



**VII SINGEP**

Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade  
International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317-8302

## **MELHORES PRÁTICAS EM MÉTODOS ÁGEIS APLICADOS NO SUPORTE E MANUTENÇÃO DE SOFTWARE: UMA REVISÃO SISTEMÁTICA DA LITERATURA**

**GUILHERME HENRIQUE MIRANDA**

Instituto de Gestão em Tecnologia da Informação - IGTI

**EDER JUNIOR ALVES**

UFMG - CEPEAD



## MELHORES PRÁTICAS EM MÉTODOS ÁGEIS APLICADOS NO SUPORTE E MANUTENÇÃO DE *SOFTWARE*: UMA REVISÃO SISTEMÁTICA DA LITERATURA

### Resumo

No ciclo de vida de um *software* a fase de suporte e manutenção compreende o maior período de tempo, além de ser a que apresenta o maior custo, em muitos casos. Não obstante, é comum que a tarefa de manter o *software* após sua entrega não seja devidamente organizada. As metodologias ágeis, aplicadas com sucesso em cenários complexos de desenvolvimento de *software*, seguem pouco utilizadas em equipes de suporte, nos ambientes de manutenção. O objetivo é desenvolver um *framework* para adoção e implementação de métodos ágeis em suporte e manutenção de *software*, elencando boas práticas a serem adotadas por equipes desta natureza. O método de pesquisa foi a Revisão Sistemática de Literatura (RSL). Criou-se um protocolo de revisão para coletar os documentos nas bases escolhidas. Os resultados apontam a combinação entre *Scrum* e *Kanban* como uma solução viável para executar com agilidade processos de suporte e manutenção de *software*. Com a lista de melhores práticas obtidas nos estudos analisados foi elaborado o *framework* alvo desta pesquisa. Conclui-se que a adoção de métodos ágeis nesses ambientes é benéfica para as equipes e seus clientes e isso pode ser feito aplicando-se de forma estruturada algumas práticas ágeis de sucesso encontradas na literatura.

**Palavras-chave:** Métodos ágeis; Manutenção e suporte para *software*; *Kanban*; *Scrum*; Revisão Sistemática da Literatura.

### Abstract

In a software life cycle, the support and maintenance phase represents the longest period and, in many cases, the highest costs, too. Notwithstanding, it is common that the tasks related to software maintenance after its release are not properly organized. The agile methods that are applied with high success rates in complex software development scenarios are still misused in technical support teams in maintenance environments. The objective is to develop a framework for agile methods adoption and implementation inside software support and maintenance context, listing best practices to be adopted by the teams. The Systematic Literature Review (SLR) was used as research method. A review protocol was created to collect the documents in the selected databases. The results point out that the combination of Scrum and Kanban is a viable solution for agile execution of software support and maintenance processes. The framework was created based on the complete list of best practices got from the analysed primary studies. The conclusion is that adopting agile methods in support and maintenance environments is beneficial for both, team and customer, and it can be done by applying some agile best practice that can be found in the literature in a structured way.

**Keywords:** Agile methods; Software support and maintenance; Kanban; Scrum; Systematic Literature Review.



## 1 Introdução

O contexto desta pesquisa é o ambiente bibliométrico definido para documentos que possuem na sua temática adoção de métodos ágeis em atividades inter-relacionadas para resolver os problemas de suporte e manutenção. Trata-se de uma meta-análise para avaliar esta problemática relacionada a ambientes complexos de desenvolvimento de *software* buscando benefícios do seu uso e práticas de sucesso para enfrentar os desafios inerentes.

Na comparação com as metodologias clássicas de gerenciamento de projetos de *software*, em particular o modelo *waterfall* (cascata), os métodos ágeis têm demonstrado resultados mais satisfatórios no tocante ao fator de sucesso, em especial em projetos de médio e grande portes (Hastie & Wojewoda, 2015).

Estes levantamentos corroboram a teoria de que as metodologias ágeis apresentam vantagens quando aplicadas no contexto de projetos de desenvolvimento de *software*. Entretanto, boa parte do ciclo de vida do *software* acontece na fase de suporte e manutenção, que se inicia com a entrega do produto ao cliente final (Tarwani & Chug, 2016).

Neste contexto, o problema desta pesquisa relaciona-se com a falta de critérios e definição dos possíveis benefícios que a aplicação de metodologias ágeis pode oferecer, com igual eficiência, para o atendimento de demandas pós-projetos. As empresas e suas equipes que escolhem adotar um método ágil na esfera do suporte e manutenção se deparam com alguns desafios, dada a variedade de possibilidades de implementação, acompanhada da ausência de foco nas questões relacionadas à manutenibilidade que tais métodos apresentam.

A presente pesquisa se justifica porque busca analisar as formas de superar estes problemas com falta de critérios, variedade de possibilidades e ausência de foco, por meio de uma Revisão Sistemática de Literatura (RSL). A revisão de literatura de Tarwani e Chug (2016) analisou os métodos ágeis na perspectiva do *eXtreme Programming* (XP), avaliando este método como opção para ser utilizado no contexto de suporte e manutenção. Assim, abre-se a possibilidade para estudos com escopo fechado nos cenários de suporte e manutenção, avaliando outros métodos ágeis candidatos existentes que possam contribuir com técnicas e ferramentas para o gerenciamento da área de suporte e manutenção de sistemas.

Nesta ótica, o objetivo é desenvolver um *framework* para adoção e implementação de métodos ágeis no ambiente de suporte e manutenção de *software* em produção, elencando as melhores práticas na adoção destes métodos para equipes de suporte técnico.

Para alcançar este objetivo busca-se responder: (Q.1) Quais vantagens ou benefícios as metodologias ágeis trazem quando aplicadas em contextos de suporte e manutenção de *software*? (Q.2) Quais são as principais dificuldades ou desafios encontrados na adoção destas metodologias nestes contextos? (Q.3) Existe alguma metodologia ágil pura ou híbrida que mostrou ser eficaz? (Q.4) É possível elencar um conjunto de melhores práticas a serem aplicadas nestes casos?

Nesta seção 1 o problema da pesquisa foi contextualizado, fornecendo o objetivo e as questões relacionadas. A seção 2 apresenta a análise dos conceitos e princípios relacionados ao tema. A seção 3 apresenta os procedimentos metodológicos efetuados para obter os resultados que são mostrados e analisados na seção 4. Por fim, a seção 5 fornece as considerações finais, limitações e possibilidades de pesquisas futuras.

## 2 Referencial Teórico

Desde a publicação do Manifesto Ágil, no início dos anos 2000, organizações buscam adotar métodos ágeis no desenvolvimento e gerenciamento de projetos no segmento de Tecnologia da Informação. Os resultados esperados são aumento na qualidade de seus produtos e da velocidade na entrega para os clientes (Rigby, Sutherland & Takeuchi, 2016).



Os valores ágeis, listados no Manifesto, pregam maior interação entre as pessoas e colaboração com os clientes, bem como uma resposta rápida e efetiva às mudanças (Beck et al., 2001). Esses itens não existem apenas no desenvolvimento de projetos, mas também no suporte e na manutenção de sistemas, por isso é esperado que o uso de metodologias ágeis traga, igualmente, benefícios para equipes desta natureza.

