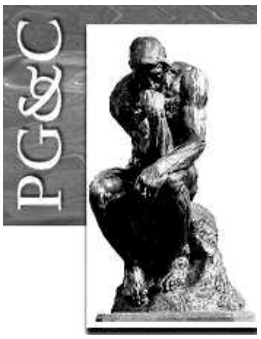


# ANÁLISE DO RELACIONAMENTO DA GESTÃO DO CONHECIMENTO ESTRUTURADA NA IMPLANTAÇÃO DE MÉTODOS ÁGEIS EM EMPRESA DE DESENVOLVIMENTO DE *SOFTWARE*

**Diego Marconi Candal**
Mestrando em Informática e Gestão do Conhecimento pela Universidade Nove de Julho, Brasil. Diretor da DMC Tecnologia, Computação e Inovação, Brasil.
E-mail: diegocandal@uni9.edu.br

**Ivanir Costa**
Doutor em Engenharia de Produção pela Universidade de São Paulo, Brasil. Professor da Universidade Nove de Julho, Brasil.
E-mail: ivanirc@uni9.pro.br

**Marcos Antonio Gaspar**
Doutor em Administração pela Universidade de São Paulo, Brasil. Professor da Universidade Nove de Julho, Brasil.
E-mail: marcos.antonio@uni9.pro.br

**Vinicius Rodrigues Pereira dos Santos**
Mestrando em Informática e Gestão do Conhecimento pela Universidade Nove de Julho, Brasil. Analista da Concremat Engenharia e Tecnologia, Brasil.
E-mail: vinicius.rodrigues@gmail.com

**Resumo**
O objetivo deste estudo é analisar a importância da Gestão do Conhecimento estruturada, na Implantação de Métodos Ágeis em empresas de desenvolvimento de software. Para tanto, foi realizada uma pesquisa exploratória, utilizando-se de revisão sistemática da literatura, aliada à análise documental em fontes especializadas em Métodos Ágeis e Gestão do Conhecimento. Como resultado imediato da revisão executada, verificou-se a existência de somente doze artigos que relacionam Gestão do Conhecimento quando se passa a utilizar Métodos Ágeis nas organizações. Após o devido mapeamento dos estudos encontrados, foram analisados os dois Métodos Ágeis mais utilizados na atualidade (Scrum e Kanban), bem como levantados os requisitos necessários para a implantação destes nas empresas de desenvolvimento de software. Os resultados apontam as peculiaridades existentes em cada método estudado em relação à Gestão do Conhecimento, bem como a importância de se considerar a Gestão do Conhecimento estruturada durante as etapas ou ciclos de desenvolvimento dos Métodos Ágeis analisados. Isto porque a implantação de métodos ágeis suportada por um processo de gestão de conhecimentos estruturado pode trazer benefícios para a empresa, tais como melhoria no processo de desenvolvimento de software, eliminação de desperdício, além de imprimir a mentalidade voltada aos diferenciais de mercado, com foco no que agrega valor ao cliente.

**Palavras-chave:** Gestão do conhecimento. Métodos ágeis. Desenvolvimento de software. Scrum. Kanban.

***ANALYSIS OF THE RELATIONSHIP OF STRUCTURED KNOWLEDGE MANAGEMENT IN THE IMPLEMENTATION OF AGILE METHODS IN SOFTWARE DEVELOPMENT COMPANY***

***Abstract***
*The objective of this study is to analyze the importance of structured Knowledge Management, in the Implementation of Agile Methods in software development companies. To achieve this goal, an exploratory research was carried out, using a systematic literature review, combined with documentary analysis in specialized sources in Agile Methods and Knowledge Management. As an immediate result, we found just 12 articles relating to the topic Knowledge Management when applying Agile Methods in organizations. And thus, after the proper mapping of the studies found, the two Agile Methods most used nowadays (Scrum and Kanban) were analyzed, as well as the necessary requirements for their implementation in software development companies were raised. The results show the peculiarities existing in each method studied in relation to Knowledge Management, as well as the importance of considering Knowledge Management structured during the stages or cycles of development of the analyzed Agile Methods. This occurs because the implementation of agile methods supported by a structured knowledge management process can bring benefits to the company, such as improvement in the software development process, elimination of waste and, in addition to stimulate the mindset focused on market differentials, focusing on what adds value to the customer.*

***Keywords:*** *Knowledge management. Agile methods. Software development. Scrum. Kanban.*

## 1 INTRODUÇÃO

Ao longo dos últimos anos, as organizações têm migrado e amadurecido seu conceito de produção, onde o trabalho deixa de ser o principal ativo, sendo substituído pelo conhecimento. Neste contexto, a rápida produção de conhecimento, que surge para acompanhar as mudanças e inovações, tem criado um volume de informações que precisa ser compartilhado também rapidamente, com o objetivo de agregar o maior valor possível ao produto ou serviço a ser entregue ao cliente (NONAKA; TAKEUCHI, 2008; ROGERS, 2017). Ao mesmo tempo, observa-se um aumento na demanda por soluções mais personalizadas e com prazos para desenvolvimento e produção mais curtos, sendo que a agilidade acabou por se tornar requisito para que as empresas mantenham seu diferencial de mercado e o consequente poder competitivo que acionistas e gestores almejam.

Associado a esta demanda surgiram, a partir do Manifesto Ágil de 2001 os denominados Métodos Ágeis (MA), entre eles o Scrum, a filosofia Lean e o KanBan conforme indica FERNANDES et al. (2019). Desde então, as empresas de software vêm adotando os MA. Conforme indicado pelo Agile Report (2020), 95% das empresas já adotaram algum MA, sendo que destas, 37% implantaram no setor de desenvolvimento de software, visando auxiliar ou controlar completamente o Plano de Desenvolvimento de Software (PDS) estipulado, sendo que em 71% destas teve como principal razão a busca por acelerar a entrega de produtos de software.

A procura pela Implantação de Métodos Ágeis (IMA) nas empresas tem origem em diferentes anseios, dentre os quais se destacam: acelerar a entrega de software, promover a habilidade para tratar mudanças de prioridade, aumentar a produtividade, aperfeiçoar o alinhamento da Tecnologia da Informação (TI) ao negócio e, por fim, melhorar a qualidade do software (AGILE REPORT, 2020). Observando-se os objetivos que se pretende alcançar com a IMA e através da análise de estudos focados nas dificuldades e desafios desta implantação, entende-se que durante este processo, surgem novos problemas e também são criados novos entendimentos de situações específicas desafiando os envolvidos, tais como cultura

organizacional e melhoria de processos (GREGORY, 2016). O mau uso de ferramentas, a distorção de valores e indevida alteração de práticas inerentes aos MA, também podem ocorrer já que em alguns casos o despreparo ou a falta de habilidade técnica da equipe, pode fazer com que eles tentem alterar por exemplo o Scrum, postergando o deadline da Sprint (ciclo iterativo), ao invés de resolver o problema durante a sprint (PRIKLADNICKI, 2014 p.34).

