer'" (LEME, 2006).

Idealmente, um Product Owner deve possuir as seguintes competências comportamentais:

1 - Possuir a coragem para tomar uma decisão *

Refere-se a ser capaz, e não hesitar, na tomada de decisão.

1 2 3 4 5

 Nada importante Extremamente importante

2 - Possuir a capacidade de admitir quando está errado *

Ser capaz e ter a humildade para reconhecer perante as pessoas envolvidas no projeto, que errou. Essa atitude visa promover o aprendizado (próprio e dos envolvidos).

1 2 3 4 5

 Nada importante Extremamente importante

3 - Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que se relaciona *

Ser sincero e responsável com as pessoas e com os compromissos assumidos.

1 2 3 4 5

 Nada importante Extremamente importante

4 - Possuir a disposição e a habilidade em ajudar a equipe *

Refere-se à necessidade de, por exemplo, sanar dúvidas relacionadas ao negócio objetivando esclarecer user stories e seus critérios de aceitação.

1 2 3 4 5

Nada importante Extremamente importante

5 - Possuir a habilidade de se adequar a diferentes contextos *

Refere-se à necessidade do PO em ter habilidades para transitar do nível estratégico ao operacional, bem como com o time de desenvolvimento, durante todo o projeto.

1 2 3 4 5

Nada importante Extremamente importante

6 - Possuir a habilidade de resolver conflitos *

Refere-se à necessidade do PO em resolver, por exemplo, conflitos interpessoais e conflitos de estratégia que possam vir a prejudicar o andamento e o sucesso do projeto.

1 2 3 4 5

Nada importante Extremamente importante

Dados do Entrevistado

Sexo *

- Masculino
 Feminino

Qual sua experiência (tempo, em anos ou meses) no desenvolvimento ágil de software utilizando o framework Scrum? ***Já exerceu o papel de PO? ***

- Sim
 Não

Se sim, por quanto tempo (em anos ou meses)? *

Se não, por favor, insira 0.

Essa experiência como PO foi positiva? *

- Sim
- Não
- Não se aplica

Qual(is) outro(s) papel(éis) desempenhado(s)? *

- ScrumMaster
- Time
- Stakeholder
- Patrocinador
- Estudante

Por quanto tempo (total, em anos ou meses) exerceu? *

Em qual(is) contexto(s) essa experiência com Scrum foi aplicada? *

- Setor público
- Iniciativa privada
- Área acadêmica

Atualmente, qual sua área de atuação no desenvolvimento de software? *

- Setor público
- Iniciativa privada
- Área acadêmica

Nos dias de hoje, segue utilizando Scrum? *

- Sim
- Não

Qual sua localização (apenas UF)? *

Sugestões, críticas e avaliações

Caso julgue necessário, escreva no campo abaixo a sugestão de novas competências ou críticas sobre o trabalho.

Se desejar receber o resultado deste questionário, deixe seu e-mail abaixo

APÊNDICE VI – RESULTADOS DO QUESTIONÁRIO

Competências Técnicas

1 - Dominar técnicas de planejamento de desenvolvimento ágil

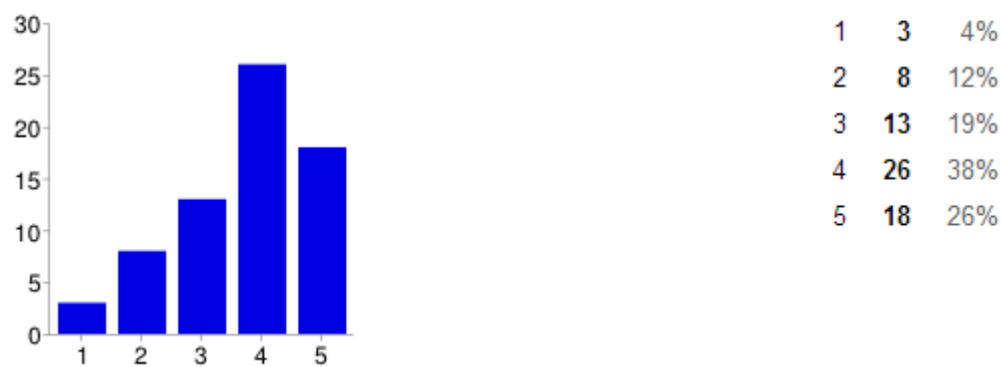


Gráfico 8. Resultado da CT1 (O autor).