Prochazka (2011) entende que o suporte e a manutenção tiveram uma abordagem evolutiva distinta dos métodos ágeis, questionando o porquê de não se considerar as lições aprendidas do desenvolvimento ágil, agregando novas teorias, técnicas e ferramentas ao suporte e aos sistemas de manutenção (Poole & Huisman, 2001).

Manutenção é definida como o processo de modificação de um *software* após sua entrega para corrigir falhas, melhorar desempenho e fazer adaptações em um ambiente alterado (Heeager & Rose, 2014; Ahmad, Kujava, Markkula & Oivo 2016). A manutenção pode ser adaptativa, perfectiva, corretiva ou preventiva. É incorreto entendê-la como correção de bugs, manutenção corretiva, uma vez que 75% das manutenções são adaptativas ou perfectivas. Suporte é definido como um conjunto de atividades, dentro dos processos de Incidentes, Eventos e Problemas, integradas em uma função chamada *Service Desk*, segundo a biblioteca *Information Technology Infrastructure Library* (ITIL) (Prochazka, 2011).

A fase de suporte e manutenção é uma das mais longas no ciclo de vida de um *software*, representando em muitos casos mais da metade do orçamento total. Assim, uma abordagem inadequada pode resultar no aumento dos custos, além de piorar a qualidade do produto e a motivação da equipe responsável (Prochazka, 2011). Nesta linha de raciocínio, este é um bom argumento para adoção de métodos ágeis neste ambiente de adaptabilidade e aperfeiçoamento típico da manutenção (Heeager & Rose, 2014).

Métodos ágeis, como o *Scrum*, sugerem o desenvolvimento iterativo e incremental de *software*, indicando uso de iterações de tempo definido e planejadas com antecedência (Schwaber & Sutherland, 2017). Observa-se uma divergência na aplicação desta abordagem em equipes de suporte e manutenção dado que suas demandas, em geral, são imprevisíveis.

De acordo com Alves, Gonçalves e Bax (2017), o método ágil *Scrum* merece atenção por parte das empresas devido a sua prevalência na literatura. Entretanto, nota-se que no contexto do suporte e manutenção para *software* em produção, algumas técnicas não são aderentes devido a imprevisibilidade e urgência. Segundo Ferrão e Canedo (2015), isto pode ser verificado ao se tentar introduzir as *sprints* (ciclos curtos) definidas no *Scrum*.

Uma *sprint* se inicia com a reunião de planejamento, onde a equipe de desenvolvimento se compromete com as entregas de um ciclo de trabalho curto e de escopo sem alteração (Schwaber & Sutherland, 2017). Para obter o máximo de benefícios em um projeto *Scrum* é recomendado manter o tempo de duração das *sprints* entre uma e quatro semanas. Com este tipo de restrição, é necessário fazer adaptações, resultando no uso de uma estrutura híbrida composta por técnicas e ferramentas advindas de diferentes métodos ágeis.

Novos métodos para desenvolvimento surgiram com o passar do tempo, buscando minimizar os problemas dos métodos tradicionais (Prochazka, 2011). Alguns autores separam estes métodos recentes em dois tipos, ágeis e *lean* (manufatura enxuta). Segundo Conboy (2009), ser *lean* significa obter bons resultados em coisas que você pode controlar, enquanto ser ágil significa obter bons resultados em coisas que você não pode controlar. O *lean* tem foco na redução e eliminação de desperdício.

O *kanban* ou programação puxada é uma ferramenta utilizada nos sistemas produtivos, dentro da manufatura enxuta, e, primordialmente, tem como base o sistema de produção *just-in-time*. Nesta técnica, a linha de produção é puxada pela demanda do cliente, adentrando nas células de trabalho. David Anderson foi quem primeiro introduziu o termo *kanban* ao desenvolvimento de sistemas, nos anos 2000, e para diferenciar os termos, ele passou a chamá-lo de *Kanban*, com a inicial maiúscula (Al-Baik & Miller, 2014).



O *framework Kanban* é menos prescritivo que outros métodos ágeis, como *Scrum* e *XP*, por exemplo. Em outras palavras, ele diz menos o que deve ou não deve ser feito. E como ele não indica o uso de iterações, é esperado que a aplicação da abordagem *lean* associada à agilidade seja mais eficaz nestes ambientes (Kniberg & Skarin, 2009).

### 3 Metodologia

A taxonomia desta pesquisa baseia-se na RSL (Kitchenham & Charters, 2007). Os procedimentos metodológicos utilizados dividem-se em três fases: planejamento, condução e elaboração dos resultados da RSL. O planejamento da RSL foi iniciado com a validação da necessidade de condução de uma revisão sobre o tema, justificando sua execução como uma forma de evoluir o estudo realizado por Tarwani e Chug (2016). As especificações das questões de pesquisa adotam o critério PICOC, sendo elaboradas em função da população, intervenção, comparação, resultados e contexto (Petticrew & Roberts, 2006).

A população deste estudo é representada pelas áreas de suporte e manutenção de *software* em que pesquisas originais foram aplicadas. A intervenção define-se pelo o uso de métodos ágeis dentro desta população. A comparação faz-se presente no universo das áreas de suporte e manutenção que não utilizam um método formal ou que usam métodos tradicionais. Os resultados esperados são as listas de desafios, benefícios e melhores práticas que os métodos ágeis trazem quando aplicados nestes contextos, para elaboração do *framework*. O contexto aceita estudos realizados em empresas, sem restrições, e, oportunamente, na academia, em aplicações práticas ou teóricas (Kitchenham & Charters, 2007).

A etapa de condução da RSL foi executada com a aplicação da estratégia de pesquisa, dos procedimentos para seleção dos estudos, da avaliação de qualidade, da extração dos dados e da sintetização dos dados obtidos. Uma pesquisa prévia foi realizada para validar os termos a serem utilizados, bem como para validar as fontes de pesquisa a serem consideradas.

Kitchenham e Charters (2007) indicam a necessidade de mesclar o uso de fontes digitais com fontes de arquivos físicos. No entanto, apenas conteúdo digital foi extraído para compor a base de documentos. Isso devido ao grande acervo disponível na *web*, com revistas e periódicos importantes nas áreas de Engenharia de *Software* e Ciência da Computação.

O protocolo da RSL teve como fonte as bases *ACM Digital Library*, *EBSCO*, *IEEE Explore*, *ScienceDirect Elsevier* e *Springer Link*. A Tabela 1 fornece seus endereços eletrônicos na segunda coluna. Os termos ou palavras-chave estão na terceira coluna. Finalizando, a quarta e última coluna apresenta a variação temporal dos documentos.