Atualmente, conforme a Agile Alliance (AGILE ALLIANCE, 2020), existem mapeados 75 métodos/técnicas ágeis. Neste estudo foram analisados somente os dois métodos mais utilizados de forma pura, ou seja, sem que exista uma mistura de métodos e técnicas. Assim, Scrum (57%) e KanBan (7%) são indicados como os dois MA mais utilizados de forma pura, para o desenvolvimento de software no mercado, pelas organizações contemporâneas (AGILE REPORT, 2020). Dessa forma esses MA são o foco dessa pesquisa.

Em empresas de desenvolvimento de software, para que os colaboradores e líderes de equipe façam seu trabalho, de acordo com as necessidades que o mercado exige, o compartilhamento e o acesso rápido e eficiente ao conhecimento, tanto sobre a aplicação dos MA quanto no domínio dos MA, das regras de negócio e dos requisitos de software, precisam ser garantidos a partir de um processo de Gestão do Conhecimento (GC) bem estruturado. A GC, por sua vez, permite uma tomada de decisão assertiva e eficiente que gere valor para o cliente de forma satisfatória, a partir de um PDS sustentável e de qualidade dentro dos MA, que por sua vez, conforme (SOMMERVILLE, 2018 p.678), é caracterizado por ser informal, dispensando a utilização de documentos e fundamentado na existência de uma cultura organizacional voltada para a qualidade, em conjunto da aplicação de boas práticas.

Alguns autores mencionam a possibilidade de se manter de forma sustentável o PDS, assim como as manutenções de software, sem a necessidade de uma complexa e extensa documentação, conforme indicado por Gomes (2014) e segundo o qual, para tanto bastaria escrever um código de extrema qualidade. Em relação a qualidade do código do software, ela é uma das premissas do método ágil XP (eXtremme Programming), em que não se utiliza documentação tradicional, mas tem como base os indivíduos, prezando também por um código bem escrito e padronizado, com comentários que permitam que qualquer programador consiga alterá-lo. Baseando-se nos valores do XP é possível manter o processo de desenvolvimento de software utilizando o conhecimento tácito presente na equipe, já que esses valores permitiriam este tipo de prática, conforme (PRIKLADNICKI, 2014 p.42). Porém, durante esta pesquisa não foram encontrados autores que prezam pela inexistência de documentação do software, o que não surpreende, já que conforme o próprio Manifesto Ágil, a documentação precisa existir, todavia não deve prevalecer sobre a entrega de software funcionando e tão pouco sobre as conversas face a face, conforme o 6º e 7º princípios do manifesto ágil que indicam “O método mais eficiente e eficaz de transmitir informações para, e por dentro de, um time de desenvolvimento é através de uma conversa face a face.” e “software funcional é a medida primária de progresso”, conforme (FERNANDES et al., 2019 p.137).

Em contraponto, autores tradicionais como Sommerville (2018), argumentam que a documentação de software é a ferramenta fundamental para o engenheiro de software, pois no momento que surgem as manutenções ou alterações que vão além do layout de telas do sistema é esta documentação que descreve os requisitos do sistema e, consequentemente, a forma que ele opera. Mesmo com a relevância e benefícios que a documentação de software sugere, sendo um tópico pertinente à GC, tem sido observado, por diversos motivos, um desincentivo à documentação de software, gerando desta forma riscos, ao longo do tempo, às empresas de software (FERNANDES et al., 2019).

É importante mencionar que a documentação de software considera o armazenamento de documentos num determinado formato padronizado, além de formas e permissões de acesso aos profissionais envolvidos. Nesse sentido, volta-se a tangenciar a GC

que, ao fazer uso de práticas e ferramentas adequadas para cada departamento ou modelo de negócio, é responsável por criar, armazenar, manter e disponibilizar conhecimento de forma adequada na empresa.

Levando-se em consideração as análises das pesquisas realizadas, buscou-se responder à seguinte pergunta de pesquisa: Qual o relacionamento de uma GC estruturada na implantação de MA em empresa de desenvolvimento de software?

Para responder à questão de pesquisa, o objetivo deste estudo é analisar a importância da Gestão do Conhecimento estruturada, na Implantação de Métodos Ágeis em empresas de desenvolvimento de software. Para tanto, serão identificados e analisados os requisitos para a implantação dos MA Scrum e Kanban e os desafios de sua implantação quando não existir na empresa a GC estruturada.

## 2 REFERENCIAL TEÓRICO

### 2.1 Gestão do conhecimento estruturada voltada ao desenvolvimento de *software*

Muitos autores se referem à GC de forma mais social, humana e pura, do que simplesmente ferramentas e práticas aplicáveis, o que guarda estreita relação com as empresas de software, já que como a maioria das empresas, estas são compostas por pessoas, que por sua vez desenvolvem o software, utilizando e gerando conhecimento. Drucker (2001, p. 10) relaciona a GC com as empresas ao discorrer que “a empresa típica será baseada no conhecimento e que principalmente as grandes empresas não têm muita escolha quanto a se converterem em organizações baseadas em conhecimento”. Com esse conceito presume-se que empresas de software possam trabalhar de forma mais eficiente ao gerirem os conhecimentos aplicáveis ao seu negócio de atuação. Isto porque o conhecimento é a matéria prima utilizada no PDS, ou seja, é o elemento fundamental para a criação de produtos ou execução de manutenções e atualizações em produtos já existentes.

Rus e Lindvall (2002, p. 29) advertem sobre a importância da GC na dimensão operacional das empresas quando mencionam “a importância de os indivíduos possuírem acesso às informações e conhecimentos corretos quando precisarem completar uma tarefa do processo ou tomarem uma decisão”. Chong e Choi (2005) e Trevisan et al. (2019) indicam onze Fatores Críticos de Sucesso (FCS) para que uma empresa de software consiga implantar e manter um sistema de GC bem estruturado: 1) treinamento dos funcionários, 2) envolvimento dos trabalhadores, 3) trabalho em equipe, 4) empoderamento de funcionários, 5) liderança da alta gerência, 6) infraestrutura de sistemas de informação, 7) medição de desempenho, 8) cultura favorável, 9) análise comparativa (benchmarking), 10) estrutura de conhecimento e, por fim, 11) eliminação de limitações organizacionais.

