

GESTÃO DO CONHECIMENTO POTENCIALIZANDO A GESTÃO DE RISCOS DE PROJETOS DE DESENVOLVIMENTO DE SOFTWARE ÁGIL UTILIZANDO O SCRUM

Cassiano Raimar Silva¹

Fernando Zaidan²

José Luis Braga³

Resumo: Este estudo buscou propor um modelo em que a gestão do conhecimento potencializasse a gestão de riscos em projetos de desenvolvimento ágil de software utilizando o Scrum sem deixar de levar em consideração o caráter tácito do conhecimento nas metodologias ágeis. Esta pesquisa foi desenvolvida com a aplicação da metodologia Design Science (DS) e do método prescritivo Design Science Research (DSR), que propõe a geração de um artefato, neste caso, um modelo de gestão do conhecimento que potencializa a gestão de riscos no desenvolvimento de software com o Scrum. Esta pesquisa baseou-se na revisão de literatura e na análise quantitativa de um questionário aplicado a 121 pessoas. Os resultados obtidos com o questionário mostraram que todas as etapas da gestão do conhecimento podem impactar positivamente na gestão de riscos. Ao verificar os riscos referentes à perda de conhecimento ao final das sprints foi identificado que devido à utilização de conhecimentos adquiridos em experiências passadas ocorria um aumento do conhecimento, não sendo um risco negativo, mas positivo. Outro resultado obtido foi sobre não ser necessária uma documentação abrangente para que não ocorra perda de conhecimento no final das sprints. Validou-se o modelo criado por meio de um grupo focal junto a envolvidos em desenvolvimento de projetos ágeis de software e, após ajustes, foi obtido o modelo proposto.

Palavras-chave: Gestão de riscos; Gestão do conhecimento; Gestão ágil de projetos de software; Scrum.

KNOWLEDGE MANAGEMENT EMPOWERING THE RISK MANAGEMENT OF AGILE SOFTWARE DEVELOPMENT PROJECTS USING SCRUM

¹ Pós Graduado em Gestão da Segurança da Tecnologia da Informação pelo Centro Universitário de Belo Horizonte. Lattes: <http://lattes.cnpq.br/1126621486929841>

² Professor e pesquisador do mestrado do IETEC. Doutor em Ciência da Informação – UFMG. Mestre em Administração e Bacharel em Ciência da Computação – FUMEC. E-mail: contato@fernandozaidan.com.br Lattes: <http://lattes.cnpq.br/4835234239471713> ORCID <https://orcid.org/0000-0001-5715-5562>.

³ Professor e Pesquisador do Mestrado em Engenharia e Gestão de Sistemas e Processos do IETEC. Doutor em Informática, PUC-Rio Mestre em Ciência da Computação DCC-UFMG. Email: zeluisbraga@gmail.com Lattes: <http://lattes.cnpq.br/5199789743673693> ORCID: <https://orcid.org/0000-0002-1068-7008>

Abstract: This study aimed to propose a model in which knowledge management enhances risk management in agile software development projects using Scrum taking into consideration the tacit character of knowledge in agile methodologies. This research was developed using the Design Science methodology (DS) and the Design Science Research (DSR), a prescriptive method which proposes the generation of an artifact, in this case, a knowledge management model that enhances risk management in the development of software with Scrum. This research was based on literature review and quantitative analysis of a questionnaire applied to 121 people. The results obtained with the questionnaire showed that all stages of knowledge management can positively impact risk management. In verifying the risks related to the loss of knowledge at the end of the sprints, it was identified that due to the use of knowledge acquired in past experiences, there was an increase of knowledge, not a negative but positive risk. Another result was that would not be necessary a detailed documentation to avoid loss of knowledge at the end of sprints. The created model was validated through a focus group with people who work with agile software project development and, after adjustments, the proposed model was obtained.

Keywords: Risk management; Knowledge management; Agile software project management; Scrum.

1 INTRODUÇÃO

A complexidade no desenvolvimento de projetos de *software*, além de ocasionar mudanças constantes, aumenta a dificuldade do gerenciamento dos projetos (PRESSMAN; MAXIM, 2016). Há algumas décadas, a gestão desses projetos utilizava a abordagem prescritiva, ou tradicional, que, por ser inflexível, inviabilizava as mudanças necessárias em futuras fases do projeto (PRESSMAN, MAXIM, 2016).

Abordagens adaptativas, como as metodologias ágeis, mostraram-se mais adequadas e flexíveis a esse cenário de constantes mudanças e se adaptaram melhor às necessidades de projetos de *software* (VARGAS, 2016). Dados do *Standish Group* (2015) mostram que projetos de desenvolvimento de software que utilizam metodologias ágeis possuem uma taxa de sucesso três vezes superior quando comparados com outras metodologias tradicionais (STANDISH GROUP, 2015).

Dentre as metodologias ágeis o atual líder de utilização, o *Scrum*, é um framework de um processo ágil utilizado para gerenciar o desenvolvimento de um produto de software (CAMPOS; FONSECA, 2008). Ele emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos (SCHWALBER;

SUTHERLAND, 2013). O *Scrum* possui vários benefícios sem apresentar nenhuma desvantagem aparente (DULOCK; LONG, 2015).

Os processos do *Scrum* permitem a uma equipe trabalhar com sucesso mesmo em um ambiente cheio de incertezas, tal como o que projetos de *software* estão inseridos (PRESSMAN; MAXIM, 2016). Essas incertezas nos eventos de projetos de *software* abrem caminho para o gerenciamento dos riscos desses projetos, podendo permitir ganhos ou perdas (DEMARCO; LISTER, 2013).

Apesar de a gestão de riscos melhorar com a adoção das metodologias ágeis, gerir riscos não está entre os principais benefícios percebidos com a agilidade (MORAN, 2014). O *Scrum* não possui rituais ou etapas específicas para gestão de riscos (MORAN, 2014; TAVARES, 2015), contudo, este é um fator crítico para sucesso de projetos ágeis de desenvolvimento de software (GARLAND; FAIRBANKS, 2010).

Os métodos da gestão de riscos, em projetos de software, possuem como premissa, de acordo com Gusmão (2007), que seus resultados sejam armazenados e passados para os vários participantes do projeto, compartilhando o conhecimento e favorecendo a tomada de decisão. Entretanto, isso não ocorre efetivamente. Neves et al. (2014) concluíram que a transferência do conhecimento com a finalidade de gerenciar os riscos não foi evidenciada como integrante da cultura das empresas de software avaliadas. Nesse sentido, os autores sugerem que a identificação dos riscos ainda ocorre de forma mais reativa do que preventiva (NEVES et al., 2014).

Essa necessidade da gestão do conhecimento para a gestão de riscos ocasiona em que uma organização não consegue gerenciar efetivamente seus riscos se não gerenciar seu conhecimento (NEEF, 2005). Realçando a importância do conhecimento para a gestão de riscos, Tavares (2015) concluiu que a reutilização do conhecimento é uma importante prática para a gestão de riscos em projetos de *software*.

Visando potencializar a gestão de riscos em projetos ágeis de desenvolvimento de *software* utilizando o *Scrum*, uma alternativa considerada por esta pesquisa foi incorporar o processo de gestão do conhecimento à metodologia do *Scrum*. No intuito de propor um modelo em que a gestão de riscos fosse potencializada pela

gestão do conhecimento em projetos ágeis de software com o *Scrum*, esta pesquisa utilizou a metodologia *Design Science* (DS), operacionalizada por meio da *Design Science Research* (DSR).

Considerando a prática identificada por Tavares (2015) de reutilização do conhecimento para gestão de riscos em projetos ágeis de desenvolvimento de *software*, uma alternativa seria gerenciar esse conhecimento durante o processo de desenvolvimento. Nesse contexto surge, então, a questão: como a gestão do conhecimento pode potencializar a gestão de riscos de projetos ágeis de desenvolvimento de *software* com o *Scrum*?