Tabela 1:

**Relação de bancos de dados e termos pesquisados em 19/03/2018**

Nome do banco de dados	Endereço eletrônico	Termos pesquisados	Anos cobertos
ACM Digital Library	<a href="https://dl.acm.org">https://dl.acm.org</a>	(agile OR scrum OR kanban OR xp OR lean) AND (software OR system) AND (maintenance OR maintainability OR evolution OR support)	2008-2018
EBSCO	<a href="https://www.ebsco.com/e/pt-br">https://www.ebsco.com/e/pt-br</a>		
IEEE Explore	<a href="http://ieeexplore.ieee.org/Xplore/home.jsp?reload=true">http://ieeexplore.ieee.org/Xplore/home.jsp?reload=true</a>		
ScienceDirect Elsevier	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>		
Springer Link	<a href="https://link.springer.com">https://link.springer.com</a>	agile AND software AND maintenance AND (scrum OR kanban OR xp OR lean OR evolution OR maintainability OR support OR system)	



A primeira parte da estrutura dos termos de pesquisa definiu a busca por metodologias ágeis, cabendo destacar que o termo *Crystal* não foi incluído na versão final da consulta por ser um termo amplamente utilizado na área de materiais, fazendo com que o número de resultados retornados aumentasse consideravelmente, com artigos irrelevantes para a pesquisa.

A segunda parte da estrutura foi utilizada para limitar trabalhos que estivessem ligados às áreas de tecnologia da informação. E a última das três partes buscava limitar o escopo da pesquisa a cenários onde suporte e manutenção de *software* estivessem presentes.

Todas as buscas foram realizadas no dia 19/03/2018. E nos casos em que era possível aplicar a pesquisa somente aos metadados dos trabalhos – título, resumo e palavras-chave – este recurso foi utilizado para minimizar resultados irrelevantes.

Esta pesquisa resultou em cerca de 1.000 registros únicos, somadas todas as fontes. Os resultados foram carregados e armazenados na ferramenta EndNote e formaram a base utilizada para aplicação dos critérios e procedimentos de seleção dos estudos.

A seleção dos estudos, seguindo os critérios de inclusão definidos no protocolo de revisão, tem como objetivo identificar os estudos primários que fornecem alguma evidência que ajude a responder às questões de pesquisa (Kitchenham & Charters, 2007). O critério de seleção contou com as seguintes prerrogativas:

- O estudo deveria estar escrito nos idiomas inglês, português ou espanhol;
- A data de publicação deveria estar compreendida entre os anos 2008 e 2018;
- O estudo deveria estar publicado em uma revista ou periódico ou fazer parte de algum congresso ou conferência ligados à área de tecnologia da informação;
- O estudo deveria apresentar uma abordagem de uso de metodologia ágil;
- O estudo deveria apresentar uma aplicação de tal metodologia dentro do contexto de suporte e/ou manutenção, podendo ser aplicado na indústria ou na academia.

Na primeira rodada de avaliação, o título, o nome do periódico de publicação e as palavras-chave foram lidos para verificar se cada registro encontrado poderia ajudar a responder às questões de pesquisa. Resultaram daí 78 estudos incluídos.

Na segunda rodada de avaliação, o resumo e a conclusão foram lidos para confirmar se os critérios de inclusão estavam sendo respeitados, em especial as duas últimas prerrogativas listadas acima. Nesta etapa, os estudos que apresentaram alguma incerteza foram também incluídos, assim, o total de estudos aceitos passou para 26.

Estes 26 estudos selecionados passaram para a fase de avaliação de qualidade, que não contou com um critério formal por três razões. Primeira, parte da qualidade dos estudos ficou garantida na etapa de condução da busca, pois apenas fontes relevantes no meio acadêmico foram acessadas. Segunda, o número de publicações a respeito do tema da pesquisa ainda é baixo, de modo que pequenas contribuições são relevantes para os resultados. Terceira, as questões não aceitam apenas uma resposta correta, mas sim uma lista de possibilidades a ser construída utilizando-se a coincidência dos dados encontrados nos estudos avaliados.

Os 26 estudos que chegaram até a etapa de avaliação da qualidade tiveram seu texto lido de forma integral. Após esta leitura, 17 estudos foram excluídos, por não apresentarem um cenário de aplicação de metodologias ágeis em ambiente de suporte e manutenção, o que não havia ficado evidente com a leitura do resumo e da conclusão.

Para a extração dos dados dos estudos primários foi implementado um projeto no *software* de automação da pesquisa qualitativa NVivo versão 12. Foram criadas quatro categorias principais, para codificar os trechos de texto com indicações dos seguintes aspectos: metodologia ágil adotada na pesquisa, benefícios obtidos com sua aplicação no cenário de suporte e manutenção de *software*, desafios encontrados durante sua introdução e práticas aplicadas que tiveram resultados positivos na visão dos autores.



Uma vez que a definição da estratégia de extração dos dados já era conhecida e estava presente no protocolo de revisão, foi possível realizar a etapa de codificação enquanto os estudos primários eram lidos. Os estudos que não apresentaram qualquer codificação relevante foram excluídos. Com essa leitura integral, 9 estudos foram devidamente codificados e tiveram seus dados coletados levados para a etapa de síntese dos dados.

Os dados sintetizados estão na forma qualitativa, com um complemento quantitativo na apresentação dos conceitos avaliados, contando quantas vezes um determinado conceito se repetiu entre os estudos. O processo consistiu em analisar e agrupar logicamente, em subcategorias, todas as codificações nos estudos primários.

Após o agrupamento, foram elaborados gráficos de barras relacionando estudos e elementos categorizados, permitindo uma indicação visual dos itens de maior recorrência. Em termos práticos, é possível assim fazer uma interpretação de quais são os benefícios mais evidentes, quais os principais pontos de atenção ou desafios mais comuns, e, dentre as práticas de sucesso, aquelas que foram mais aplicadas e trouxeram algum resultado positivo. Tal formatação auxilia a construção do *framework* de aplicação de métodos ágeis em equipes de suporte e manutenção de *software*, apresentado na seção 4.

#### 4 Análise dos Resultados

Ahmad et al. (2016), Choudhari e Suman (2015), Concas, Lunesu, Marchesi e Zhang (2013), Ferrão e Canedo (2015), Heeager e Rose (2015), Polk (2011), Prochazka (2011), Seikola, Loisa e Jagos (2011) e Terlecka (2012) são os 9 estudos primários selecionados para análise após aplicação dos critérios de inclusão e qualidade.

A Tabela 2 apresenta um resumo dos artigos segundo as seguintes classificações: tipo de publicação, ano de publicação, idioma e cenário de aplicação. Para esta última classificação, foi levado em conta o seguinte critério: estudos realizados em ambiente acadêmico foram classificados como “Academia”; estudos com uma proposta de validação dos conceitos por meio de simulações, como “Simulação Computacional”; estudos realizados em ambientes empresariais, como “Empresa”.