Ainda conforme Trevisan et al. (2019, p. 74), os FCS apresentados acima, para que possibilitem a implantação de um sistema de GC estruturado “são especialmente imprescindíveis e necessitam estar presentes apoiando a organização nos seus processos intensivos em conhecimento”. Os autores ainda complementam que, “se as organizações quiserem competir no mercado com sucesso e almejem alcançar crescimento nos negócios, igual atenção e ênfase devem ser dadas a todos os FCS mencionados”. Em adição, os autores argumentam ainda que os FCS “fornecerão uma perspectiva melhor de como gerenciar as atividades de conhecimento, a fim de maximizar a eficácia relacionada ao conhecimento e aos ativos organizacionais”.

Na visão de Rus e Lindvall (2002, p. 26), “os engenheiros de software vêm se engajando em atividades relacionadas à GC voltadas ao aprendizado, captura e reuso de experiências, há muito tempo”. Este argumento também é evidenciado por Gaspar et al. (2016) ao discorrerem que as empresas pertencentes, a indústria de software em especial,

adotam como premissa básica de sua operação, a geração e disseminação de conhecimento, o que exige que esse tipo de empresa considere a GC relevante ao seu negócio. Ainda conforme esses autores, após pesquisa realizada em empresas brasileiras de desenvolvimento de software de médio ou grande porte foram mapeadas as práticas e ferramentas de TI voltadas à GC implantada de forma estruturada.

Em suma, sem práticas específicas de GC e sem as devidas ferramentas de TI a ela associadas, assim como em caso de inexistência de apoio por parte dos stakeholders, a GC não pode ser implantada de forma eficiente na empresa. Como forma de auxiliar as empresas a implantarem um sistema de GC estruturado foi lançada a norma ISO 30401:2018, com o objetivo de instituir princípios e requisitos para a organização estabelecer, implementar, manter, rever e incrementar um sistema de GC efetivo (ISO, 2018). Tal normatização estabelece assim os princípios, requisitos e diretrizes para a implantação de um sistema de GC estruturado.

### 2.2 Manifesto Ágil e a agilidade para desenvolvimento de *software*

No ano de 2001, um grupo de desenvolvedores interessados em software simples, rápido, iterativo e de qualidade, formaram a 'Aliança Ágil' (AGILE ALLIANCE, 2020). Essa aliança propôs um manifesto (MANIFESTO, 2001) e uma declaração de princípios. O manifesto reuniu representantes de diferentes métodos considerados ágeis que vinham sendo propostos, como o Extreming Programming (XP), Scrum, Crystal, entre outros, além de pessoas interessadas em novas abordagens diferentes das anteriores, orientadas a extensa documentação e muitas vezes morosas (LARMAN, 2004).

O Manifesto Ágil, de acordo com Cockburn e Higsmiyh (2001), promove as melhores formas de desenvolver softwares, por meio dos seus 4 valores: 1) Indivíduos e interações mais que processos e ferramentas; 2) Software em funcionamento mais que documentação abrangente; 3) Colaboração com o cliente mais que negociação de contratos; 4) Responder a mudanças mais que seguir um plano.

Esses valores podem ser traduzidos nos doze princípios dos MA também listados por Cockburn e Higsmiyh (2001) e Sabbagh (2013), quais sejam: 1) Prioridade é satisfazer o cliente; 2) Mudanças nos requisitos são bem-vindas; 3) Entregar frequentemente software funcionando; 4) Pessoas de negócio e desenvolvedores devem diariamente em conjunto por todo o projeto; 5) Construir projetos em torno de indivíduos motivados; 6) Interação do time face a face; 7) Software funcionando é a medida primaria de progresso; 8) Os processos ágeis promovem desenvolvimento sustentável; 9) A continua atenção a excelência técnica e bom projeto aumenta a agilidade; 10) Simplicidade, pois é essencial evitar-se o desperdício no desenvolvimento do produto; 11) As melhores arquiteturas, requisitos, e projetos emergem de equipes auto organizáveis e com maior autonomia; 12) Em intervalos regulares a equipe refina e ajusta seu comportamento para de tornar cada vez mais efetiva, promovendo a melhoria incremental contínua.

Os métodos que seguem esses princípios são considerados como ágeis. Segundo Munoz e Oktaba (2011), os MA representam uma alternativa para o desenvolvimento de sistemas e de software, com foco no fator humano, do produto de software, e aumentando o relacionamento com clientes. Estes métodos fornecem entregas frequentes de software funcionando, permitindo alterações de requisitos e envolvimento direto do cliente.

### 2.3 Requisitos para implantação de métodos ágeis para desenvolvimento de *software*

Nas documentações oficiais dos métodos selecionados para este estudo (Scrum e Kanban), buscou-se os requisitos necessários para a suas implantações. Os dois métodos foram

selecionados utilizando-se como base as seguintes premissas: a) deve ser um método puro, ou seja, não pode ser híbrido ou uma mistura de diferentes métodos ou técnicas; b) deve ser amplamente utilizado pelas empresas.

Os dois MA puros que se enquadraram nas premissas indicadas foram o Scrum (57% de aplicação por empresas no mercado) e o KanBan (7% de aplicação por empresas no mercado) (AGILE ALLIANCE, 2020). Foi possível identificar ainda elevada quantidade de métodos mistos/híbridos já catalogados. Alguns destes métodos estão se destacando, a exemplo do Scrumban (um híbrido entre Scrum e Kanban) que, conforme o Agile Report (2020), conseguiu 10% de aderência junto aos profissionais praticantes de ágil. Porém, por não ser considerado um método puro, foi desconsiderado para fins desta pesquisa.

Durante a condução desta pesquisa não foi encontrada classificação para os pré-requisitos da IMA envolvendo os dois MA selecionados (Scrum e Kanban). Portanto, optou-se por determinar estas classificações para assim dividir os requisitos dos MA escolhidos em duas categorias: requisitos fundamentais e requisitos opcionais. Tal segregação tomou por base a intenção precípua deste estudo em elencar todos os requisitos disponíveis para que somente em seguida cada requisito fosse devidamente classificado por grau de importância, baseando-se nas categorizações descritivas conforme segue:

- Requisitos fundamentais: são aqueles requisitos considerados indispensáveis. Caso um requisito pertencente a esta categoria não seja encontrado na cultura, nos processos ou na infraestrutura da organização, é necessário que a maturidade dos processos seja melhorada ou que a cultura organizacional seja revista, ou ainda que a infraestrutura seja atualizada, para que somente então seja considerada a possibilidade da IMA.
- Requisitos opcionais: são requisitos que, apesar de serem recomendados, não são obrigatórios ou essenciais, sendo que durante ou após a IMA, estes poderiam continuar a ser implantados sem grande prejuízo.