2 - Dominar o negócio em que o projeto está inserido

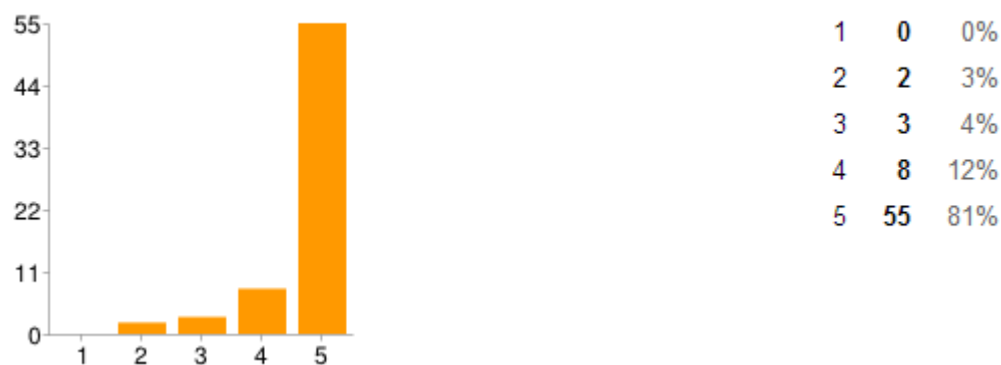


Gráfico 9. Resultado da CT2 (O autor).

3 - Dominar técnicas de priorização do Product Backlog

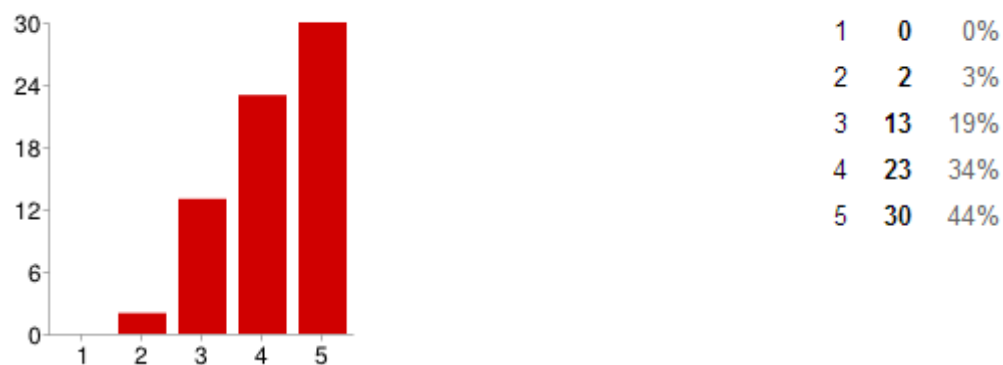


Gráfico 10. Resultado da CT3 (O autor).

4 - Dominar a técnica de escrita de user story

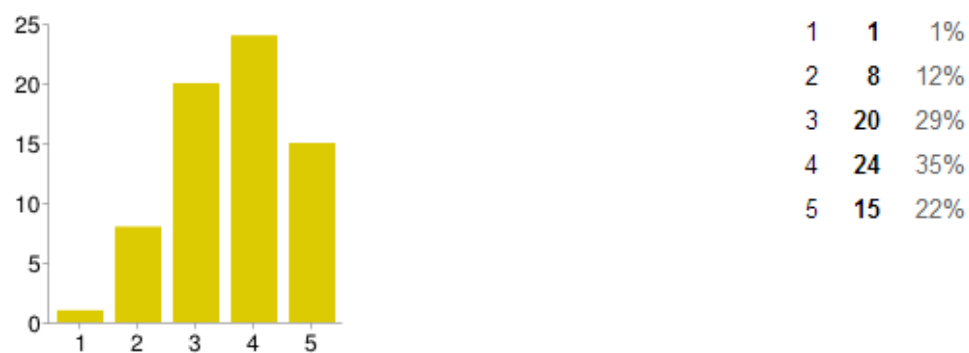


Gráfico 11. Resultado da CT4 (O autor).