Tabela 2:

##### Resumo das informações gerais dos estudos primários incluídos.

Estudo Primário	Tipo de publicação	Ano de publicação	Idioma	Aplicação
Ahmad et al.	Anais de Congresso	2016	Inglês	Empresa
Choudhari e Suman	Artigo de Periódico	2015	Inglês	Academia
Concas et al.	Artigo de Periódico	2013	Inglês	Simulação Computacional
Ferrão e Canedo	Anais de Congresso	2015	Português	Empresa
Heeager e Rose	Artigo de Periódico	2015	Inglês	Empresa
Polk	Anais de Congresso	2011	Inglês	Empresa
Prochazka	Anais de Congresso	2011	Inglês	Empresa
Seikola et al.	Anais de Congresso	2011	Inglês	Empresa
Terlecka	Anais de Congresso	2012	Inglês	Empresa

Quanto aos anos de publicação, não há estudos primários relevantes para a pesquisa para os anos de 2017 e 2018, segundo critérios utilizados para busca e inclusão desta RSL. Isso se deve a alguns motivos, como o fato desta pesquisa não ter se estendido à *grey literature*, onde a publicação de material segue critérios menos rígidos, comparado às







compartilhamento de conhecimento”. Choudhari e Suman (2015), dentro do contexto do XP, destacam um maior envolvimento do cliente, que faz parte da equipe. Ahmad et al. (2016) apontam para o esforço dos membros da equipe em auxiliar uns aos outros nas situações de bloqueio, pois agora entendem melhor as necessidades de cada um, além de destacarem a troca de conhecimento que se percebe quando há membros com diferentes níveis de conhecimento dentro da equipe.

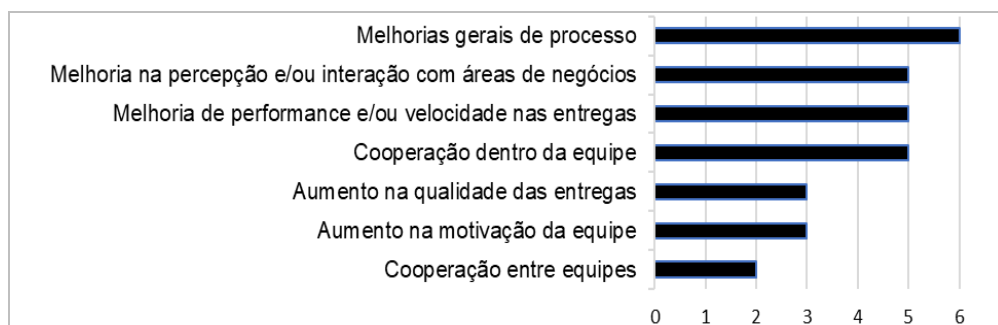
**Cooperação entre equipes:** Prochazka (2011) e Seikola et al. (2011) destacam melhoria na atuação *cross* funcional quando se trabalha com métodos ágeis.

**Melhoria de performance e/ou velocidade nas entregas:** em Polk (2011) temos a indicação de que a equipe de suporte que passou a utilizar *Kanban* apresentou melhorias consistentes nos seus tempos de ciclo, durante o período de acompanhamento. Seikola et al. (2011) indicam igualmente que a equipe de suporte passou a resolver os problemas de forma mais rápida, ao adotar uma metodologia ágil. Concas et al. (2013) identificam redução nos tempos de entrega e de espera e um aumento na capacidade da equipe em executar seu trabalho. Segundo Choudhari e Suman (2015), a aplicação de práticas do XP melhorou a produtividade da equipe de manutenção. Ferrão e Canedo (2015) destacam que a equipe passou a conseguir finalizar um número maior de atividades, quando *Scrum* e *Kanban* passaram a ser utilizados em conjunto.

**Melhoria na percepção e/ou interação com áreas de negócios:** Polk (2011) cita que um planejamento de prioridades nas tarefas permite indicar aos clientes uma data de reunião de revisão bastante confiável. Prochazka (2011) indica que o uso da abordagem ágil ajuda a atingir os objetivos das áreas negócios com as entregas. Concas et al. (2013) sugerem que o aumento na satisfação dos clientes é fruto do uso de *Lean-Kanban* na equipe de manutenção utilizada em seu estudo. Heeager e Rose (2015) destacam melhoria nas estimativas das *sprints*, o que aumenta a confiança do cliente na equipe. Choudhari e Suman (2015) indicam a melhoria no entendimento das necessidades dos clientes no levantamento de requisitos e também a possibilidade de se receber *feedbacks* instantâneos, devido às entregas contínuas.

**Melhorias gerais de processo:** Polk (2011) indica que o processo tornou a equipe mais flexível, permitindo-a manter um ritmo sustentável de trabalho, enquanto Prochazka (2011) indica uma maior mitigação de riscos em conjunto com uma diminuição na burocracia. Seikola et al. (2011) chamam a atenção para a criação de uma mentalidade de melhoria contínua e Heeager e Rose (2015) para uma maior eficiência no planejamento de reuniões, sendo que em ambos os estudos e também em Ferrão e Canedo (2015) e Ahmad et al. (2016) há um destaque para as melhorias na visibilidade geral do processo.

A Figura 2 mostra a frequência de menções de cada um dos benefícios elencados nos estudos primários avaliados.



**Figura 2:** Frequência de menções dos benefícios nos estudos primários.

Podemos notar, com base nesses dados, que o uso de métodos ágeis em equipes de suporte e manutenção auxilia na melhoria dos processos, de forma particular no aumento da



visibilidade do trabalho das equipes. Maior cooperação interna e com os clientes, bem como melhoria na performance da equipe também são frequentemente citados.

Assim, pode-se entender que a resposta para a primeira questão é afirmativa, e é possível conhecer alguns dos benefícios ou vantagens que se encontram nestes contextos.

#### 4.2 Lista de dificuldades ou desafios

**Conhecimento centralizado:** Prochazka (2011) sugere que as demandas de suporte e manutenção costumam ser recorrentes por haver pouco compartilhamento de conhecimento em equipes desta natureza. Seikola et al. (2011) elencam, entre outros desafios encontrados em suas observações, a dificuldade em manter o aprendizado contínuo. Heeager e Rose (2015) chamam a atenção para a escassez de documentação, o que dificulta a atuação em casos de suporte e manutenção devido à falta de conhecimento.

**Dificuldade em limitar trabalho em andamento:** segundo Seikola et al. (2011), quando se opta por uma metodologia que sugere a limitação do trabalho em andamento, como o *Kanban*, é desafiador acreditar no efeito positivo que há em limitar o progresso, enquanto se vê o *backlog* de solicitações de suporte crescendo.

**Dificuldades em enxergar o planejamento e/ou progresso:** Terlecka (2012) indica que a principal dificuldade com o uso do *Kanban* foi a equipe perceber o progresso de suas atividades, pela ausência de prazos de entrega. O mesmo é citado por Heeager e Rose (2015), no entanto sem citar a metodologia *Kanban* especificamente. E embora se tenha visto na relação de benefícios que há um aumento na visibilidade de curto prazo, Ahmad et al. (2016) apresentam a dificuldade que existe em se enxergar o planejamento de longo prazo.

**Dificuldades em estabelecer um time:** Polk (2011) cita que, em equipes muito grandes, algumas pessoas tendem a se desengajar quando a auto-organização se faz necessária. Dificuldades com auto-organização também são citadas por Ferrão e Canedo (2015). Seikola et al. (2011) remontam à dificuldade de fazer um grupo de indivíduos se tornar verdadeiramente um time. Heeager e Rose (2015) questionam o trabalho em equipe nos cenários de manutenção dado que em geral os trabalhos são realizados pontual e individualmente. Ahmad et al. (2016) elencam alguns cenários onde a falta de comunicação e colaboração dificultam o estabelecimento de um time.