### 2.4 Requisitos mapeados para implantação do Scrum no processo de desenvolvimento de *software*

Scrum é um framework que permite aos profissionais tratarem e resolverem problemas complexos e adaptativos, enquanto entregam produtos com o mais alto valor possível de forma produtiva e criativa. Tal definição consiste na designação de times Scrum associados a papéis, eventos, artefatos e regras. Cada componente no framework serve a um propósito específico, sendo essencial para o uso e sucesso do Scrum (SCRUM GUIDES, 2020).

No Scrum Guide, também são descritas as regras do framework Scrum. Este documento oficial é assinado, autorizado e mantido pelos criadores do Scrum, Ken Schwaber e Jeff Sutherland. Leve, simples de entender e difícil de dominar, são as características do Scrum descritas em documentação. Também é definido por seus criadores que o Scrum somente funcionará e será Scrum, se ele existir em sua totalidade, ou seja, com todos os seus papéis, eventos, artefatos e regras (SCRUM GUIDES, 2020).

Após análise do documento oficial do Scrum e com base nos valores do Scrum (comprometimento, coragem, foco, abertura e respeito), entende-se que para ser implantado com sucesso, o Scrum necessitaria dos requisitos expostos no Quadro 1:

**Quadro 1** – Mapeamento dos requisitos para implantação do Scrum

| Requisitos | Tipo | É parte da GC ? |
|---|---|---|
| Equipe pequena | Fundamental | Não |
| Comprometimento da equipe | Fundamental | Não |
| Stakeholders conscientes | Fundamental | Não |
| Foco na entrega | Fundamental | Não |
| Repositório de conhecimento | Fundamental | Sim |
| Facilidade de adaptação | Fundamental | Não |

Fonte: Autores

Ainda conforme o Scrum Guides (2020), os membros do Time Scrum aprendem e exploram estes valores à medida que trabalham com os eventos, papéis e artefatos do Scrum. Ou seja, imagina-se que os profissionais aprenderiam e internalizariam a cultura de GC e a cultura ágil, conforme o próprio Scrum fosse sendo implantado na empresa, bastaria então seguir os próprios valores indicados pelo Scrum.

## 2.5 Requisitos mapeados para implantação do Kanban no processo de desenvolvimento de *software*

O Kanban teve origem na indústria automobilística em 1960, tendo sido criado por Taiichi Ohno. Ele consiste num gerenciador de fluxo de tarefas que quando idealizado, teve como objetivo auxiliar o método de produção Just In Time (JIT). De acordo com Monden (2015, p. 9), "o Kanban é um sistema de informações que harmoniosamente controla as quantidades da produção, em cada processo".

O Kanban é um tipo de ferramenta que, quando mal utilizada pode trazer problemas. Para que seja devidamente utilizada em conformidade com seus princípios, foi estabelecida uma definição mais simplificada: o objetivo do Kanban é alcançar e conquistar o JIT. Kanban é o nervo central da produção, dando aos colaboradores poder de decisão e autonomia necessárias para que iniciem o trabalho sem que recebam orientação para isso.

OHNO (2019) destaca que as seis regras do Kanban precisam ser seguidas para que se obtenha os ganhos esperados em termos de produtividade e eliminação de desperdícios. As seis regras do Kanban são:

> 1) o processo posterior vai ao processo anterior para pegar produtos; 2) o processo anterior produz itens na quantidade e sequência indicadas pelo Kanban; 3) é proibido pegar ou produzir entregáveis sem um Kanban; 4) requer que um Kanban esteja anexado a cada entregável; 5) nada que possua defeito deve ser enviado ao processo seguinte, resultando em um processo 100% isento de produtos defeituosos; e 6) reduzir o número de Kanban aumenta a sua sensibilidade (OHNO, 2019, p. 30).

Durante a condução desta pesquisa não foram encontrados requisitos mínimos explicitamente citados por autores para o Kanban. Portanto, utilizou-se os mesmos requisitos anteriormente indicados para o Scrum, Quadro 1, para que seja possível estabelecer uma comparação a contento.

## 3 MÉTODO E MATERIAIS

Esta é uma pesquisa exploratória-qualitativa com objetivo de estabelecer a correlação entre os temas 'Gestão do Conhecimento estruturada' e 'implantação de métodos ágeis em empresas de software'.

Para identificar os possíveis requisitos para os MA selecionados (Scrum e Kanban) foi consultada a documentação mais atual disponível destes. No caso do Scrum foi utilizada a última versão da documentação disponível, o Scrum Guide 2017 (SCRUM GUIDES, 2020). Já no caso do Kanban foi consultada bibliografia específica referente ao método JIT, que é suportado e auxiliado pelo Kanban.

No caso específico do Kanban, por não haver documentação oficial, optou-se por consultar a bibliografia original do criador do Kanban e a partir destes documentos foi possível efetuar o cruzamento das informações encontradas, buscando assim explicitar as exigências para se obter a IMA e o que uma sólida GC oferece para cada um dos métodos.

Além da documentação disponível para os MA escolhidos, também foi conduzida revisão sistemática da literatura utilizando-se das seguintes bases de dados (Web os Science, Scopus e ACM Digital Library). Na busca efetuada foram encontrados 21 artigos, tendo como termos de busca a expressão 'Gestão do Conhecimento AND Métodos Ágeis' ou 'Knowledge Management AND Agile Methods', no período de 2016 a junho de 2020.

Obteve-se 16 trabalhos, uma vez que se detectou cinco trabalhos repetidos em diferentes bases e quatro trabalhos que não tinham alinhamento ao objetivo desta pesquisa e que foram excluídos, resultando em 12 trabalhos para uma revisão detalhada.

É oportuno ressaltar que alguns artigos não tinham seu conteúdo completo disponível e por esta razão, nestes casos específicos foi feita a leitura apenas do título e resumo dos artigos. Para os demais documentos encontrados na pesquisa foi realizada a leitura do resumo, introdução e conclusão de cada trabalho. A análise destes artigos pretendia responder à seguinte pergunta de pesquisa: "Qual o relacionamento de uma GC estruturada na implantação de MA em empresas de desenvolvimento de software?". No Quadro 2 é exposto um breve relato sobre os doze artigos que foram analisados.

No Quadro 2 nota-se a evidenciação de um termo que apareceu em muitos artigos analisados durante a revisão sistemática. O termo "Global Software Development" (GSD) que faz menção ao PDS, quando se considera o envolvimento de profissionais ou equipes distribuídos por diversos países.