5 - Dominar a técnica de validação de requisitos

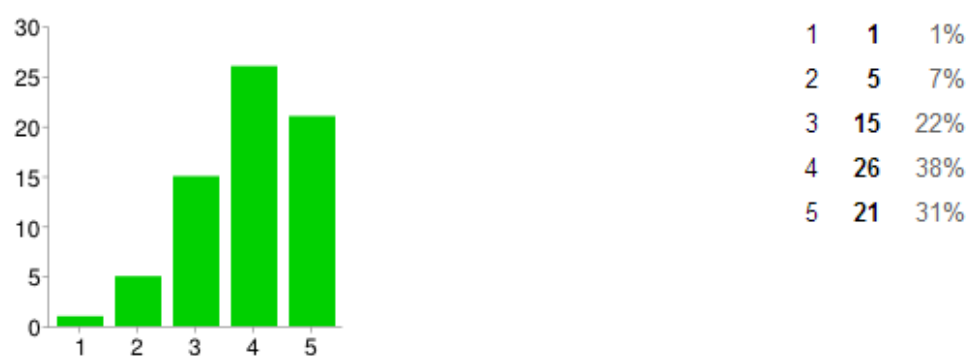


Gráfico 12. Resultado da CT5 (O autor).

6 - Dominar as técnicas de monitoramento do ROI (Return on Investment) do produto



Gráfico 13. Resultado da CT6 (O autor).

7 - Dominar as técnicas de comunicação



Gráfico 14. Resultado da CT7 (O autor).

8 - Possuir visão estratégica



Gráfico 15. Resultado da CT8 (O autor).

9 - Possuir visão sistêmica



Gráfico 16. Resultado da CT9 (O autor).

10 - Possuir conhecimento das tecnologias utilizadas no desenvolvimento do software

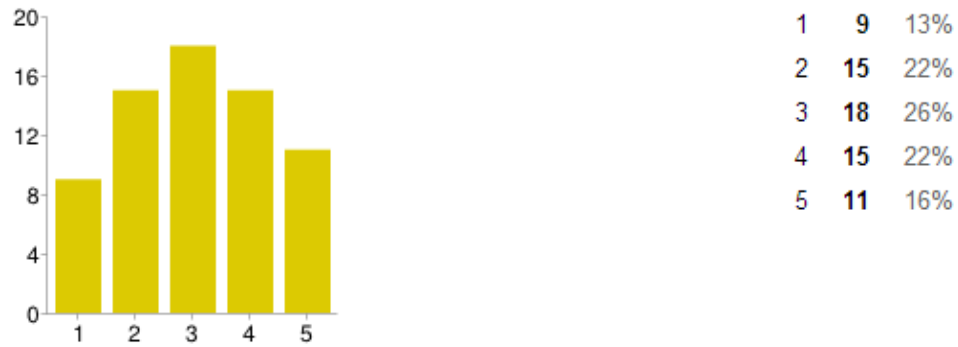


Gráfico 17. Resultado da CT10 (O autor).