2 REVISÃO DE LITERATURA

2.1 Gestão de projetos de desenvolvimento de software

Projetos de desenvolvimento de *software* são atividades complexas, as quais envolvem inovações tecnológicas e mudanças constantes nos requisitos do cliente (PRESSMAN; MAXIM, 2016). Essas alterações de escopo criam uma dificuldade em se concluir o projeto no tempo, no orçamento e com resultado satisfatório (OLIVEIRA; GOMES; LIMA, 2014).

Na busca por melhores maneiras de desenvolver projetos de *softwares*, foi redigido, em 2001, o manifesto para o desenvolvimento ágil de *software*, ou, simplesmente, manifesto ágil (BECK et al. 2001a). As metodologias que compõe o movimento ágil compartilham os princípios definidos no manifesto ágil (BECK et al. 2001a), e diferem ao propor processos diferentes para executá-los (SOMMERVILLE, 2007).

Metodologias ágeis são definidas como sendo processos aplicados ao desenvolvimento de *software* de maneira a permitir que eles sejam iterativos, incrementais, auto-organizados e emergentes (LINDVALL et al., 2002). Dentre as principais metodologias ágeis utilizadas no desenvolvimento de *software* citadas por Sakai e Ricarte (2015) está o *Scrum*.

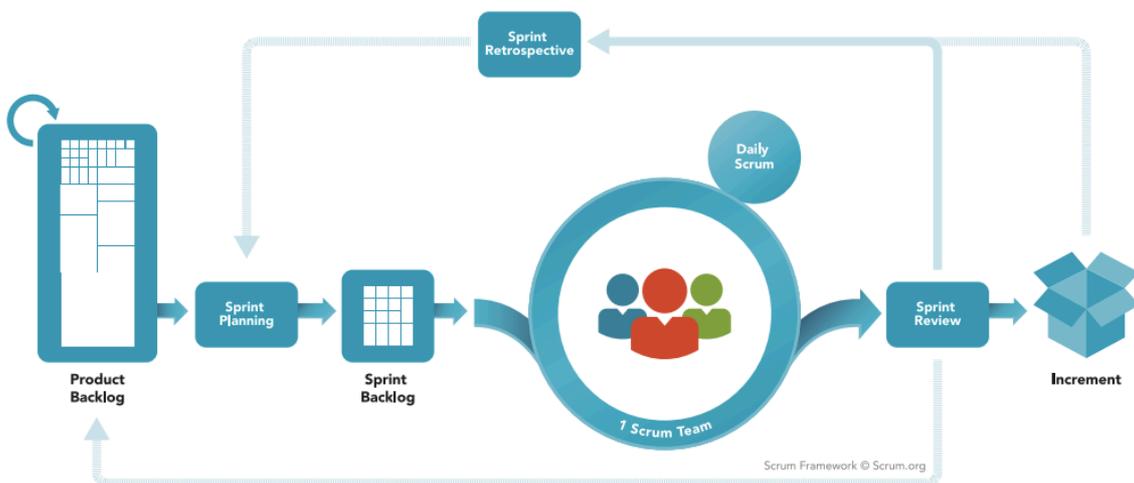
Esta pesquisa optou por focar na metodologia *Scrum*, principalmente pelo alto índice de sua utilização em empresas de software, como comprovou o relatório anual *State of Agility*, o qual apontou que 58% dos pesquisados responderam que a

utilizavam. Ao considerar o *Scrum* e suas variações, este percentual de utilização sobe para 76% (VERSIONONE, 2017).

2.2 Scrum

O *Scrum* é um framework de um processo ágil utilizado para gerenciar o desenvolvimento de um produto de *software* (CAMPOS; FONSECA, 2008). Ele é composto por papéis e suas respectivas responsabilidades, cerimônias de tempo fixo e artefatos que são produzidos durante o projeto (SCHWABER; SUTHERLAND, 2013).

Figura 1 – Framework Scrum



Fonte: SCRUM, 2019.

O *Scrum* emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos (SCHWABER; SUTHERLAND, 2013) e seus princípios são compatíveis com o manifesto ágil e objetivam orientar as atividades dentro de um processo de desenvolvimento de *software* (PRESSMAN; MAXIM, 2016).

2.3 Riscos

Os riscos tratados nesta pesquisa estão inseridos no contexto da gestão de projetos de desenvolvimento de software. Segundo Pressman e Maxim (2016), os riscos em projetos de *software* podem ser classificados em riscos de projeto, riscos técnicos e riscos de negócio.

Segundo os autores: “riscos de projeto são aqueles que afetam o orçamento, cronograma, recursos, e os impactos do cliente e seus requisitos no projeto” (PRESSMAN; MAXIM, 2016, p. 779).

Riscos técnicos são aqueles relacionados à implementação do projeto (PRESSMAN; MAXIM, 2016).

Riscos de negócio são aqueles que impactam a viabilidade do *software* a ser desenvolvido.

Chapman (2016) conceitua gestão de riscos em projetos como sendo um processo de gerenciamento proativo que visa explorar oportunidades e tratar ameaças para garantir objetivos acordados, definidos e divulgados em um projeto. Para que uma organização consiga gerenciar efetivamente seus riscos, ela deve gerenciar o seu conhecimento (NEEF, 2005).

2.4 Conhecimento

O conhecimento pode ser definido como a compreensão adquirida pela combinação interpretativa de dados e informação (GAGANDEEP, 2012). O conhecimento possui natureza intangível e está diretamente relacionado à mente humana (GONZALEZ; MARTINS, 2017). A retenção do conhecimento gera um aumento da vantagem competitiva para as empresas, gerando ganhos também para os funcionários (ZAIDAN, 2008).

O conhecimento pode ser classificado, segundo Polanyi (1967), em explícito e tácito. O conhecimento tácito é pessoal, relativo ao contexto e experiências do indivíduo. É difícil de ser formulado e repassado. O conhecimento explícito, ou codificado, refere-se ao conhecimento transmissível em linguagem formal e sistemática (NONAKA; TAKEUCHI, 1997).

Devido ao caráter multidisciplinar da gestão de conhecimento, a revisão de literatura feita por Gonzalez e Martins (2017) indica que há uma diversidade de modelos e temas ligados à gestão do conhecimento, porém, eles identificaram as duas principais vertentes desses temas. A primeira vertente é aquela que considera a gestão do conhecimento como tendo seu escopo restrito à tecnologia da

informação. A segunda vertente envolve a gestão do conhecimento como forma de desenvolvimento organizacional, ressaltando a estrutura e a cultura organizacional como facilitadores na interação entre os indivíduos, proporcionando o compartilhamento do conhecimento (GONZALEZ; MARTINS, 2017).

Esta pesquisa alinha-se à segunda vertente, considerando a gestão do conhecimento conforme o proposto por Gonzales e Martins (2017), combinando os processos organizacionais à tecnologia da informação e sendo uma atividade que desenvolve, armazena e transfere conhecimento visando à tomada de decisões corretas. A gestão do conhecimento também é tratada nesta pesquisa como sendo um processo com fases específicas (PINHO; REGO; CUNHA, 2012) com o objetivo de disseminar e utilizar o conhecimento entre os indivíduos (CORMICAN; O' SULLIVAN, 2003).

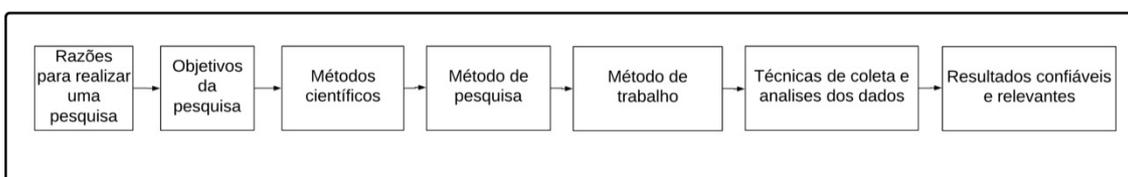
Apesar de as etapas desse processo variarem bastante, de acordo com o autor, o processo de gestão do conhecimento é constituído essencialmente de quatro etapas: aquisição, armazenamento, distribuição e utilização do conhecimento (CORMICAN; O' SULLIVAN, 2003; MAGNIER-WATANABLE; SENOO, 2008).