**Falta de um cliente próximo:** Heeager e Rose (2015) indicam que o cliente raramente está disponível quando se trabalha em equipes de suporte e manutenção, uma vez que as demandas são geralmente pequenas e de clientes variados. Ahmad et al. (2016) também citam que um dos desafios é não contar com um cliente próximo.

**Imprevisibilidade e/ou alto número de urgências:** Seikola et al. (2011) cita que é comum demandas de alta prioridade chegarem, forçando a equipe a mudar as tarefas em que está trabalhando. Terlecka (2012) indica que quando se faz uso do *Scrum* em uma equipe de suporte e manutenção, ao se priorizar os itens do *sprint backlog* o cliente real não está sendo atendido. Em Heeager e Rose (2015) vemos que *sprints* com características de manutenção estão sujeitas a interrupções por demandas urgentes dos clientes com pouco espaço para elaboração de releases integradas. Ferrão e Canedo (2015) indicam que é comum a equipe interromper atividades da *sprint* para atender demandas prioritárias do cliente, gerando alto número de horas extras e desmotivação. E Ahmad et al. (2016) apontam que quando um cliente escala um problema severo não é possível seguir os planos formais da *sprint*.

A Figura 3 mostra a frequência de menções de cada um dos desafios elencados nos estudos primários avaliados.

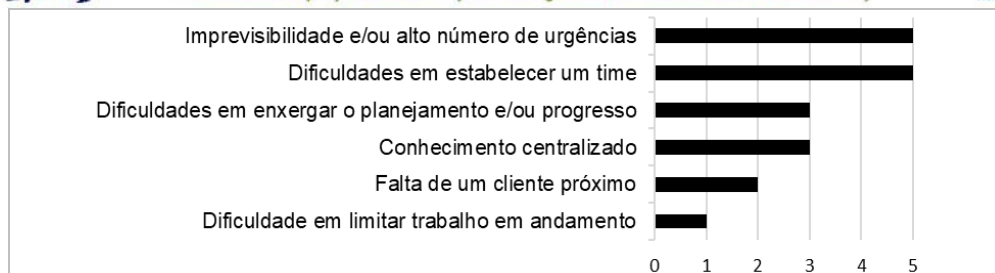


Figura 3: Frequência de menções dos desafios nos estudos primários.

É possível constatar, utilizando-se os dados aqui levantados, que as iterações de desenvolvimento de tempo fixo, como as *sprints* do *Scrum*, não são muito adequadas em equipes de suporte e manutenção, devido à imprevisibilidade das demandas. Outro desafio comumente notado é a formação de um time verdadeiramente, que vai além de um grupo de pessoas que simplesmente trabalham em conjunto.

Os resultados aqui obtidos ajudam a responder à segunda questão de pesquisa, indicando quais são as dificuldades ou desafios mais recorrentes quando se decide adotar um método ágil em um ambiente de suporte e manutenção.

### 4.3 Comparativo entre metodologias

Nenhum dos estudos avaliados apresentou uma comparação entre um conjunto de metodologias para validar qual delas se mostrava mais eficaz. A maior parte dos estudos já partia desde o início com uma sugestão de metodologia a ser aplicada e observada.

Ahmad et al. (2016) apresentam um cenário de troca do *Scrum* para o *Kanban*. Choudhari e Suman (2015) avaliam as adaptações necessárias para o uso do XP em um contexto de suporte e manutenção, enquanto Heeager e Rose (2015) fazem o mesmo com o *Scrum*. Concas et al. (2013) sugerem uma simulação computacional, comparando o desempenho teórico do *Kanban*, contra uma abordagem tradicional.

Ferrão e Canedo (2015), Seikola et al. (2011) e Terlecka (2012), sugerem a utilização de um híbrido de *Scrum* e *Kanban*, aproveitando o melhor em cada um dos *frameworks*. Polk (2011) contempla um cenário de uso direto do *Kanban* na equipe de suporte e manutenção. Prochazka (2011) é o único que não cita explicitamente a metodologia aplicada, uma vez que as menções à palavra *kanban* são em letras minúsculas. Neste estudo é apresentada uma série de práticas e técnicas indicadas genericamente como ágeis ou *lean*.

Para efeitos comparativos, as metodologias foram divididas em três categorias: as que apresentam o uso de uma metodologia ágil apenas (*Scrum*, XP); as que apresentam uso de ágil com abordagem *lean* (*Kanban*); as que apresentam um híbrido entre as outras duas. A Figura 4 representa graficamente estes resultados.

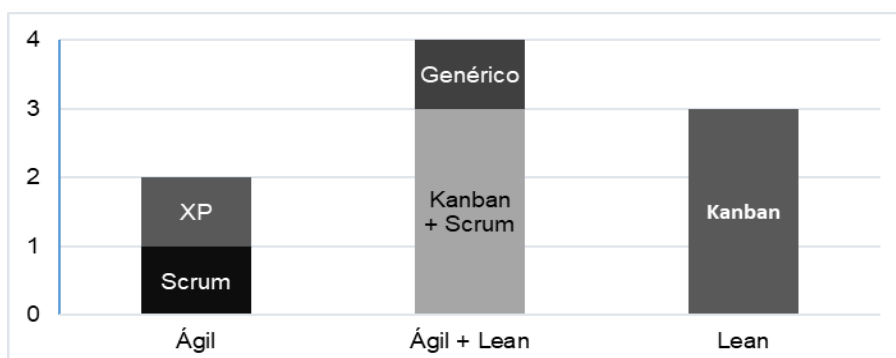


Figura 4: Identificação das metodologias apresentadas por tipo de abordagem



Desta forma, não conseguimos encontrar uma resposta única para a terceira questão, temos apenas uma indicação de que a metodologia *Kanban* e o uso de um híbrido entre ágil puro e ágil com abordagem *lean*, como *Kanban* e *Scrum*, vêm sendo bastante utilizados nas pesquisas referentes à aplicação de metodologias ágeis em ambientes de suporte e manutenção. Isso já foi observado anteriormente, na análise da nuvem de palavras, onde é possível notar uma frequência maior do termo *Kanban* quando comparado ao *Scrum*.

#### 4.4 Lista de melhores práticas

Temos abaixo um conjunto de 17 práticas mais comuns entre os estudos. Houve mais práticas codificadas, porém estão listadas somente aquelas citadas em ao menos três estudos.

**Compartilhar conhecimento:** Prochazka (2011) sugere que deve haver oportunidade para aprendizagem no trabalho, para se adquirir conhecimento e experiência. Heeager e Rose (2015) citam que para haver um maior nivelamento de conhecimento entre os especialistas da equipe foi necessário um grande foco em transferência de conhecimento. Choudhari e Suman (2015) indicam que o uso de métodos ágeis ajuda na aquisição de conhecimento e troca de experiências, minimizando problemas com a troca de membros das equipes de suporte. E Ahmad et al. (2016) mencionam o fato de que os gerentes movimentam os membros com maior conhecimento entre diferentes equipes para que haja este compartilhamento.