11 - Possuir conhecimento do impacto de teste funcional no projeto

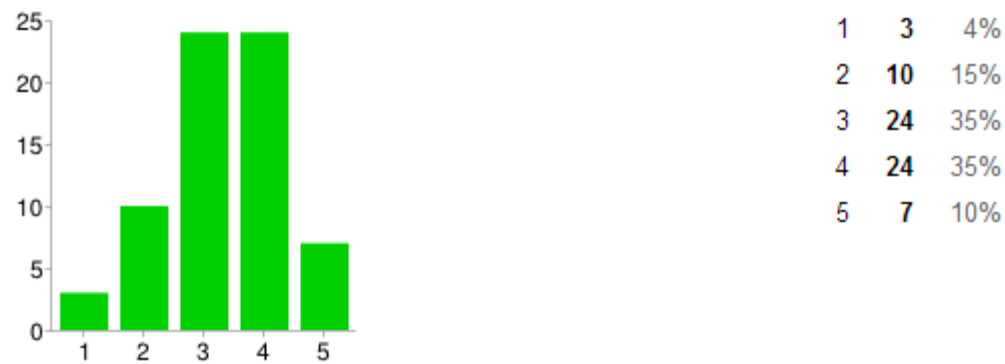


Gráfico 18. Resultado da CT11 (O autor).

12 - Possuir conhecimento do impacto da refatoração no projeto



Gráfico 19. Resultado da CT12 (O autor).

13 - Possuir conhecimento dos ritos e cerimônias realizadas no Scrum

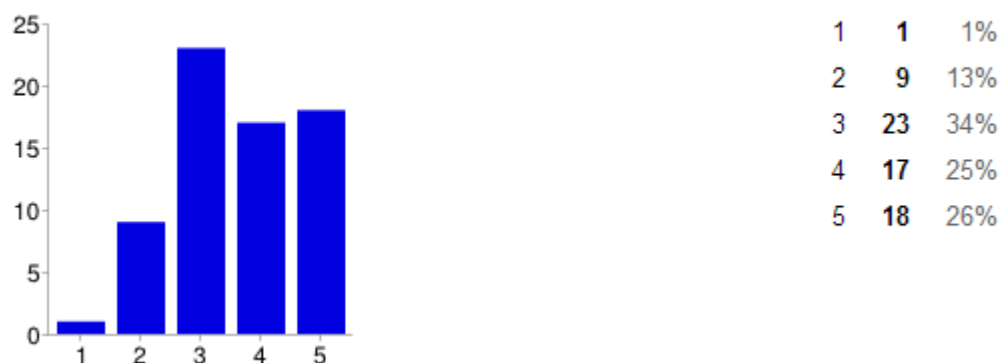


Gráfico 20. Resultado da CT13 (O autor).

14 - Possuir o conhecimento dos valores ágeis



Gráfico 21. Resultado da CT14 (O autor).

15 - Possuir o conhecimento dos princípios ágeis



Gráfico 22. Resultado da CT15 (O autor).

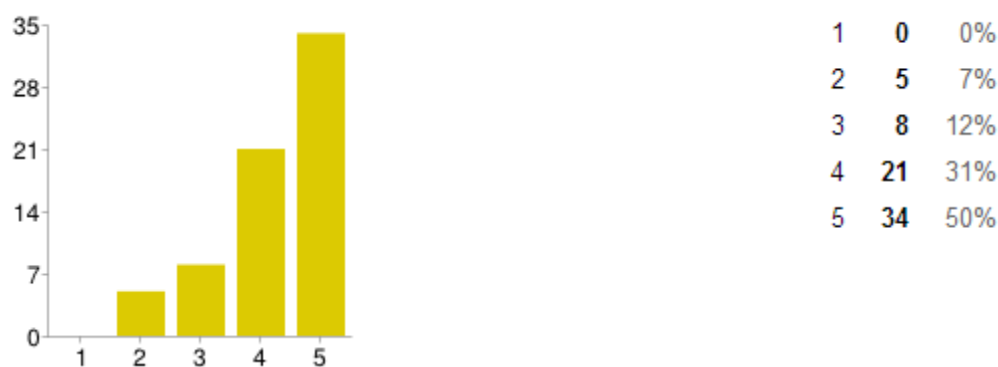
16 - Dominar técnicas de negociação

Gráfico 23. Resultado da CT16 (O autor).