3 METODOLOGIA

O paradigma dominante nesta pesquisa foi a *Design Science* (DS), o qual é utilizado para desenvolver e projetar soluções para melhorar sistemas existentes, resolver problemas ou criar artefatos (DRESCH; LACERDA; JÚNIOR, 2015). A DS foi escolhida, em primeiro lugar, por auxiliar na condução de pesquisas da área de gestão, conferindo não apenas rigor, mas também relevância à pesquisa (DRESCH; LACERDA; JÚNIOR, 2015).

Esta pesquisa segue a estratégia proposta por Dresch, Lacerda e Júnior (2015) para condução de pesquisas científicas fundamentadas na DS, conforme demonstrado na Figura 2.

Figura 2 – Passos para execução de pesquisas científicas que utilizam DS



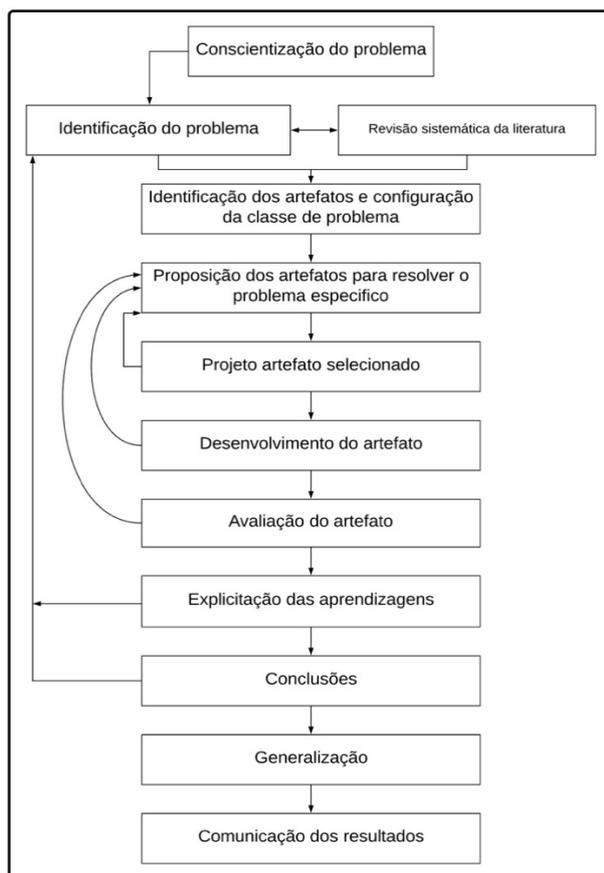
Fonte: DRESCH; LACERDA; JÚNIOR, 2015, p. 62. (Adaptada pelo autor)

O método científico abduutivo foi utilizado para criar o questionário e propor o modelo, sendo todas as questões criadas com base na literatura. O método científico dedutivo foi utilizado ao validar o modelo criado. Para operacionalizar a DS foi utilizada a Design Science Research (DSR). Segundo Dresch, Lacerda e Júnior (2015), a DSR é o método prescritivo que fundamenta a condução da pesquisa quando o objetivo a ser alcançado é, dentre outros, um modelo. A DSR é um método que pode ser utilizado para diminuir o distanciamento entre teoria e prática (DRESCH; LACERDA; JÚNIOR, 2015).

3.1 Método de trabalho

O método de trabalho é a sequência de passos que o pesquisador utilizou para realizar sua pesquisa (DRESCH; LACERDA; JÚNIOR, 2015).

Figura 3 – Etapas da DSR



Fonte: DRESCH; LACERDA; JÚNIOR, 2015, p. 67. (Adaptada pelo autor)

Esses passos devem ser estruturados e seguidos de maneira que seja possível a replicabilidade da pesquisa (MENTZER; FLINT, 1997). O método de trabalho

utilizado nesta pesquisa é um detalhamento das etapas da DSR, que estão apresentadas na Figura 3.

Após a identificação do problema apresentado na introdução, a revisão sistemática da literatura foi desenvolvida abordando os principais conceitos envolvidos na gestão de projetos ágeis de *software*, gestão de riscos e gestão do conhecimento.

Na etapa de identificação dos artefatos e configuração da classe de problemas, foi gerado um questionário baseado na revisão de literatura.

Após o resultado do questionário aplicado e com base também no referencial teórico foi desenvolvido um modelo que propõe a gestão do conhecimento para atuar na gestão de riscos em projetos ágeis de desenvolvimento de software. Esse modelo foi validado utilizando a técnica de grupo focal.

3.2 Técnicas de coleta e análise de dados

Para a elaboração do questionário proposto na etapa de identificação de artefatos foi utilizado a sequência lógica proposta por Aaker et al. (2001).

Inicialmente, é necessário obter informações a partir de fontes secundárias sobre os conceitos relativos ao problema da pesquisa. O referencial teórico serviu de entrada para essa etapa. Em seguida, o questionário foi dividido em duas partes. Na primeira foram formuladas as questões que visam classificar e selecionar o universo pesquisado; na segunda, as questões foram formuladas, principalmente com base no referencial teórico.

Em seguida foram definidos o texto dessas questões e o aspecto visual do questionário, que foi implementado utilizando a ferramenta Typeform. Os textos das questões da primeira etapa foram de múltipla escolha, tendo como última opção a alternativa "Outros". O texto das questões da segunda etapa foi elaborado com questões fechadas utilizando a escala Likert com 5 níveis, conforme a Figura 4. Essa escala permite que um valor numérico seja dado a uma opinião (COLLIS; HUSSEY, 2005). Também foi adicionada uma questão aberta ao final para que o respondente pudesse acrescentar algo referente à pesquisa.

Figura 4 – Escala *Likert* cinco níveis

O diagrama mostra uma escala Likert de cinco níveis, representada por cinco retângulos azuis empilhados verticalmente. Cada retângulo contém uma letra em um quadrado amarelo e um texto em português:

- A Concordo totalmente
- B Concordo parcialmente
- C Não concordo e nem discordo
- D Discordo parcialmente
- E Discordo totalmente

Fonte: AUTOR, 2018. (Ferramenta *Typeform*)

Para evitar erros e ambiguidade, Aaker et al. (2001) sugerem que o questionário seja testado com uma pequena amostra. Nessa etapa foi gerada uma versão piloto para teste com quatro pessoas. Esse teste identificou erros de digitação, problemas de falta de exemplos na questão que abordava as ferramentas sugeridas e dúvidas quanto à escala de tempo de trabalho com o *Scrum*.

O tamanho da amostra foi definido pela fórmula cuja população não é conhecida, nem as proporções populacionais desses indivíduos. A fórmula para cálculo amostral a ser utilizada, segundo Triola (1999), está apresentada a seguir.

$$n = \frac{Z_{\alpha/2}^2 \cdot 0,25}{E^2}$$

Para esta pesquisa foi adotada, por conveniência do autor, uma margem de erro de 9%, com um nível de confiança de 95%. Aplicando esse valor na fórmula mencionada, o tamanho mínimo da amostra deve ser de 119 respondentes.

Para análise dos dados foi utilizado o software Minitab 18⁴. Nele foram gerados os gráficos das questões e os dados da estatística descritiva. Também foram analisados os resultados de estatística multivariada, conforme sugerido pela DSR.