**Quadro 2** – Descritivo da revisão sistemática de artigos encontrados na pesquisa

| Autor | Objetivos | Resultados | Relatos | Relacionamento da GC e a IMA |
|---|---|---|---|---|
| Gregory (2016) | Responder duas questões de pesquisa (1ª: com quais desafios os praticantes de ágil se deparam? 2ª: Como estes desafios se manifestam no ambiente organizacional ). | Os desafios são cumulativos e vão se empilhando, pois ao mesmo tempo que se tenta migrar para os MA, existem outros problemas de negócio que simultaneamente afetam o praticante de ágil. Notou-se que os desafios do nível de governança são pouco pesquisados. Como resultado de um dos estudos feitos, foi criada uma tabela com os desafios devidamente | É possível entender que conforme o tempo passa e os MA se consolidam, o entendimento de algumas situações é realizado e surgem novas pesquisas e novos questionamentos. | A questão da GC não é listada como um desafio neste estudo, pois existem outros pilares pendentes para que o ágil traga o resultado esperado.<br>As pessoas e a cultura organizacional e melhoria de processos são apontados como desafios.<br>Ou seja, existem sérios problemas sendo enfrentados pelos membros de equipe ao implantar ágil e por alguma razão os praticantes de ágil demonstraram |

| | | | | |
|---|---|---|---|---|
| | | separados por grupos. | | desinteresse e relevam a importância da GC no PDS. |
| Rolland (2016) | Permitir através de estudo de caso, que seja identificado o que permite aos MA, que estes sejam utilizados em grandes projetos e de que forma. | Foi possível escalar os métodos ágeis no projeto devido a entendimentos ao longo do projeto, que levaram a inserção de itens adicionais dentro dos MA, incluindo etapas ou novos processos que juntos permitiram o sucesso do projeto. | O estudo mostra que é inviável aplicar MA de forma pura em grandes projetos, sendo que foi necessário inserir novas práticas no PDS, para que fosse possível. Ainda assim a primeira entrega do projeto fracassou. | Entre as novas atitudes e práticas inseridas no PDS a grande maioria mencionava GC.<br>Um dos principais desafios foi a questão da transformação/tradução do conhecimento entre times e *stakeholders*. |
| Adnan (2017) | Melhorar a acuracidade na estimativa de esforço para novas tarefas de desenvolvimento e promover a GC, focando no Scrum e utilizando o modelo de ontologia de estimação por multiagente. | As abordagens utilizadas, com as técnicas conseguiram uma acuracidade maior do que as práticas utilizadas nos MA. Como o *planning poker*, que consiste em estimar em conjunto o tempo necessário para cada tarefa, fazendo uso de cartas com valores na busca de um consenso e *delph*i, onde especialistas de forma anônima têm acesso às previsões estimadas por outros especialistas que responderam questionários, de forma que todos reflitam sobre suas próprias estimativas, aprimorando-as. | Artigo propõe a utilização de um método baseado em ontologia, integrado ao Scrum e por fim conseguiu provar uma melhor eficiência se comparado a métodos mais tradicionais. | Durante este modelo, o conhecimento tácito do ágil, é transformado em explícito para que seja reaproveitado no futuro ao se estimar tempo para execução de tarefas.<br>Indica que ao perder membros da equipe, a empresa perde conhecimento quando se está utilizando MA sem a devida preocupação com GC. |
| Ahmed (2017) | Verificar os desafios existentes ao se implantar MA em GSD, utilizando revisão de literatura para fundamentar | Através de revisão de literatura feita com estudos entre 2014 e 2016. | Foram identificados 7 desafios que foram elencados de acordo com a frequência que ocorrem. | Os MA fazem forte uso de conhecimento tácito, enquanto o GSD é completamente baseado em conhecimento explícito e esta contradição implica em grandes desafios para as equipes envolvidas no |

| | | | | |
|---|---|---|---|---|
| | próximas pesquisas. | | | PDS. |
| Dingsøyr (2017) | Demonstrar como os MA foram adaptados e complementados com práticas tradicionais para suportar um grande projeto. | Uma descrição completa e modelo de um grande projeto de desenvolvimento de software, que combina MA com método tradicional. | Mostra como processos adicionais ao ágil, focados em envolvimento do cliente, arquitetura de software e coordenação entre times foi organizada com um número de funções extras ao nível do time de *"features"*. Apresenta alguns fenômenos desafiadores que ocorrem com projetos de grande escala. | Verificou-se que os times auto gerenciáveis inerente ao ágil precisam de uma efetiva rede de conhecimento e que o conhecimento tácito durante as reuniões (eventos do Scrum e outros MA) podem não ser interessantes para todos os participantes que ficam dispersos. Conclui-se que não existe um correto direcionamento para transmissão e compartilhamento do conhecimento. |
| Inayat (2017) | Entender como os integrantes do PDS lidam com a análise de requisitos utilizando MA quando a equipe está em um cenário de GSD. | Foram identificados alguns padrões de colaboração semelhantes aos utilizados no método tradicional de desenvolvimento de software. | Entende-se que inseridos em um GSD, os MA sempre dependem de práticas externas para que a equipe entregue software com sucesso. Os MA sozinhos não obteriam sucesso. | Verificou-se que os gerentes de projeto acabam sendo peças chaves nos processos de compartilhamento de conhecimento. Identificou-se que a distância não afeta em nada a GC em um cenário de GSD. |
| Mzwandile (2017) | Investigar práticas de GC em micro e pequenas empresas de software no Sul da África e verificar se a GC beneficiou estas empresas. | O estudo encontrou seis práticas de GC nas empresas: aquisição de conhecimento, criação, armazenamento, compartilhamento e aplicação. | O estudo identificou que a GC beneficiou as empresas tornando-as mais efetivas, eficientes e produtivas. | Mesmo quando aplicada em um formato mais informal, sem utilização de complexos procedimentos e normas, a GC traz benefícios para as empresas de software e permite aprimoramento dos processos e rotinas. |
| Voigt (2017) | Criar um método que permita a criação de documentação | Foi criado um método escalável que permite que os desenvolvedores documentem o | Autor corrobora a ideia de que os MA estão desprezando totalmente a | O método aborda diversas perguntas para determinar o que deve ser feito ou não para que a documentação seja |