**Compartilhar responsabilidades:** Polk (2011) aponta que a equipe compartilha com o *Product Owner* (P.O.) a responsabilidade na criação de *user stories* e na realização de estimativas, enquanto Terlecka (2012) indica que, ao perceber que não havia um P.O., a própria equipe assumiu as responsabilidades que caberiam a esse papel. Heeager e Rose (2015) chamam a atenção para a propriedade compartilhada pelas tarefas, favorecendo a cooperação dentro da equipe. E Choudhari e Suman (2015) citam a responsabilidade coletiva pelo código e sua qualidade.

**Conduzir reunião de planejamento:** Polk (2011) sugere reuniões de planejamento envolvendo todas as equipes que estejam envolvidas no processo. Terlecka (2012), por sua vez, sugere uma reunião de planejamento diária, misturando elementos das reuniões diárias e das reuniões de planejamento indicadas no *Scrum*, enquanto Concas et al. (2013) sugerem reuniões semanais conduzidas em conjunto com responsáveis pela área de negócios. Heeager e Rose (2015) propuseram mudanças na execução da reunião de planejamento, por entenderem que se trata de um artefato importante para o processo. Choudhari e Suman (2015) indicam que no cenário de manutenção com o uso de XP o planejamento deve ser feito em dois níveis: o planejamento da iteração e o da entrega.

**Conduzir reunião de retrospectiva:** Prochazka (2011), Seikola et al. (2011), Terlecka (2012), Ferrão e Canedo (2015) e Ahmad et al. (2016) reconhecem a importância das retrospectivas para que a equipe possa reunir lições aprendidas e agir sobre seus pontos fracos. Mesmo em cenários híbridos entre *Scrum* e *Kanban* ou com uso de *Kanban* somente, esta prática é sugerida nos estudos primários supracitados.

**Conduzir reunião diária:** com exceção de Concas et al. (2013), que apresenta uma simulação computacional, os demais estudos indicaram a necessidade de se conduzir uma reunião diária. Polk (2011) sugere reuniões de 15 minutos, assim como Prochazka (2011) que indica que a reunião mantém a equipe sincronizada. Seikola et al. (2011) e Ahmad et al. (2016) citam que esta cerimônia realça o trabalho em equipe. Terlecka (2012) sugere reuniões de 30 minutos, para que possa incluir também o planejamento das próximas 24 horas. Heeager e Rose (2015) citam que a reunião diária é uma das bases para aprendizado da equipe. Choudhari e Suman (2015) sugerem que a reunião seja feita em frente ao quadro. Ferrão e Canedo (2015) a enxergam como uma oportunidade de mapear o fluxo de trabalho.



**Destacar visualmente determinadas tarefas no quadro:** Polk (2011) sugere que fiquem destacadas as 5 tarefas de maior prioridade. Seikola et al. (2011) e Ferrão e Canedo (2015) utilizam cores diferentes para indicar tipos de atividades ou severidades. Terlecka (2012) indica uma zona de espera, para tarefas bloqueadas por interferência externa.

**Limitar o trabalho em andamento:** Polk (2011) indica que com a limitação do trabalho em andamento ficou mais fácil manter o trabalho fluindo em ritmo cadenciado. Seikola et al. (2011) e Terlecka (2012) citam que esta limitação foi aplicada com bons resultados nas equipes de suporte e manutenção, deixando os times decidirem os próprios limites. Já Ferrão e Canedo (2015) sugerem que, inicialmente, o limite deve ser o mínimo possível, para que possa ser ajustado posteriormente. Concas et al. (2013), em sua simulação computacional, aplicam a limitação do trabalho em andamento e os resultados são melhores quando comparados com a abordagem sem essa limitação. Ahmad et al. (2016), indica também o uso deste limite, por ser um dos pilares do *Kanban*.

**Manter um backlog:** Polk (2011) e Choudhari e Suman (2015) mencionam que as equipes utilizam *User Stories* como itens de trabalho. Terlecka (2012) cita a existência de um *Product Backlog*, com a lista de itens a serem trabalhados. Concas et al. (2013) chamam de *backlog* o repositório de demandas criadas em sua simulação computacional. Heeager e Rose (2015) também fazem menção a um *backlog* com os itens a serem trabalhados.

**Medir performance e/ou velocidade:** Polk (2011) utiliza como padrão as medidas de tempo de ciclo, calculado em *Story Points*. Seikola et al. (2011) sugerem algumas métricas como tempo de fila, tempo de espera, percentual de atrasos, número e percentual de casos em cada uma das colunas do quadro. Heeager e Rose (2015) indicam que é recomendado trabalhar com um conjunto enxuto de métricas, incluindo aspectos como satisfação dos membros do time, cooperação, qualidade das entregas e precisão nas estimativas.

**Não realizar estimativas baseadas em unidades absolutas:** Polk (2011) sugere o uso de *Story Points*, que é uma unidade de medida relativa. Prochazka (2011) e Choudhari e Suman (2015) apontam que o uso de *Planning Poker* ajuda, inclusive, a trazer alguma diversão para o processo de estimativa. Em geral estimativas realizadas com esta técnica são baseadas em *Story Points*. Terlecka (2012) e Concas et al. (2013) sugerem que as atividades sequer precisam de uma estimativa.

**Possuir uma definição de feito (*Definition of Done*):** Seikola et al. (2011) citam que a definição de feito foi incluída em cada uma das colunas do quadro, para guiar a movimentação de itens entre elas. Terlecka (2012) já aponta o uso da definição de feito apenas para alguns tipos de tarefas. Ahmad et al. (2016) indicam que, com a definição de feito, são removidas do processo as tarefas parcialmente concluídas.

**Puxar o trabalho em vez de empurrar:** Seikola et al. (2011) citam que como qualquer membro da equipe é capaz de tratar qualquer um dos itens do *backlog*, criou-se genuinamente uma mentalidade de puxar o trabalho a ser feito. Terlecka (2012), Ferrão e Canedo (2015) e Ahmad et al. (2016) também dizem que os itens são puxados pelos membros da equipe conforme sua prioridade. O mesmo comportamento é observado em Concas et al. (2013), quando é apresentada a equipe ágil base para a elaboração do algoritmo de simulação.

**Rotacionar responsáveis pelo suporte:** Prochazka (2011) e Seikola et al. (2011) indicam que é necessário rotacionar os membros da equipe entre atividades de suporte e novos desenvolvimentos, tanto para disseminar o conhecimento, como também para melhor entendimento das necessidades dos clientes. Ahmad et al. (2016) cita que com o uso do *Kanban*, tornou-se possível rotacionar os membros das equipes de suporte.