Visando aumentar a confiabilidade desse artefato, a DSR sugere a validação do modelo utilizando a técnica de grupo focal (DRESCH; LACERDA; JÚNIOR, 2015), a qual tem como objetivo coletar informações por meio das interações grupais (MORGAN, 1997). Essa técnica é preferencialmente adotada em pesquisas

⁴ Disponível em: <http://www.minitab.com/pt-br/>

avaliativas, podendo ser a principal fonte de dados ou funcionar como uma técnica complementar em pesquisas quantitativas (BOMFIM; LENY, 2009).

O grupo focal seguiu as boas práticas apontadas por Bomfim e Leny (2009) quanto a local, equipamentos necessários, e número de participantes. Ele foi realizado no dia 13 de março de 2018 nas dependências da empresa Minutrade, em Belo Horizonte, Minas Gerais. A atividade foi filmada e teve duração de 90 minutos, contando com a presença de nove pessoas, sendo seis desenvolvedores, dois *scrum* master e um *product owner*, contemplando, assim, todos os papéis descritos pelo Scrum.

A composição do grupo considerou o perfil dos participantes, de modo a ter todos os perfis desejados, formando, então, uma amostra intencional (DALL'AGNOL; TRENCH, 1999).

Na primeira parte do debate, os resultados obtidos com o questionário serviram de roteiro para as discussões entre os participantes. Após algum tempo discutindo um gráfico apresentado era feita uma intervenção ligando o gráfico discutido com o próximo a ser discutido. Os gráficos apresentados foram divididos entre os que obtiveram maiores médias, isto é, que a questão proposta obteve maior concordância, e entre os gráficos que representavam as questões que obtiveram as maiores discordâncias. Na segunda parte do grupo focal, após apresentação do modelo e de seus componentes, foi discutido, então, o modelo proposto.

O grupo focal foi encerrado após um resumo do tema apresentado e um agradecimento aos participantes. As análises das discussões serão apresentadas no capítulo seguinte.

4 RESULTADO

4.1 Questionário – Impactos da gestão do conhecimento

O questionário com 21 questões foi respondido por 127 pessoas, sendo que seis foram excluídas por não apresentarem experiência com o *Scrum*, totalizando, então, uma amostra de 121 pessoas. Conforme mencionado no capítulo de metodologia, foi adotada, por conveniência do autor, uma margem de erro de 9% com um nível de confiança de 95%, a qual exigia uma amostra superior a 119 respondentes.

As primeiras questões buscavam classificar os respondentes quanto às características individuais e quanto à forma que utilizavam o *Scrum*. A classificação dos respondentes é importante não apenas para funcionar como um filtro sobre a capacidade dos participantes no tema do questionário, mas também para permitir verificar se a amostra é heterogênea quanto ao tempo de experiência e quanto às funções executadas. No Quadro 1, a seguir, são exibidos os resultados obtidos na classificação dos participantes.

Quadro 1 – Quadro com a classificação dos respondentes quanto a características individuais

QUESTÃO	GRÁFICO
Qual seu grau de escolaridade?	<ul style="list-style-type: none"> ■ Ensino superior ■ Especialização ■ Mestrado ■ Ensino médio
Qual sua função na sua empresa?	<ul style="list-style-type: none"> ■ Desenvolvedor ■ Líder técnico / Scrum Master ■ Product Owner ■ Gerente de Projetos ■ Outros
Você possui quanto tempo de experiência em projetos de desenvolvimento de <i>software</i> ?	
Você possui quanto tempo de experiência com <i>Scrum</i> ?	

Fonte: AUTOR, 2019.

A partir do Quadro 1 pode-se perceber que a maioria dos respondentes possui pelo menos graduação e mais de três anos de experiência com projetos de desenvolvimento de software e com *Scrum*. A amostra apresenta todos os papéis do *Scrum* e diferentes níveis de experiência.

Após a classificação dos respondentes foram apresentadas aos respondentes 12 questões sobre a interação entre conhecimento, gestão de riscos e o *Scrum*. Para medir o quanto os respondentes concordavam com o questionamento apresentado foi utilizada a escala *Likert* com 5 níveis (COLLIS; HUSSEY, 2005). Esses dados também foram utilizados para criar o modelo do tópico 2 deste capítulo.

Nesta etapa foram utilizadas 7 questões para avaliar se a gestão do conhecimento impactaria positivamente a gestão de riscos no desenvolvimento ágil de *software* com o *Scrum*. Considerando a gestão do conhecimento como sendo as etapas de aquisição, armazenamento, distribuição e utilização do conhecimento, como definido no capítulo referencial teórico, o impacto da gestão do conhecimento sobre a gestão de riscos foi analisado relacionando cada uma destas etapas com os resultados obtidos no questionário, conforme apresentado no Quadro 2, a seguir.

Quadro 2 – Relação das etapas com os resultados obtidos no questionário

Gráfico	Objetivo
	Etapa aquisição – Verificar se o incentivo ao aprimoramento do conhecimento pessoal seria uma boa prática no processo de desenvolvimento de software com o Scrum para realizar a gestão dos riscos.
	Etapa armazenamento – Avaliar se um processo de gestão de riscos que exija a criação de documentação garante que ele seja seguido pela equipe.
	Etapa armazenamento – Validar se os riscos identificados no início do projeto devem ser registrados para consulta durante o projeto.
	Etapa distribuição – Identificar se o ambiente organizacional que facilita o compartilhamento do conhecimento pode mitigar riscos no cenário desta pesquisa.

Gráfico	Objetivo												
<table border="1"> <caption>Dados do Gráfico 1: Etapa distribuição</caption> <thead> <tr> <th>Item</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>10</td> </tr> <tr> <td>2</td> <td>25</td> </tr> <tr> <td>3</td> <td>15</td> </tr> <tr> <td>4</td> <td>50</td> </tr> <tr> <td>5</td> <td>20</td> </tr> </tbody> </table>	Item	Valor	1	10	2	25	3	15	4	50	5	20	<p>Etapa distribuição – Esta questão tem como objetivo identificar se a comunicação em uma equipe ágil que privilegie a inteligência coletiva é suficiente para gerenciar os riscos em projetos de software com Scrum.</p>
Item	Valor												
1	10												
2	25												
3	15												
4	50												
5	20												
<table border="1"> <caption>Dados do Gráfico 2: Etapa utilização - Validar</caption> <thead> <tr> <th>Item</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>2</td> <td>5</td> </tr> <tr> <td>3</td> <td>8</td> </tr> <tr> <td>4</td> <td>50</td> </tr> <tr> <td>5</td> <td>65</td> </tr> </tbody> </table>	Item	Valor	1	2	2	5	3	8	4	50	5	65	<p>Etapa utilização – Validar se durante o desenvolvimento ágil com o Scrum a equipe executa atividades que proporcionam a melhoria e o aprendizado contínuo no projeto através do feedback constante.</p>
Item	Valor												
1	2												
2	5												
3	8												
4	50												
5	65												
<table border="1"> <caption>Dados do Gráfico 3: Etapa utilização - Avaliar</caption> <thead> <tr> <th>Item</th> <th>Valor</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>2</td> <td>5</td> </tr> <tr> <td>3</td> <td>8</td> </tr> <tr> <td>4</td> <td>30</td> </tr> <tr> <td>5</td> <td>85</td> </tr> </tbody> </table>	Item	Valor	1	2	2	5	3	8	4	30	5	85	<p>Etapa utilização – Esta questão tem como objetivo avaliar se práticas que incentivam a reutilização do conhecimento sobre riscos podem permitir mitigar riscos no processo de desenvolvimento de software com Scrum.</p>
Item	Valor												
1	2												
2	5												
3	8												
4	30												
5	85												

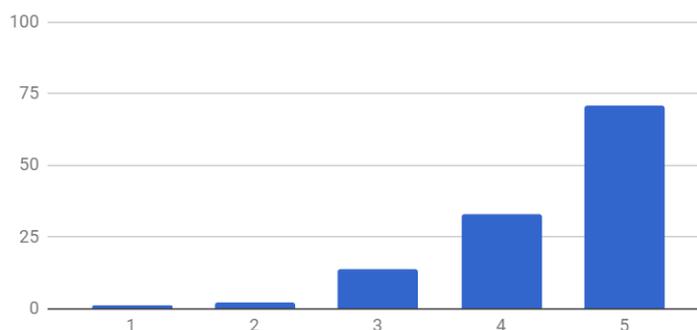
Fonte: AUTOR, 2019.