| | | | | |
|---|---|---|---|---|
| | do software, mesmo quando se utiliza MA no PDS. | software sem corromper a filosofia ágil. | documentação de software e que isso pode trazer graves consequências principalmente quando existe saída de membros da equipe de desenvolvimento. | criada de forma enxuta e eficiente. |
| Khalil (2019) | Objetivo: Responder à questão: Como a engenharia ágil e práticas de colaboração ajudam a suportar sistemas de gestão do conhecimento ? Identificar de que forma as práticas ágeis suportam o gerenciamento de conhecimento em grandes contextos onde ajustes e colaboração estreita, são mais difíceis de alcançar. | Através de revisão de estudos empíricos, constatou que a GC caminha em paralelo com as práticas ágeis, através da criação de repositórios, comunicação constante e desenvolvimento iterativo | Nota-se uma comparação entre a "antiga GC" (com excessiva e complexa documentação) e a "nova GC", dentro de um novo cenário, mais dinâmico e com prazos mais curtos para a inovação. Salienta que no cenário atual o conhecimento se torna obsoleto rapidamente. | As práticas ágeis aprimoram as atividades de GC que envolvem tanto conhecimento tácito como explícito. Práticas ágeis contribuem para a criação de uma intensa cultura de conhecimento. A GC é enfatizada através da comunicação, desenvolvimento iterativo, repositórios de conhecimento e práticas de engenharia. Práticas de colaboração, como reuniões diárias, desenvolvimento iterativo, reuniões retrospectivas com o cliente e programação em pares (dois programadores trabalhando em um mesmo código), geram *feedback* e ampliam o conhecimento e seu compartilhamento. As práticas ágeis também tornam o conhecimento visível e disponível, criando um espaço de trabalho informativo. Sistemas de TI são fundamentais para criação, armazenamento, acesso, compartilhamento e uso do conhecimento. |
| Scatolino (2019) | Aplicar equações estruturais sobre | Foi percebido que ambos os construtos (GC e MA) da pesquisa | Constatou-se que é necessário que os gestores observem com | A influência da GC na qualidade de software é maior que a influência dos MA. |

| | | | | |
|---|---|---|---|---|
| | respostas obtidas através de um questionário desenvolvido para profissionais de TI., com a intenção de avaliar a influência da GC e MA na qualidade de software. | influenciam de forma positiva a qualidade do software desenvolvido. | mais atenção a GC aplicada sobre o PDS. | |
| Shameem (2020) | Identificar quais são as principais barreiras que possam minar a implantação de MA em GSD. | As barreiras encontradas foram divididas em 5 categorias e através de *survey* com especialistas e revisão da literatura. | Foi criado um ranking com estas barreiras encontradas classificando-as quanto impeditiva seria cada uma na implantação de MA. | Classificação das barreiras que envolvem GC:<br>Falta de treinamento em ágil - 5º lugar no ranking;<br>Documentação limitada do projeto - 14º lugar;<br>Falta de compartilhamento de conhecimento - 19º lugar. |
| Tenório (2020) | Discutir como seria possível que as atividades do Scrum suportassem o ciclo de GC para converter o conhecimento do indivíduo em conhecimento comum dentro das equipes de desenvolvimento de software. | A revisão da literatura indicou que o Scrum permite que o conhecimento se espalhe pelos membros da equipe. Para isso os autores uniram as etapas do Scrum ao ciclo de GC. Os autores relacionaram em uma tabela, as atividades que compõe o Scrum com as etapas do ciclo de GC. | Demonstra como é possível de forma sistêmica incorporar os ciclos de GC ao Scrum e desta forma o Scrum não seria utilizado apenas como uma ferramenta de Gestão; | Mostra que a GC precisa ser aplicada nas empresas junto com os métodos ágeis para que se obtenha melhores resultados. |

Fonte: Autores

Ao unir os resultados da revisão sistemática e os resultados da pesquisa documental executada a partir do cruzamento de dados, consegue-se o embasamento teórico suficiente para que sejam apresentadas as correlações entre os temas considerados nesta pesquisa. Ou seja, ao expor as correlações identificadas e as correlações não identificadas foi possível evidenciar a importância da GC quando do uso e da implantação de MA em empresas de desenvolvimento de software, foco deste artigo.

Com base no contexto apresentado na literatura, ou seja, de que o PDS tem como esteio o conhecimento, que por sua vez precisa ser tratado como um ativo da empresa de

software; foi possível relacionar a GC com a IMA em PDS. Foram encontrados 10 principais trabalhos, sendo seus autores ROLLAND (2016), ADNAN (2017), AHMED (2017), DINGSØYR (2017), MZWANDILE (2017), VOIGT (2017), KHALIL (2019), SCATOLINO (2019), SHAMEEM (2020) e TENÓRIO (2020), que discorrem sobre a GC aplicada em empresas de desenvolvimento de software, bem como, sobre a IMA e sugerindo ser benéfica a junção das duas práticas.

## 4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Constatou-se que muitos autores pesquisados na revisão da literatura identificam uma dependência entre GC e MA, sendo que em algumas situações os MA necessitam de apoio externo para que consigam prosperar e tornar o PDS sustentável (ROLLAND, 2016; ADNAN, 2017; DINGSØYR, 2017; MZWANDILE, 2017; KHALIL, 2019; SHAMEEM, 2020; TENÓRIO; 2020).

Em alguns estudos foi possível identificar ainda de forma significativa a preocupação com o conhecimento tácito dos MA que, sem o devido suporte da GC, seria perdido. Tal situação é extremamente prejudicial à empresa de software, visto que o custo para se obter aqueles conhecimentos será muito mais alto no futuro, principalmente em casos nos quais a rotatividade dos profissionais da equipe de desenvolvimento de software costuma ser elevada (ADNAN, 2017; AHMED, 2017; VOIGT, 2017).

Tal situação se agrava com o comportamento agressivo das empresas de software no mercado de trabalho em relação aos profissionais mais qualificados. Conforme indicado pela ComputerWorld (2019) em uma análise executada com meio bilhão de profissionais por uma plataforma especializada em carreiras, descobriu-se que as maiores taxas de rotatividade são registradas na indústria de software.

Nesse contexto, evidenciado na ComputerWorld (2019), mesmo entre as empresas consideradas gigantes de tecnologia, a média de permanência do colaborador é de aproximadamente 21 meses. Isto se deve ao fato de que o profissional, ao receber uma melhor proposta de trabalho, acaba por abandonar o atual emprego, sendo que o conjunto de conhecimento tácito deste profissional inserido no desenvolvimento do software que não foi transformado em conhecimento explícito, acaba sendo perdido junto com o profissional desenvolvedor que se desligou da empresa.

Em complemento, foram verificados também os requisitos de cada método ágil escolhido para este estudo (Scrum e Kanban). Os requisitos foram categorizados conforme exposto no Quadro 3 no qual indica-se cada requisito de MA e se este se insere ou não à GC. Uma observação importante é que o impacto somente existirá, quando um requisito que é criado durante a implantação da GC na empresa, estiver mapeado como requisito fundamental para o respectivo método analisado.