**Trabalhar com iterações de tempo fixo:** Terlecka (2012) propõe o uso de micro *sprints* de duração de um dia. Heeager e Rose (2015) sugerem o uso de *sprints* regulares, por estarem apoiando os estudos na aplicação do *Scrum*. Choudhari e Suman (2015) sugerem ciclos de duas semanas, dentro da aplicação do XP. Ferrão e Canedo (2015), por sua vez,



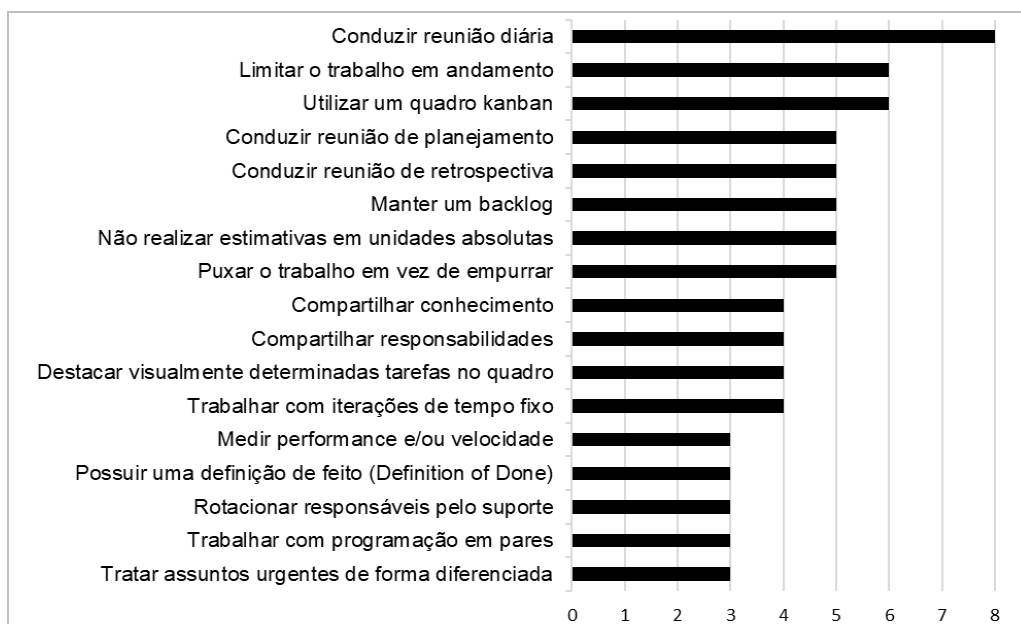
indicam que, mesmo utilizando o *Kanban*, trabalhar com tempos de ciclo limitados – um mês – é adequado para mensurar o desempenho do fluxo.

**Trabalhar com programação em pares:** Prochazka (2011) entende que o uso de trabalho em pares pode ser vantajoso para tratamento de incidentes críticos e na investigação de causa raiz de problemas. Choudhari e Suman (2015) citam que a programação em pares induz uma busca adicional de alternativas e tende a tornar as tarefas de manutenção mais agradáveis. Ahmad et al. (2016) sugerem o uso de programação em pares como opção para compartilhamento de conhecimento.

**Tratar assuntos urgentes de forma diferenciada:** Polk (2011) sugere que, havendo interrupções, esses itens devem substituir outros planejados de menor prioridade. Seikola et al. (2011) indicam colocar uma área destacada no topo do *backlog* para tratamento de assuntos urgentes. Ahmad et al. (2016) sugerem uma fila expressa para lidar com trabalhos extremamente urgentes ou de prioridade crítica.

**Utilizar um quadro *kanban*:** Polk (2011) cita que a equipe partiu de um quadro genérico de *stories* para um quadro *kanban*, incluindo regras específicas para cada uma das colunas. Prochazka (2011) comenta que o uso de um painel *kanban* é importante para a visualização. Seikola et al. (2011), Ferrão e Canedo (2015) e Ahmad et al. (2016) indicam que as equipes acompanhadas usavam um quadro *kanban*, enquanto Terlecka (2012) faz menção apenas ao uso de um quadro físico, porém como existe uma limitação da quantidade de tarefas em cada coluna, entende-se se tratar de um quadro *kanban*.

A Figura 5 mostra a frequência de menções de cada uma das melhores práticas elencadas nos estudos primários avaliados.



**Figura 5:** Frequência de menções das melhores práticas nos estudos primários.

O resultado aqui encontrado está de acordo com aquele obtido para a questão Q.3, que remete ao uso de um híbrido entre *Kanban* e *Scrum*. Isso porque a reunião diária e a reunião de planejamento são práticas sugeridas pelo *Scrum*, enquanto a limitação do trabalho em andamento e a utilização de um quadro *kanban* são sugeridas pelo *Kanban*. E assim é apresentada uma resposta à questão Q.4, com uma relação de práticas de sucesso observadas nos estudos primários aqui avaliados, dando origem ao *framework* proposto na próxima seção.



#### 4.5 Framework para aplicação de métodos ágeis em equipes de suporte e manutenção

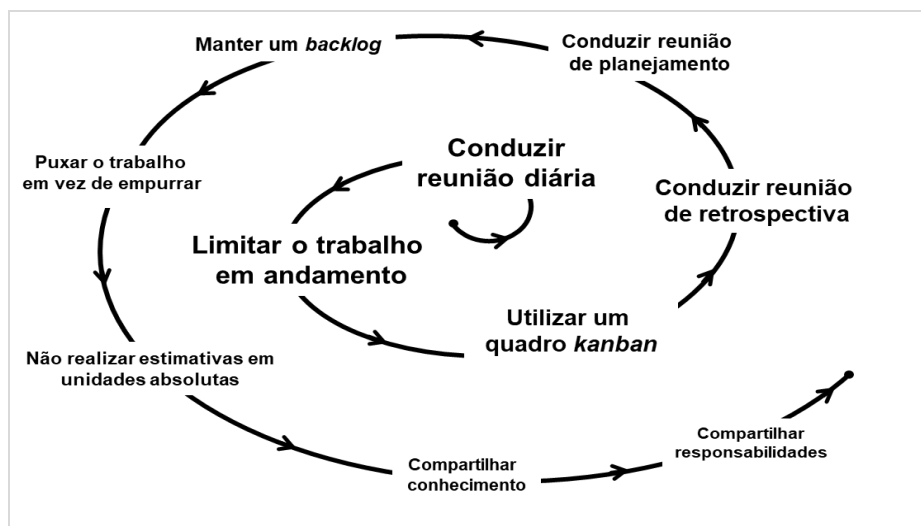
Respondidas as questões de pesquisa, é possível propor um *framework* para aplicação de métodos ágeis em equipes de suporte e manutenção. Este *framework* é um híbrido entre *Kanban* e *Scrum*, buscando utilizar o que cada uma das metodologias apresenta de melhor, segundo o que foi levantado com a realização desta RSL. Dois dos princípios básicos do *Kanban*, segundo Kniberg e Skarin (2009), compõem a base deste *framework*:

- visualização do fluxo de trabalho, com a prática “Utilizar um quadro *kanban*”
- limitação do trabalho em andamento, com “Limitar o trabalho em andamento”

E os eventos de inspeção e adaptação, aspectos chave do *Scrum*, conforme Schwaber e Sutherland (2017), os complementam:

- melhorar comunicação, identificar impedimentos, tomar decisões de forma rápida, com a prática “Conduzir reunião diária”
- identificar pontos de melhoria e criar um plano para colocar ações de melhoria em prática, com “Conduzir reunião de retrospectiva”

Essas são as práticas fundamentais deste *framework*, que se complementam com as demais, apresentadas na Figura 6.