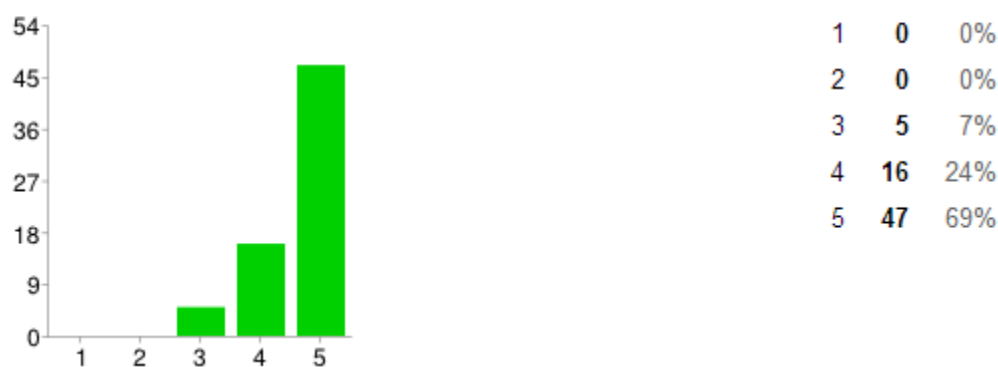
Competências Comportamentais**1 - Possuir a coragem para tomar uma decisão**

Gráfico 24. Resultado da CC1 (O autor).

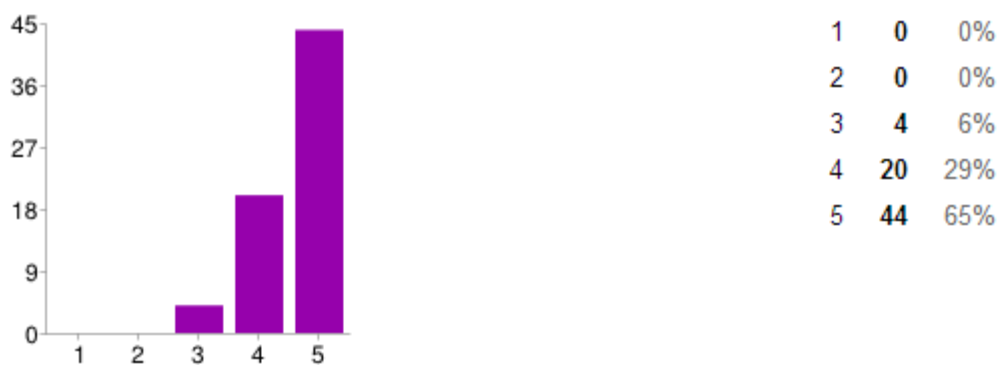
2 - Possuir a capacidade de admitir quando está errado

Gráfico 25. Resultado da CC2 (O autor).

3 - Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que se relaciona

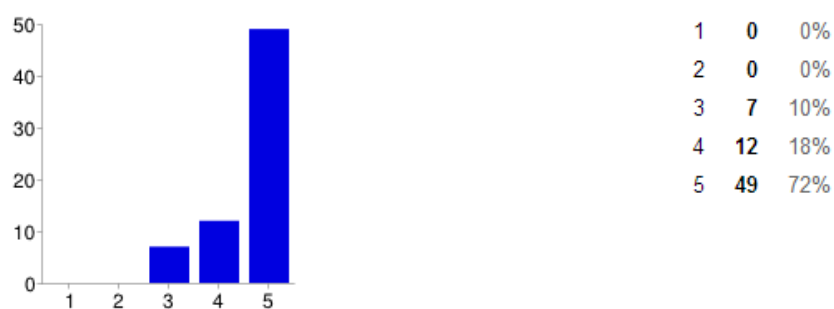


Gráfico 26. Resultado da CC3 (O autor).