O Quadro 2 mostra que todas as fases da gestão do conhecimento podem impactar positivamente a gestão de riscos. Um ambiente organizacional que incentiva o aprimoramento pessoal e o compartilhamento do conhecimento é um fator mitigador de riscos. Além disso, o conhecimento que o *Scrum* proporciona pelo aprendizado contínuo e *feedback* constante aliado à necessidade de registro de riscos desde o início do projeto para consulta posterior permite a prática de reutilizar o conhecimento aprendido sobre riscos como forma de mitigá-los.

Os fatores que não apresentaram um alto nível de concordância com o questionamento proposto refletem o equilíbrio que deve haver na gestão de riscos em projetos de desenvolvimento ágil de *software* com o *Scrum*. Utilizar processos formais para a gestão de riscos que levam à criação de uma documentação abrangente de riscos ou gerir os riscos apenas privilegiando a inteligência coletiva não se mostraram adequados ou suficientes para gerir riscos em projetos de *software* com o *Scrum*.

Além dos fatores apresentados, a Questão 11 tinha o objetivo de avaliar se o compartilhamento do conhecimento e as boas práticas de gestão do conhecimento poderiam ser utilizadas para mitigar riscos no desenvolvimento ágil de *software* utilizando o *Scrum*.

Gráfico 1 – Utilização da gestão do conhecimento para mitigar riscos no cenário apresentado nessa pesquisa (Questão 11)



Fonte: AUTOR, 2019.

Mais de 70 pessoas concordam totalmente que a gestão do conhecimento pode ser utilizada para mitigar riscos no desenvolvimento de *software* com *Scrum*. Esse resultado mostra a gestão do conhecimento como uma alternativa viável para gestão de riscos em projetos dessa natureza (HALTIWANGER, 2012; TAVARES, 2015).

4.2 MODELO DE GESTÃO DO CONHECIMENTO POTENCIALIZANDO A GESTÃO DE RISCOS NO SCRUM E SUA VALIDAÇÃO

Para criação do modelo, conforme sugerido no paradigma dominante nesta pesquisa, deveriam ser identificados, segundo Dresch, Lacerda e Júnior (2015), os artefatos pertencentes à classe do problema desta pesquisa. Dentre os artefatos que Dresch, Lacerda e Júnior (2015) citam, apresentam-se os constructos, modelos e métodos.

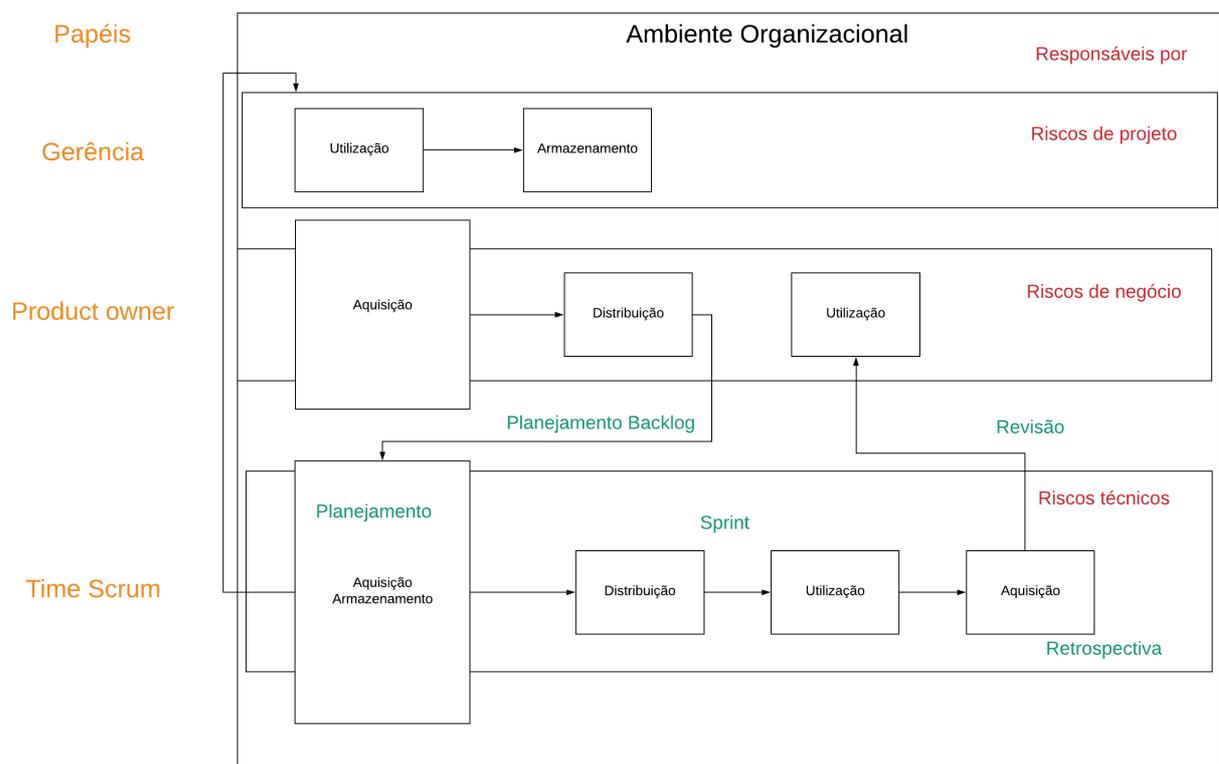
Os artefatos necessários para criação do modelo são principalmente o modelo de desenvolvimento de *software* utilizando *Scrum*, proposto por Schwaber e Sutherland (2013); as fases da gestão do conhecimento, definidas por Cormican e O'Sullivan (2003) e Magnier-Watanable e Senoo (2008); e as classificações de risco em desenvolvimento de *software*, definidas por Pressman e Maxim (2016). Todos esses artefatos foram apresentados durante a revisão bibliográfica. Além destes

artefatos, o resultado do questionário também serviu como base para criação do modelo proposto.

O modelo inicialmente proposto nesta pesquisa está contido em um ambiente organizacional que facilita o compartilhamento do conhecimento e segmentado em três camadas, sendo: camada do ciclo de desenvolvimento, camada da gestão de produtos e camada da gestão de projetos. Cada camada possui um ou mais responsáveis por um dos tipos de riscos de desenvolvimento de *software* definidos por Pressman e Maxim (2016) e as fases da gestão do conhecimento pertencentes a essa camada.

Figura 5 – Modelo inicialmente proposto

Modelo de gestão do conhecimento na gestão de riscos em projetos ágeis



Fonte: AUTOR, 2019.

Conforme resultado obtido na Questão 10, um ambiente organizacional que facilita o compartilhamento do conhecimento impacta na gestão de riscos ao mitigá-los no desenvolvimento ágil de *software*. Nesse sentido, Jamil (2005) aponta a relevância do ambiente organizacional para a gestão do conhecimento. O questionário também mostrou que o ambiente organizacional deve incentivar o aprimoramento pessoal e o surgimento de comunidades de práticas.

A maior taxa de sucesso das metodologias ágeis e, dentre elas, a maior utilização de *Scrum* para o desenvolvimento de *software* (VERSIONONE, 2017), motivou esta pesquisa a adotar o *Scrum* como base para o ciclo de desenvolvimento proposto neste modelo. Nessa camada foram adicionados aos ritos do *Scrum* as fases da gestão do conhecimento que potencializariam a gestão de riscos durante o desenvolvimento de *software*.