**Quadro 3** – Mapeamento dos requisitos dos métodos Scrum e Kanban e respectivo impacto na GC

| Método | Tipos de requisitos | Requisitos | Impacto na GC |
|---|---|---|---|
| Scrum | Fundamental | Equipe pequena | Não |
| Scrum | Fundamental | Comprometimento da equipe | Não |
| Scrum | Fundamental | Stakeholders conscientes | Não |
| Scrum | Fundamental | Foco na entrega | Não |
| Scrum | Fundamental | Repositório de conhecimento | Sim |
| Scrum | Fundamental | Facilidade de adaptação | Não |
| Kanban | Opcional | Equipe pequena | Não |
| Kanban | Fundamental | Comprometimento da equipe | Não |
| Kanban | Opcional | Stakeholders conscientes | Não |
| Kanban | Opcional | Foco na entrega | Não |

| | | | |
|---|---|---|---|
| Kanban | Opcional | Repositório de conhecimento | Sim |
| Kanban | Opcional | Facilidade de adaptação | Não |

Fonte: Autores

Conforme indicado no Scrum Guides (2017), o Scrum requer um repositório de conhecimento, pois o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido. Ou seja, o conhecimento que o Scrum gera precisa ser devidamente gerido, armazenado e compartilhado pela empresa. Sem a adequada GC os projetos desenvolvidos com o método ágil Scrum, em algum momento sofreria retrabalho, que significaria, por exemplo, refazer um módulo de software sem necessidade em função de não haver o conhecimento devidamente disponível na equipe de desenvolvedores de software.

Para a empresa é um risco manter o conjunto de conhecimentos dos profissionais de forma tácita apenas, conforme o Scrum tende a se posicionar como método de trabalho para o desenvolvimento de software. Tal situação não se enquadra nas premissas do pensamento ou filosofia pregada nos MA, que aponta para um processo enxuto, ou seja, fazer somente o necessário, quando for necessário, sem desperdício e fazendo certo da primeira vez (FERNANDES et al., 2019).

Já para o método ágil Kanban, foi entendido que este se apresenta como uma ferramenta que auxilia os profissionais no desenvolvimento de software, muitas vezes sendo aplicado em conjunto com o Scrum. Acredita-se que atualmente, sua principal função no processo de desenvolvimento de software seja permitir a fácil visualização dos estados dos processos à medida que evoluem. Isso não condiz com a idealização estipulada pelo Kanban oriunda do sistema Toyota de produção (filosofia JIT), além de estar além de sua capacidade de controle.

O único ponto que foi herdado corretamente, nos MA, da filosofia JIT e Kanban é a autonomia que os colaboradores e integrantes da equipe possuem para trocar de lugar as tarefas (cartões com histórias) de acompanhamento do processo de desenvolvimento de software. Isto porque, basicamente, as empresas adaptaram a ideia do quadro Kanban para manter um controle sobre o que está sendo executado, o que está parado, o que foi concluído e o que ainda precisa ser feito. O Kanban também permite a visualização simplificada da quantidade máxima de trabalho que conseguiria ser executada em cada ciclo de produção, por cada indivíduo. É importante frisar que o Kanban não define o ciclo, ou seja, é possível imaginar qualquer fluxo, repetitivo e contínuo como sendo um ciclo.

Para a implantação do Scrum, o risco maior detectado foi a perda de conhecimento tácito, já que em momento algum existe de forma clara e bem definida, o processo em que o conhecimento é registrado de forma explícita pela equipe de desenvolvimento de software, de maneira padronizada e passível de consulta posterior.

Já no método Kanban não foram detectados riscos em requisitos fundamentais. Isto talvez se deva ao fato de que não se trata de um método ou técnica que trabalhe diretamente com o conhecimento aplicado no desenvolvimento do produto. Ou seja, o Kanban apenas auxilia os profissionais na execução do processo, tendo como objetivo original garantir a disponibilidade dos materiais no processo produtivo da empresa na qual o produto é desenvolvido.

## 5 CONCLUSÕES

Os métodos ágeis possibilitam benefícios ao cotidiano dos profissionais inseridos em equipes envolvidas no PDS, pois permitem a padronização e proporcionam maior dinamismo ao PDS. Isto porque os MA contêm requisitos previamente definidos e gerenciáveis. De forma similar, a GC também permite o estabelecimento de controle sistemático na criação,

armazenamento e disponibilização de conhecimentos empregados no PDS. Estes conhecimentos, ao se utilizar MA, poderiam ser desperdiçados e perdidos, caso não sejam transformados em conhecimento explícito durante o PDS. Nota-se, portanto, que pode ser um risco as empresas de desenvolvimento de software serem ágeis em seus processos e se esquecerem da gestão de seu ativo mais importante: o conhecimento aplicado no desenvolvimento de software, tanto aquele utilizado para seguir as regras dos MA, quanto no desenvolvimento de projetos de software.

Após as análises efetuadas na pesquisa, é possível entender que cada MA se comporta de maneira específica e que, embora todos tenham seus pré-requisitos, sendo alguns bem definidos e outros sujeitos a interpretações subjetivas; nem todos necessitam de GC estruturada para que a IMA seja feita. Assim, a sua implantação pode trazer benefícios para a empresa, tais como melhoria no PDS, eliminação de desperdício, pensamento e mentalidade voltados para o diferencial de mercado com foco no que agrega valor ao cliente, entre outras possibilidades. É importante ressaltar que o principal foco de atenção ao se executar a IMA na organização deve ser nos profissionais e na cultura organizacional. Isto porque as ferramentas, de forma isolada, não têm utilidade quando um membro da equipe não estiver ciente de todo o processo que envolve aquela ferramenta. Ou seja, este colaborador também precisa estar motivado e ter o respaldo necessário de seus superiores para aquela transformação que está por vir na organização, como pregam os doze princípios do Manifesto Ágil.

Como toda mudança, o pensamento ou filosofia ágil e seus princípios, começa com a mudança comportamental dos profissionais envolvidos, além do comprometimento e autonomia dada aos membros da equipe. Um dos principais benefícios do uso dos MA é a construção de projeto ao redor de indivíduos motivados dando a eles o ambiente e suporte necessários e confiar que farão o seu trabalho.

Esta pesquisa colabora com a Academia ao tornar mais claro o relacionamento entre a GC e os MA em um PDS, notadamente sob a ótica da correlação e possíveis dependências entre estes temas. Isto porque nesta pesquisa verificou-se a existência de pouquíssimos estudos que correlacionam os dois construtos considerados a ponto de fundamentar uma inviabilidade da IMA. Também colabora com os profissionais e gestores de equipes de desenvolvimento de software ao sinalizar as correlações, possíveis benefícios e riscos da implantação de métodos ágeis em empresas que não tenham um sistema de GC estruturado.

Uma das limitações encontradas nesta pesquisa foi a ausência de especificação de requisitos nas documentações dos MA analisados. Também se indica como limitação, a não execução de estudo de caso aplicado em times ou empresas de desenvolvimento de software, do modo a permitir a comparação da IMA em empresa com GC estruturada, em relação à implantação de IMA em empresa sem GC estruturada.