4 - Possuir a disposição e a habilidade em ajudar a equipe

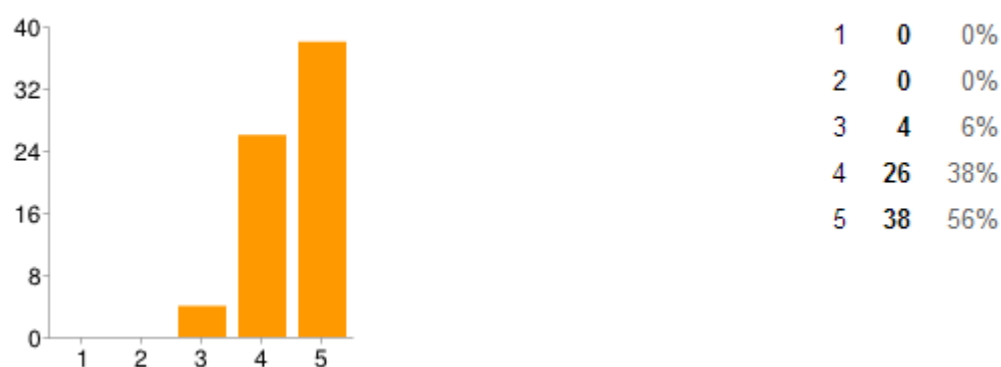


Gráfico 27. Resultado da CC4 (O autor).

5 - Possuir a habilidade de se adequar a diferentes contextos



Gráfico 28. Resultado da CC5 (O autor).

6 - Possuir a habilidade de resolver conflitos



Gráfico 29. Resultado da CC6 (O autor).

Dados do Entrevistado

Sexo

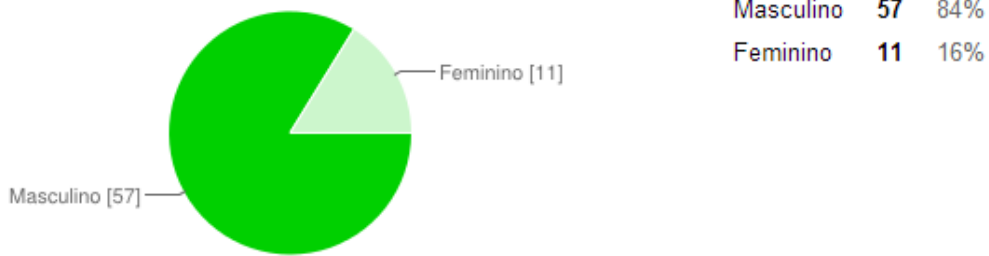
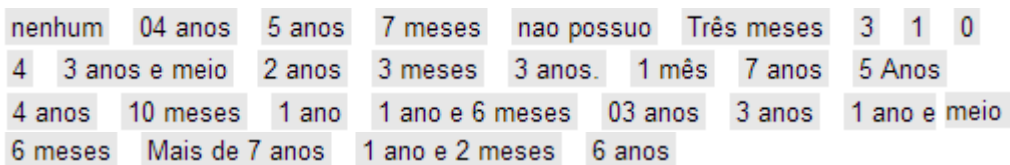


Gráfico 30. Sexo do respondente (O autor).

Qual sua experiência (tempo, em anos ou meses) no desenvolvimento ágil de software utilizando o framework Scrum?



Já exerceu o papel de PO?

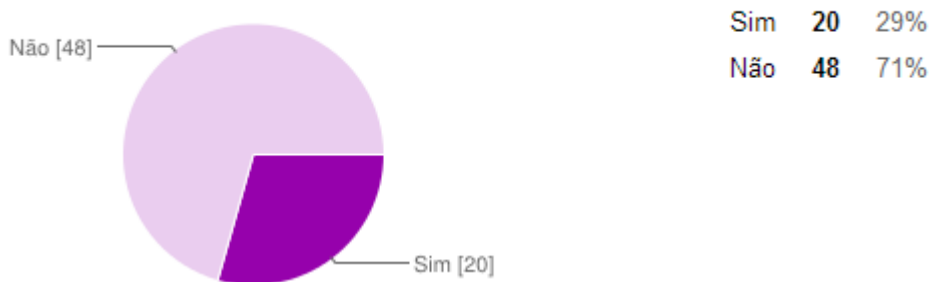
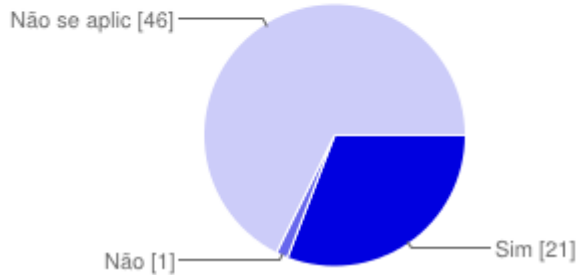


Gráfico 31. Atuação como PO (O autor).