No Quadro 3 são apresentadas as fases da gestão do conhecimento associadas aos ritos do *Scrum*, além da discussão de como essas fases podem potencializar a gestão de riscos.

Quadro 3 – Fases da gestão do conhecimento associadas aos ritos do *Scrum*

RITO NO SCRUM	FASE DA GC	DISCUSSÃO
Planejamento <i>sprint</i>	Aquisição, armazenamento	Durante o planejamento da <i>sprint</i> , esse modelo propõe que sejam criados conhecimentos tácitos ao se discutir sobre os possíveis riscos identificados nas tarefas selecionadas para execução naquela <i>sprint</i> utilizando <i>brainstorming</i> e explícitos ao armazenar esses riscos em uma base de conhecimento.
Execução da <i>sprint</i>	Distribuição, utilização	Ao executar a <i>sprint</i> , esse modelo sugere que ocorra a distribuição do conhecimento por contato social (GONZALEZ, 2017) e que o conhecimento gerado em outras etapas possa ser utilizado nas tomadas de decisões.
Reunião diária	Distribuição	Na reunião diária, além de cumprir os objetivos do <i>Scrum</i> de compartilhar o que foi feito no dia anterior e o que será feito no dia da reunião ou se há algum impedimento na tarefa (SCHWABER; SUTHERLAND, 2013), poderia também compartilhar algum risco encontrado.
Reunião de revisão	Distribuição	Ao inspecionar o que foi entregue como incremento pelo time, este poderia compartilhar o conhecimento com o <i>product owner</i> sobre os riscos identificados.
Reunião de retrospectiva	Aquisição	Durante a reunião de retrospectiva, esse modelo propõe que ocorra a absorção do conhecimento gerado, incorporando as lições aprendidas e mitigando os riscos no futuro (GAGANDEEP, 2012).

Fonte: AUTOR, 2019.

Os atores responsáveis pela gestão de riscos nessa camada são o time de desenvolvimento *Scrum*. Na abordagem desse modelo, *Scrum master* e time de desenvolvimento são os responsáveis por riscos técnicos. Esses atores devem se preocupar com a implementação do projeto, removendo a ambiguidade dos requisitos implementados e lidando com as incertezas causadas pelas ferramentas utilizadas no desenvolvimento de *software* (PRESSMAN; MAXIM, 2016).

A camada de gestão de produtos, nesse modelo, é a responsável por gerenciar o conhecimento sobre o *software* a ser criado, seja como ele deve parecer ou quais funcionalidades ele deve possuir. Como é papel do *product owner* definir os itens do *backlog* de produto e representar os *stakeholders* durante o rito do *Scrum* de revisão do produto (SCHWABER; SUTHERLAND, 2013), as fases da gestão do conhecimento que potencializam a gestão de riscos durante o desenvolvimento de *software* para essa camada visam gerenciar o conhecimento para a criação do *backlog* de produto e a utilização desse conhecimento na revisão de *software* executada juntamente com o time de desenvolvimento. Esse conhecimento utilizado durante a revisão do produto serve como *feedback* para adaptação do *backlog* de produto.

Quadro 4 – Fases da GC associados a camada de gestão de produtos

ATIVIDADE	FASE DA GC	DISCUSSÃO
Criação de <i>product backlog</i>	Aquisição, distribuição	Devido a ser o representante de outros envolvidos no projeto, o <i>product owner</i> deve adquirir conhecimentos pelo processo criativo de indivíduos ou grupos na organização, fornecendo insumos para a criação do <i>product backlog</i> . Além disso, o <i>product owner</i> tem, após essa etapa, que compartilhar o conhecimento com o time de desenvolvimento sobre os riscos de negócios identificados.
Reunião de revisão	Utilização	Na reunião de revisão, esse modelo propõe que o <i>product owner</i> possa localizar, acessar e então utilizar o conhecimento (ZACK, 1999), visando maximizar o valor do produto realizado pelo time de desenvolvimento (SCHWABER; SUTHERLAND, 2013) e possibilitando a geração de novos conhecimentos (MAGNIER-WATANABLE; SENOO, 2008), como o tempo gasto pelo time naquela entrega de valor, para serem aproveitados no futuro.

Fonte: AUTOR, 2019.

O ator responsável pela gestão de riscos, nessa camada, é o *product owner*. Esse modelo propõe que o *product owner* seja o responsável pelos riscos de negócio, os quais impactam na viabilidade de o *software* ser desenvolvido, seja por falta de mercado, inadequação na estratégia da empresa ou baixo orçamento (PRESSMAN; MAXIM, 2016).

A camada de gestão de projetos foi incluída nesse modelo devido a ser necessário um ator para ser responsável pelos riscos de projeto. Pressman e Maxim (2016) definem os riscos em projetos de *software* como sendo riscos técnicos, riscos de negócio e riscos de projeto. Porém, nenhum dos papéis definidos no *Scrum* possuem como responsabilidade os riscos de projeto. Riscos de projeto são aqueles que

afetam o orçamento, cronograma, recursos e os impactos do cliente e seus requisitos no projeto (PRESSMAN; MAXIM, 2016).

Quadro 5 – Fases da GC associados a camada de gestão de projetos

ATIVIDADE	FASE DA GC	DISCUSSÃO
Tomada de decisões estratégicas quanto ao conhecimento	Utilização	Nesta etapa o modelo propõe que o conhecimento gerado pelo time de desenvolvimento possa ser utilizado nas tomadas de decisões sobre a necessidade de capacitação da equipe. Essa atividade se mostrou necessária, conforme demonstrado na Questão 7 do questionário.
Retenção do conhecimento no projeto	Armazenamento	Como responsável pela retenção do conhecimento do time de desenvolvimento neste modelo, o gerente dessa camada pode utilizar 3 abordagens, segundo Gonzalez (2017). A primeira seria utilizar os próprios indivíduos como instrumentos de retenção do conhecimento tácito. A segunda maneira seria institucionalizar o conhecimento utilizando a estrutura e cultura organizacional como facilitadores do compartilhamento do conhecimento. Essa prática foi mostrada na Questão 10 do questionário aplicado como sendo um fator que mitiga riscos no desenvolvimento de <i>software</i> com <i>Scrum</i> . Esse gestor também teria o papel de fomentar o compartilhamento de conhecimento por meio da formação de comunidades de práticas. Essa prática de gestão do conhecimento permite mitigar riscos (GAGANDEEP, 2012).

Fonte: AUTOR, 2019.

Nessa camada, o ator responsável pela gestão de riscos é o gerente de projetos. Nesse modelo, os gerentes de projetos são responsáveis por tomadas de decisões estratégicas quanto ao conhecimento, visando ter um ambiente organizacional que facilite o compartilhamento do conhecimento e a retenção do conhecimento no projeto.