Futuras pesquisas poderiam desenvolver estudos em campo para que se comprove ou não, através de análise qualitativa e quantitativa, uma melhor fluidez do PDS em empresas nas quais a IMA foi realizada somente após a empresa atingir certo nível de estruturação do sistema de GC. Também se indica a possibilidade de considerar outros MA, além de Scrum e Kanban.

## REFERÊNCIAS

ADNAN, M., AFZAL, M. Ontology based multiagent effort estimation system for scrum agile method. **IEEE Access**, v. 5, p. 25993-26005, 2017.

AGILE ALLIANCE. **Agile 101.** 2020. Disponível em: https://www.agilealliance.org/agile101/agile-glossary. Acesso em: 20 maio. 2020.

AGILE REPORT. **14th Annual State of Agile Report**. 2020. Disponível em: https://explore.digital.ai/state-of-agile/14th-annual-state-of-agile-report. Acesso em: 18 jun. 2020.

AHMED, Z., MANSOR, Z., AHMAD, K. An analysis of knowledge management challenges in agile global software development. **Journal of Telecommunication, Electronic and Computer Engineering**, v. 9, n. 3-4 special issue, p. 63-66, 2017.

CHONG, S. C.; CHOI, Y. S. Critical factors in the successful implementation of knowledge management. **Journal of Knowledge Management Practice**, v. 6, n. 3, 2005.

COCKBURN, A. HIGHSMIYH, J. Agile software development: The business of innovation. **The Computer Journal IEEE Computer**, v. 34, p. 120-122, 2001.

COMPUTERWORLD. **Retenção de talentos é um dos maiores desafios da TI em 2019**. 2019. Disponível em: https://computerworld.com.br/2019/03/13/retencao-de-talentos-e-um-dos-maiores-desafios-da-ti-em-2019. Acesso em: 25 mai. 2020.

DINGSØYR, T., MOE, N.B., FÆGRI, T.E., SEIM, E.A. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. **Empirical Software Engineering**, v. 23, n. 1, p. 490-520, 2018.

DRUCKER, P. O advento da nova organização. In: Harvard Business Review. **Gestão do conhecimento**. 2 ed. Rio de Janeiro: Campus, 2001.

FERNANDES, A. A.; DINIZ, J. L.; ABREU, V. F.; RICCOTTA, R.; COSTA, I. **Governança Digital 4.0.** Rio de Janeiro: Brasport, 2019.

GASPAR, M. A.; SANTOS, S. A. dos.; KUNIYOSHI, M. S.; DONAIRE, D.; PREARO, L. C.; MAGALHÃES, F. L. F. de. Gestão do conhecimento em empresas atuantes na indústria de Software no brasil: Um estudo das práticas e ferramentas utilizadas. **Inf. & Soc.:** Est., v. 26, n. 1, p. 151-166, 2016.

GOMES, A. **Métodos ágeis para desenvolvimento de software**. Porto Alegre: Bookman, 2014.

GREGORY, P., BARROCA, L., SHARP, H., DESHPANDE, A., TAYLOR, K. The challenges that challenge: Engaging with agile practitioners' concerns. **Information and Software Technology**, v. 77, p. 92-104, 2016.

INAYAT, I.; MARCZAK, S.; SALIM, S.; DAMIAN, D. Patterns of collaboration driven by requirements in agile software development teams findings from a multiple case study. **Requirements Engineering: Foundation for Software Quality**, REFSQ 2017, v. 10153, p. 131-147, DOI: 10.1007/978-3-319-54045-0_10, 2017.

ISO. **ISO 30401:2018**. 2018. Disponível em: https://www.iso.org/obp/ui/#iso:std:iso:30401:ed-1:v1:en. Acesso em 22 maio. 2020.

KHALIL, C., KHALIL, S. Exploring knowledge management in agile software development organizations. **International Entrepreneurship and Management Journal**, v. 16, n. 2, p. 555-569, 2020.

LARMAN, C. Agile and iterative development: A manager's guide. 1st ed. **Agile Software Development Series**, 2004.

MANIFESTO. **Manifesto for agile software development**. 2001. Disponível em: http://agilemanifesto.org/. Acesso em 31 mai. 2020.

MONDEN, Y. **Toyota production system**: An integrated approach to just-in-time. 4th ed. Flórida: CRC Press, 2012.

MUNOZ, O.; OKTABA, Hanna. **Especialización de MoProSoft basada en el método ágil Scrum**. Madrid: Editorial Académica Española, 2011.

NONAKA, I.; TAKEUCHI, H. **Gestão do conhecimento**. Porto Alegre: Bookman, 2008.

OHNO, T. **Toyota production system**: Beyond large-scale production. Flórida, CRC, 2019.

ROGERS, D. **Transformação digital**: repensando o seu negócio para a era digital. São Paulo: Grupo Autêntica. 2017.

ROLLAND, K. Scaling across knowledge boundaries: A case study of a large-scale agile software development project. XP '16 Scientific Workshop, 2016. **Proceedings**... XP, 2016, p. 1-5. Disponível em: https://doi-org.ez345.periodicos.capes.gov.br/10.1145/2962695.2962700. Acesso em: 15 jun. 2020.

RUS, I.; LINDVALL, M. Knowledge management in software engineering. **IEEE Software**, v. 19, n. 3, p. 26-38, 2002.

SABBAGH, R. **Scrum- Gestão ágil para projetos de sucesso**. São Paulo: Casa do Código, 2013.

SCATOLINO, A; CAMILO, R. Influence of agile methods application and knowledge management in software quality: a multivariate analysis. **Revista de Gestão e Projetos**. v. 10, n. 3, p. 65-80, 2019.

SCRUM GUIDES. **Scrum Guide 2017**. 2017. Disponível em: https://www.scrumguides.org. Acesso em: 20 maio. 2020.

SHAMEEM, M.; KUMAR, R.; NADEEM, M.; KHAN, A. Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. **Applied Soft Computing**, v. 90, 2020. DOI: 10.1016/j.asoc.2020.106122, 2020.

SOMMERVILLE, IAN. **Engenharia de software**. 10 ed. São Paulo: Pearson Education do Brasil, 2018.

TENÓRIO, N.; PINTO, D.; SILVA, J.; DE ALMEIDA, I.; BORTOLOZZI, F. Knowledge management in the software industry: how Scrum activities support a knowledge management cycle. **Navus-Revista de Gestão e Tecnologia**, v. 10, p. 01-13. 2020.

TREVISAN, L. **Fatores críticos de sucesso relacionados à gestão do conhecimento**: um estudo em organização de desenvolvimento de software. Dissertação (Mestrado). Marília, 2019.

VOIGT, S. A method for documenting agile software projects. European Conference on Knowledge Management, 18th, 2017. **Proceedings…** 2017, p. 1035-1044.

Recebido em/Received: 21/01/2021 | Aprovado em/Approved: 21/02/2021