Se sim, por quanto tempo (em anos ou meses)?

seis meses 3 1 0 6 durante 4 anos 2 anos 4 meses
 3 meses 2 meses 1 ano 3 anos 1 ano e meio 6 meses

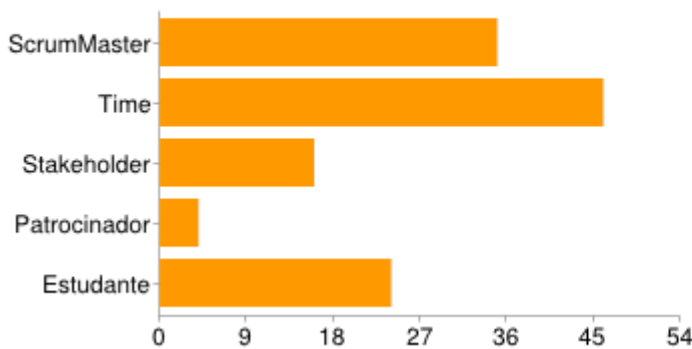
Essa experiência como PO foi positiva?



Sim	21	31%
Não	1	1%
Não se aplica	46	68%

Gráfico 32. Caráter da experiência como PO (O autor).

Qual(is) outro(s) papel(éis) desempenhado(s)?



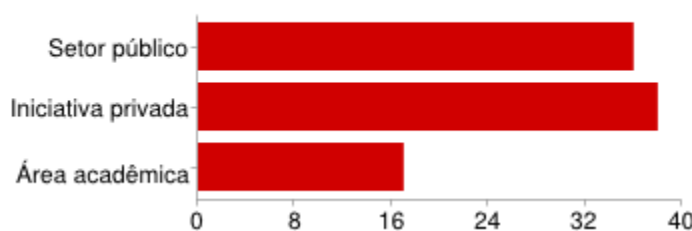
ScrumMaster	35	28%
Time	46	37%
Stakeholder	16	13%
Patrocinador	4	3%
Estudante	24	19%

Gráfico 33. Outros papéis desempenhados (O autor).

Por quanto tempo (total, em anos ou meses) exerceu?

04 anos 5 anos 7 meses seis meses 3 anos - Três meses 3
 2 1 0 4 1 ano e 10 meses 3 anos e meio 1 ano scrumMaster e 3 anos time
 2 anos 3 meses 03 1 mês 7 anaos hein?! 1 ano 4 anos 10 meses
 1 ano 3 anos 1 ano e meio 4 anos. 6 meses 1 ano e 2 meses 6 anos

Em qual(is) contexto(s) essa experiência com Scrum foi aplicada?



Setor público	36	40%
Iniciativa privada	38	42%
Área acadêmica	17	19%

Gráfico 34. Contexto da experiência com Scrum (O autor).

Atualmente, qual sua área de atuação no desenvolvimento de software?

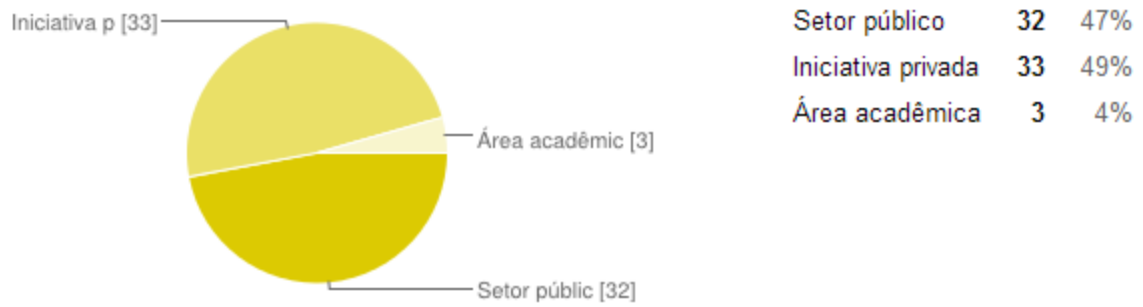


Gráfico 35. Área de atuação do respondente (O autor).