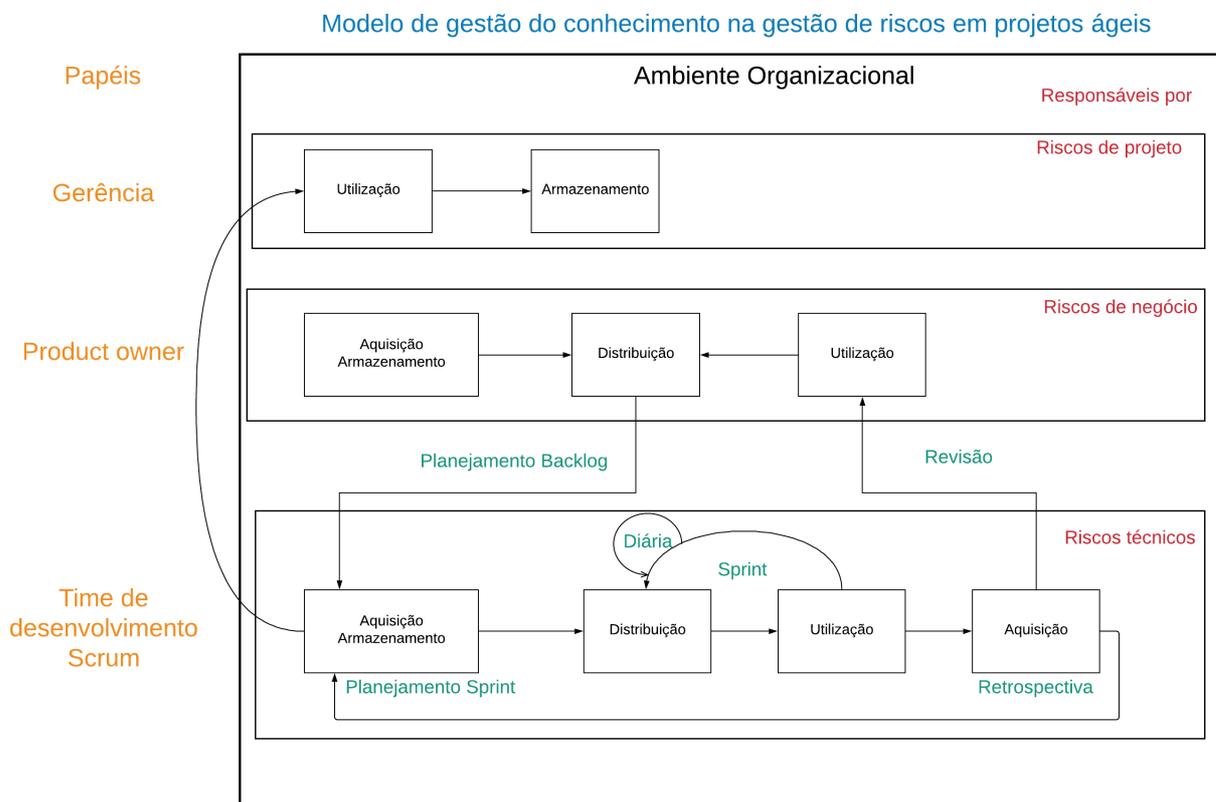
Após a criação do modelo, ele foi validado junto a um grupo focal, conforme detalhado na seção da metodologia. Nesse grupo focal estiveram presentes representantes de todos os papéis propostos na metodologia *Scrum*. As modificações propostas pelo grupo focal levaram ao modelo proposto por esta pesquisa. Nessa nova proposta se destacam:

Quadro 6 – Discussão sobre as modificações propostas pelo grupo focal para o modelo

MODIFICAÇÃO	DISCUSSÃO
Demonstração mais clara de onde está o <i>Scrum</i> no modelo.	O novo modelo proposto apresenta mais claramente o modelo do <i>Scrum</i> , passando a apresentar o formato proposto por Schwaber e Sutherland (2013), além de todos os seus ritos.
Conhecimento adquirido na retrospectiva servindo como base para o planejamento da <i>sprint</i> . Importância de tecnologia utilizada mais próxima das ferramentas utilizadas no cotidiano dos usuários. Conhecimentos utilizados durante a reunião de revisão pelo time de produto repassada com o time de desenvolvimento na próxima reunião de planejamento da <i>sprint</i> .	A Questão 8 do questionário aplicado mostrou que a reutilização do conhecimento causa a retenção do conhecimento. O grupo focal sugeriu que o conhecimento adquirido na retrospectiva fosse utilizado para identificação mais assertiva dos riscos na próxima <i>sprint</i> . Essa técnica de mitigação de riscos também foi citada por Tavares (2015) e Gagandeep (2012).
Cada uma das camadas terá a sua base de conhecimento.	Considerando a importância de a tecnologia utilizada ser mais próxima do cotidiano dos usuários, cada camada terá a sua ferramenta que implementa a base de conhecimento, permitindo a cada camada uma forma de localizar, acessar e, então, utilizar o conhecimento (ZACK, 1999).
Bases de conhecimento da camada de produto e do ciclo de desenvolvimento são acessíveis a todos.	O grupo focal sugeriu que, apesar de cada camada poder ter a sua forma de armazenar conteúdo, o conteúdo criado por uma camada também serviria para outra.

Fonte: AUTOR, 2019.

Figura 6 – Modelo proposto após grupo focal



Fonte: AUTOR, 2019.

Dessa forma, foi finalizada a discussão com a apresentação do modelo proposto.

5 CONCLUSÕES

O presente trabalho buscou uma alternativa que permitisse potencializar a gestão de riscos em desenvolvimento ágil de *software* com *Scrum*. O modelo proposto permite gerenciar o conhecimento gerado em prol da gestão de riscos ao definir os atores responsáveis pelos tipos de riscos em desenvolvimento de software. Com as fases da gestão do conhecimento em cada camada, o modelo permite que sejam utilizadas as ferramentas de gestão do conhecimento de maneira a incentivar a reutilização do conhecimento sobre riscos.

A utilização da DSR permitiu aproximar este trabalho teórico da prática por meio da aplicação de um questionário e validação utilizando grupo focal. Os resultados obtidos com o questionário permitiram identificar que todas as fases da gestão do conhecimento impactam a gestão de riscos ao fornecer mais informações para a tomada de decisão durante a análise de riscos e mitigar riscos.

Tanto no grupo focal quanto na literatura foram encontradas evidências de que há a necessidade de o ambiente organizacional facilitar os entraves e a resistência inicial no compartilhamento do conhecimento, sendo necessário incentivos a criação não impositiva de base de conhecimentos. Apesar do caráter tácito das metodologias ágeis e da retenção de conhecimento com o Scrum devido à reutilização do conhecimento, os resultados apontaram para a necessidade de se realizar o registro dos riscos identificados no início do projeto para consulta durante a sua execução.

As principais contribuições desta pesquisa foram ao propor uma integração entre as fases da gestão do conhecimento e a metodologia de desenvolvimento de *software Scrum* visando potencializar a gestão de riscos. Ao identificar as fases da gestão do conhecimento e quais ritos do *Scrum* podem ocorrer permitiu criar um modelo que possibilita utilizar as ferramentas da gestão do conhecimento na gestão de riscos.

Para trabalhos futuros são propostos estudos que permitam aprofundar mais na interação entre equipes diferentes e multidisciplinares. Sugerem-se, ainda, os

questionamentos: “formas diferentes de se agrupar as pessoas no desenvolvimento de software permitem mais o compartilhamento de conhecimento?”

Concluiu-se que a gestão do conhecimento pode potencializar a gestão de riscos durante o desenvolvimento de software ágil com o Scrum ao mostrar a relevância do ambiente organizacional que facilite o compartilhamento de conhecimento, das práticas que incentivam a reutilização do conhecimento sobre riscos e do registro de riscos que estejam disponíveis a todos durante a execução do projeto como fatores que mitigam riscos.

6 REFERÊNCIAS

AAKER, D. A. *et al.* **Marketing research**. 7. ed. New York: John Wiley & Sons, 2001.

BECK, K. *et al.* **Manifesto for agile software development**. 2001a. Disponível em <http://agilemanifesto.org>. Acesso em: 07 abr. 2019.

BECK, K. *et al.* **Principles behind the agile manifesto**. 2001b. Disponível em <http://agilemanifesto.org/principles.html>. Acesso em: 07 abr. 2019.

BOMFIM T.; LENY, A. Grupos focais: conceitos, procedimentos e reflexões baseadas em experiências com o uso da técnica em pesquisas de saúde. **Physis - Revista de Saúde Coletiva**, v.19, n. 3, 2009. ISSN 0103-7331. Disponível em: <http://www.redalyc.org/articulo.oa?id=400838224013>. Acesso em: 21 jan. 2018.