**Figura 6:** Práticas essenciais do *framework* de agilidade em equipes de suporte e manutenção

O *framework* proposto pode ser definido como a espiral da agilidade para equipes de suporte e manutenção. As práticas nucleares são as mais fortemente indicadas, segundo resultados obtidos na pesquisa. À medida que se avança no braço da espiral encontram-se práticas também importantes, porém que apresentaram menor prevalência na literatura.

#### 5 Considerações Finais

Com a execução da Revisão Sistemática da Literatura foi possível elaborar uma proposta de *framework* para aplicação e implementação de métodos ágeis no ambiente de suporte e manutenção de *software*, elencando melhores práticas na adoção destes métodos para equipes de suporte técnico e comunidade acadêmica interessada na temática.

Práticas sugeridas pelo *Scrum*, como realizar reuniões diárias e conduzir reuniões de retrospectiva regularmente, se somam a práticas sugeridas pelo *Kanban*, como visualizar o fluxo de trabalho em um quadro e limitar a quantidade de trabalho em andamento, para formar a estrutura principal do *framework* aqui proposto. Isso vem de encontro ao resultado



encontrado nos estudos analisados, que acenam com o uso de um híbrido entre *Scrum* e *Kanban* como sendo a melhor alternativa para tratamento de demandas de suporte e manutenção dentro de equipes que trabalham com métodos ágeis.

A aplicação de agilidade em equipes de suporte e manutenção reflete, como benefícios, uma melhoria geral do processo de trabalho, melhoria na interação com as áreas usuárias, maior velocidade nas entregas e maior cooperação dentro da equipe. Por sua vez, as principais dificuldades na adoção são a imprevisibilidade das demandas, que afeta o planejamento das equipes, o estabelecimento de um time, em vez de um grupo de pessoas apenas, a visualização do plano de médio e longo prazos e a centralização do conhecimento.

De toda forma, o presente estudo traz algumas limitações, como os idiomas e o período dos artigos utilizados na condução da RSL, bem como a seleção de estudos que não se estendeu a materiais impressos e nem à *grey literature*, que inclui trabalhos de origem não acadêmica. Esta pesquisa pode ser ampliada e novos resultados podem corroborar ou complementar aqueles obtidos no presente artigo.

Futuramente, em continuidade aos tópicos aqui expostos, pode-se conduzir um estudo de caso com uma equipe de suporte e manutenção que venha a utilizar metodologias ágeis, para validar os resultados dessa pesquisa. Uma aplicação prática deste *framework* ajudaria a contemplar situações não discutidas nos estudos primários, gerando valores adicionais à lista de práticas essenciais aqui elencadas.

### Referências

- Ahmad, M. O., Kuvaja, P., Oivo, M., & Markkula, J. (2016). Transition of software maintenance teams from Scrum to Kanban. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016–Março (i), 5427–5436. <https://doi.org/10.1109/HICSS.2016.670>.
- Al-Baik, O., & Miller, J. (2014). The Kanban approach, between agility and leanness: a Systematic Review. *Empirical Software Engineering*, 20(6), 1861–1897. <https://doi.org/10.1007/s10664-014-9340-x>.
- Alves, E. J., Gonçalves, C. A., & Bax, M. P. (2017). Métodos ágeis sob a ótica da Informação. *Informação & Informação*, 22(3), 178. <https://doi.org/10.5433/1981-8920.2017v22n3p178>
- Beck, K., Beedle, M., Bennekum, A. van, Cockburn, A., Cunningham, W., Fowler, M., ... Thomas, D. (2001). Manifesto para desenvolvimento ágil de software. Recuperado em Novembro 22, 2017, de <http://agilemanifesto.org/iso/ptbr/manifesto.html>.
- Choudhari, J., & Suman, U. (2015). An empirical evaluation of iterative maintenance life cycle using XP. *ACM SIGSOFT Software Engineering Notes*, 40(2), 1–14. <https://doi.org/10.1145/2735399.2735406>.
- Conboy, K. (2009). Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*, 20(3), 329–354. <https://doi.org/10.1287/isre.1090.0236>.
- Concas, G., Lunesu, M. I., Marchesi, M., & Zhang, H. (2013). Simulation of software maintenance process, with and without a work-in-process limit. *Journal of Software: Evolution and Process*, 25(12), 1225–1248. <https://doi.org/10.1002/smr.1599>.





- Ferrão, S. É. R., & Canedo, E. D. (2015). A study of the applicability of an agile methodology scrum allied to the Kanban method. *2015 10th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6. <https://doi.org/10.1109/cisti.2015.7170382>.
- Hastie, S., & Wojewoda, S. (2015). *Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch*. 1–9. Recuperado de <https://www.infoq.com/articles/standish-chaos-2015>.
- Heeager, L. T., & Rose, J. (2014). Optimising agile development practices for the maintenance operation: nine heuristics. *Empirical Software Engineering*, 20(6), 1762–1784. <https://doi.org/10.1007/s10664-014-9335-7>.
- Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. *Engineering*, 45(4ve), 1051. <https://doi.org/10.1145/1134285.1134500>
- Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - making the most of both*. Work (Vol. 1). <https://doi.org/10.1017/CBO9781107415324.004>.
- Petticrew, M., & Roberts, H. (2006). *Systematic Reviews in the Social Sciences: A practical guide*. Cebma.Org. <https://doi.org/10.1027/1016-9040.11.3.244>.
- Polk, R. (2011). Agile & Kanban in coordination. *Proceedings - 2011 Agile Conference, Agile 2011*, 263–268. <https://doi.org/10.1109/AGILE.2011.10>.
- Poole, C., & Huisman, J. W. (2001). Using extreme programming in a maintenance environment. *IEEE Software*, 18(6), 42–50. <https://doi.org/10.1109/52.965801>.
- Prochazka, J. (2011). Agile support and maintenance of IT services. *Information Systems Development* (p. 597–609). Springer New York. [https://doi.org/10.1007/978-1-4419-9790-6\\_48](https://doi.org/10.1007/978-1-4419-9790-6_48).
- Rigby, D. K., Sutherland, J., & Takeuchi, H. (2016). *Embracing agile*. *Harvard Business Review*. Recuperado em Novembro 23, 2017, de <https://hbr.org/2016/05/embracing-agile>.
- Schwaber, K., & Sutherland, J. (2013). *Um guia definitivo para o Scrum: As regras do jogo*. Scrum.Org and ScrumInc. <https://doi.org/10.1053/j.jrn.2009.08.012>.
- Seikola, M., Loisa, H.-M., & Jagos, A. (2011). Kanban implementation in a telecom product maintenance. *Proceedings - 37th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2011*, 321–329. <https://doi.org/10.1109/SEAA.2011.56>.
- Tarwani, S., & Chug, A. (2016). Agile methodologies in software maintenance: A Systematic Review. *Informatica; Ljubljana*, 40(4), 415–426. Recuperado de <https://search.proquest.com/docview/1883493065/abstract/A613FB9EBF844548PQ/6>
- Terlecka, K. (2012). Combining Kanban and Scrum -- Lessons from a team of sysadmins. *Proceedings - 2012 Agile Conference, Agile 2012*, 99–102. <https://doi.org/10.1109/Agile.2012.20>.