Nos dias de hoje, segue utilizando Scrum?

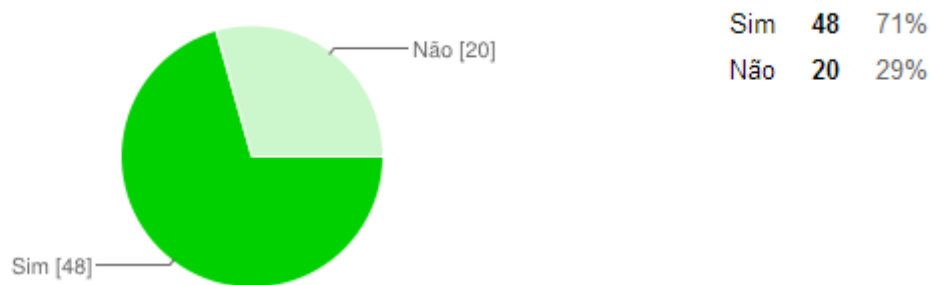


Gráfico 36. Continuidade na utilização do *Scrum* (O autor).

Qual sua localização (apenas UF)?

DF pi df PI GO PB Taguatinga Brasilia rj SP CE

APÊNDICE VII – RANKING DAS COMPETÊNCIAS TÉCNICAS

Quadro 9. Ranking das competências técnicas (O autor).

R	ID	Competência	Resultado	
			Absoluto	Percentual
1º	CT8	Possuir visão estratégica	67	99%
2º	CT2	Dominar o negócio em que o projeto está inserido	66	97%
3º	CT7	Dominar as técnicas de comunicação	66	97%
4º	CT3	Dominar técnicas de priorização do <i>Product Backlog</i>	66	97%
5º	CT9	Possuir visão sistêmica	65	96%
6º	CT15	Possuir o conhecimento dos princípios ágeis	64	94%
7º	CT16	Dominar técnicas de negociação	63	93%
8º	CT6	Dominar as técnicas de monitoramento do ROI (retorno sobre investimento) do produto	63	93%
9º	CT5	Dominar a técnica de validação de requisitos	62	92%
10º	CT14	Possuir o conhecimento dos valores ágeis	61	90%
11º	CT4	Dominar a técnica de escrita de <i>user story</i>	59	87%
12º	CT13	Possuir conhecimento dos ritos e cerimônias realizadas no <i>Scrum</i>	58	86%
13º	CT1	Dominar técnicas de planejamento de desenvolvimento ágil	57	84%
14º	CT11	Possuir conhecimento do impacto de teste funcional no projeto	55	81%
15º	CT12	Possuir conhecimento do impacto de refatoração no projeto	48	71%
16º	CT10	Possuir conhecimento técnico das tecnologias utilizadas no desenvolvimento do software	44	65%

APÊNDICE VIII – RANKING DAS COMPETÊNCIAS COMPORTAMENTAIS

Quadro 10. Ranking das competências comportamentais (O autor).

R	ID	Competência	Resultado	
			Absoluto	Percentual
1º	CC3	Possuir a capacidade de estabelecer uma relação de confiança com as pessoas que relaciona	68	100%
2º	CC1	Possuir a coragem para tomar uma decisão	68	100%
3º	CC2	Possuir a capacidade de admitir quando está errado	68	100%
4º	CC4	Possuir a disposição para ajudar a equipe	68	100%
5º	CC5	Possuir a habilidade de se adequar a diferentes contextos	65	96%
6º	CC6	Possuir a habilidade de resolver conflitos	64	94%