CHAPMAN, R. J. **The rules of project risk management: implementation guidelines for major projects**. London: Routledge, 2016.

COCKBURN, A; HIGHSMITH, J. Agile software development: the people factor. **Computer**, n. 11, p. 131-133, 2001.

CORMICAN, K.; O'SULLIVAN, D. A collaborative knowledge management tool for product innovation management. **International Journal of Technology Management**, v. 26, n. 1, p. 53-67, 2003.

DALL'AGNOL, C. M.; TRENCH, M. H. Grupos focais como estratégia metodológica em pesquisas na enfermagem. **Revista gaúcha de enfermagem**, Porto Alegre. v. 20, n. 1 (jan. 1999), p. 5-25, 1999.

DEMARCO, T; LISTER, T. **Waltzing with bears: managing risk on software projects**. Addison-Wesley, 2013.

DRESCH, A.; LACERDA, D. P.; JÚNIOR, J. A. V. A. **Design science research: método de pesquisa para avanço da ciência e tecnologia**. Porto Alegre: Bookman Editora, 2015

DULOCK, M. J.; LONG, H. Digital Collections are a sprint, not a marathon: adapting scrum project management techniques to library digital initiatives. **Information Technology & Libraries**, v. 34, n. 4, p. 5-17, 2015.

CAMPOS, A.; FONSECA, I. Por que SCRUM?. **Engenharia de Software Magazine**, Rio de Janeiro, Devmedia Group, 4. ed., set. 2008. Disponível em: <http://www.devmedia.com.br/articles/viewcomp.asp?comp=9868>. Acesso em: 04 fev. 2019.

COLLIS, J; HUSSEY, R. **Pesquisa em administração: um guia prático para alunos de graduação e pós-graduação**. Porto Alegre: Bookman, 2005.

GAGANDEEP, S. **Use of knowledge management techniques for risk management**. 2012. Master's Thesis (Master's Programme International Project Management) – Department of Civil and Environmental Engineering, Chalmers University of Technology, Göteborg, 2012.

GARLAND, D.; FAIRBANKS, G. **Just enough architecture: a risk-driven approach**. Boulder: Marshall & Brainerd, 2010.

GONZALEZ, R. V. D.; MARTINS, M. F. O Processo de gestão do conhecimento: uma pesquisa teórico-conceitual. **Gest. Prod.**, São Carlos, v. 24, n. 2, p. 248-265, jun. 2017.

GUSMÃO, C. M. G. **Um modelo de processo de gestão de riscos para ambientes de múltiplos projetos de desenvolvimento de software**. 2007. Tese (Doutorado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2007.

JAMIL, G. L. **Gestão da informação e do conhecimento em empresas brasileiras: estudos de múltiplos casos**. 2005. Tese (Doutorado em Ciências da Informação) – Escola de Ciência da Informação, Universidade Federal de Minas Gerais, Belo Horizonte, 2005.

HALTIWANGER, G. S. **Establishing relationships between risk management and knowledge transfer**. 2012. Doctor of Philosophy (PhD) – Engineering Management, Old Dominion University, DOI: 10.25777/1k9t-2d73

LINDVALL, M.; BASILI, V. R.; BOEHM, B.; COSTA, P.; DANGLE, K.; SHULL, F.; TESORIERO, R.; WILLIAMS, L.; ZELKWITZ, M. V. Empirical findings in agile methods. **Proceedings of Extreme Programming and agile methods XP/Agile Universe**. p. 197-207, 2002.

MAGNIER-WATANABLE, R.; SENOO, D. Organizational characteristics as prescriptive factors of knowledge initiatives. **Journal of Knowledge Management**, v. 12, n. 1, p. 21-36, 2008.

MORAN, Alan. **Agile risk management**. Springer: Cham, 2014.

MENTZER, J. T.; FLINT, D. J. Validity in logistics research. **Journal of business logistics**, v. 18, n. 1, 1997.

- MORGAN, D. L. **Focus group as qualitative research**. London: Sage, 1997.
- NEVES, S. M. *et al.* Risk management in software projects through knowledge management techniques: cases in Brazilian incubated technology-based firms. **International Journal of Project Management**, v. 32, n. 1, p. 125-138, 2014.
- NEEF, D. Managing corporate risk through better knowledge management. **The Learning Organization**, v. 12, n. 2, p. 112-124, 2005.
- NONAKA, I.; TAKEUCHI, H. **Criação de conhecimento na empresa**: como as empresas japonesas geram a dinâmica da inovação. 3. ed. Rio de Janeiro: Campus, 1997.
- OLIVEIRA, L. R.; GOMES, G. S.; LIMA, F. P. Análise de riscos pelo uso de métodos ágeis na gestão de projetos de desenvolvimento de software. **Gestão e Projetos: GeP**, v. 5, n. 2, p. 90-101, 2014.
- PINHO, I.; REGO, A.; CUNHA, M. P. Improving knowledge management processes: a hybrid positive approach. **Journal of Knowledge Management**, v. 16, n. 2, p. 215-242, 2012.
- PROJECT MANAGEMENT INSTITUTE. **A guide to the project management body of knowledge (PMBOK® Guide)**. 5. ed. Pennsylvania: PMI, 2012.
- PRESSMAN, R.; MAXIM, B. R. **Engenharia de Software**: uma abordagem profissional, 8. ed. Porto Alegre: Mc Graw Hill Education, 2016.
- POLANYI, Michael. **The tacit dimension**. Garden City: Anchor, 1967.
- SAKAI, E.; RICARTE, I. L. M. Gestão do conhecimento em empresas de desenvolvimento ágil de *software*: conceitos e práticas essenciais. **Journal on Advances in Theoretical and Applied Informatics**, v. 1, n. 1, p. 13-21, 2015.
- SAKAI, E. **Gestão do conhecimento no desenvolvimento ágil de software**: técnicas que promovem a criação do conhecimento organizacional. 2017. Dissertação (Mestrado) Universidade Estadual Campinas, Campinas, 2017.
- SCHWABER, K.; SUTHERLAND, J. **The scrum guide**: the definitive guide to scrum: the rules of the game, 2013. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>. Acesso em: 15 mar. 2019.
- SCRUM. **The home of Scrum**. Disponível em: <https://www.scrum.org/>. Acesso em: 22 jun. 2019.
- SOMMERVILLE, I. **Engenharia de software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.
- STANDISH GROUP. **Chaos report**. 586 Old Kings Highway, Dennis, MA 02638, USA, 2015.

TAVARES, B. G. **Análise da Gestão de Riscos no desenvolvimento de projetos de software via Scrum**. 2015. Dissertação (Mestrado em Engenharia de Produção) – Universidade Federal de Itajubá, Itajubá, 2015.

TRIOLA, M. F. **Introdução à estatística**. 7. ed. Rio de Janeiro: LTC, 1999.

VARGAS, L. M. Gerenciamento ágil de projetos em desenvolvimento de *software*: um estudo comparativo sobre a aplicabilidade do Scrum em conjunto com PMBOK e/ou PRINCE2. **Revista de Gestão e Projetos - GeP**, [S.l.], v. 7, n. 3, p. 48-60, dec. 2016. ISSN 2236-0972.

VERSIONONE, 2017. **12th Annual State of Agile Development Survey**. Disponível em: <http://stateofagile.versionone.com>. Acesso em: 18 dez. 2017.

ZACK, M. H. Managing codified knowledge. **Sloan Management Review**, v. 40, n. 4, p. 45-59, 1999.

Z Aidan, F. H. Processo de desenvolvimento de sistemas de informação como forma de retenção do conhecimento organizacional para aplicação estratégica: um estudo de múltiplos casos. 2008. 130 f. 2008. Dissertação (Mestrado)–FUMEC, BH.