

## **Agile software development: best practices and complementary actions**

**Claudia Carrijo Ravaglia**

*claudia@topdown.com.br*

**MIRIAN PICININI MÉXAS**

*mirian.picinini@gmail.com*

**kleber da silva nunes**

*kleber@topdown.com.br*

To increase competitiveness of companies in IT sector, is necessary to understand the best practices being adopted for the management of software development projects and the reasons for failures in this process. Thus, this knowledge can be used in new projects providing more quality to the final product. In order to evaluate agile software management, its restrictions, best practices and complementary practices, an exploratory survey was carried out at the Scopus and Web of Science databases, searching for articles on agile software development project management, identifying their best practices and the complementary actions adopted that can add value to their management. As a contribution, it is expected that the failures and successes that were identified can be used as lessons learned and generate competitive advantages for companies that use the agile methodology in the management of software development projects.

Keywords: Agile methods, Software development, Project management

## **Desenvolvimento de software ágil: melhores práticas e ações complementares**

Área Temática: 1. GESTÃO PELA QUALIDADE

**Claudia Carrijo Ravaglia**

*claudia@topdown.com.br*

**MIRIAN PICININI MÉXAS**

*mirian.picinini@gmail.com*

**kleber da silva nunes**

*kleber@topdown.com.br*

Para que as empresas de tecnologia da informação fiquem cada vez mais competitivas, é necessário entender as melhores práticas que estão sendo adotadas no gerenciamento de projetos de desenvolvimento de software e os motivos de suas falhas ocorridas no processo. Assim, estes conhecimentos poderão ser utilizados em novos projetos proporcionando cada vez mais qualidade ao produto final. Com o objetivo de avaliar o gerenciamento de software ágil, suas restrições, as melhores práticas e as práticas complementares, foi realizada análise exploratória nas bases Scopus e Web of Science, buscando artigos sobre gerenciamento de projeto de desenvolvimento de software ágil, identificando suas melhores práticas e as ações complementares adotadas que agregaram valor ao gerenciamento. Como contribuição, espera-se que as falhas e sucessos aqui identificados possam ser utilizados como lições aprendidas e gerar vantagens competitivas para as empresas que utilizam a metodologia ágil no gerenciamento de projetos de desenvolvimento de software.

Palavras-chave: Métodos ágeis, Desenvolvimento de software, Gerenciamento de projetos

## **1 Introdução**

### **1.1 Considerações Iniciais**

Empresas de tecnologia da informação (TI) precisam ser cada vez mais competitivas, devido à globalização e a velocidade com que as mudanças vêm ocorrendo no mundo. Deste modo, precisam investir em qualidade na condução de seus projetos e na gestão do desenvolvimento de *software* para minimizar seus prazos e custos.

Não existe uma forma certa de gerenciamento e controle para alcançar o sucesso de um projeto. Muitos projetos de *software* falham por diversos fatores tais como: o custo foi excedido; atrasos na entrega; o escopo planejado não foi entregue, dentre vários outros motivos. O aprendizado é contínuo e errar também faz parte do processo de crescimento e amadurecimento do conhecimento. Por estes motivos é preciso constante monitoramento da qualidade do produto.

Para evitar que os problemas de gerenciamento nos projetos de desenvolvimento de *software* ocorram, existem duas grandes vertentes, os princípios clássicos de gerenciamento de projetos e os métodos ágeis de desenvolvimento, compartilhando os mesmos objetivos: concluir projeto com a qualidade prometida, no prazo acordado, otimizando os recursos agregados. O desafio está em escolher, adaptar e associar estas abordagens de forma a alcançar a excelência na execução do projeto (VARGAS, 2016).

### **1.2 Formulação da Situação Problema**

Os adeptos dos métodos de desenvolvimento ágil se contrapõem aos métodos convencionais e consideram os projetos de desenvolvimento tradicionais engessados, resistentes a mudanças, com custos elevados, grande quantidade de documentação e burocratização excessiva. Já os desenvolvedores que utilizam os métodos tradicionais acreditam que os desenvolvedores ágeis não conseguirão evoluir seus *softwares* de forma que atendam a um escopo maior e integrado.

Vale destacar que existem muitos processos em ambas as metodologias que podem agregar valor ao resultado final, sendo o escopo deste estudo a avaliação do gerenciamento de desenvolvimento de *software* ágil.

### **1.3 Objetivo Geral**

Dentro deste contexto, foi realizada uma pesquisa na literatura sobre o gerenciamento de projeto de desenvolvimento de *software* que utilizam os métodos ágeis, com o objetivo de

avaliar suas restrições, as melhores práticas e as práticas complementares que são utilizadas para alcançar melhores resultados e qualidade do *software* desenvolvido.

Contribuindo desta forma, para que as falhas e sucessos aqui identificados possam ser utilizados como lições aprendidas para o gerenciamento de projetos que utilizam a metodologia ágil no desenvolvimento de *software*.

## **1.4 Metodologia**

Foi realizada uma análise exploratória através de um levantamento bibliográfico, realizando buscas por artigos disponíveis no portal: *SCOPUS* e *Web of Science*, sobre métodos ágeis e gerenciamento de projetos de *software*, onde foram avaliadas as melhores práticas adotadas e as ações complementares necessárias.

## **2 Fundamentação Teórica**

Todo empreendimento, seja ele de escopo pessoal ou que gere resultados financeiros, é considerado projeto e depende de uma série de fatores para se tornar um sucesso. Para tanto, Pinto, Vasconcelos e Lezana (2014) definem que se deve estabelecer os objetivos do projeto e realizar o gerenciamento de suas demandas: escopo, custo e prazo.

De acordo com o GUIA PMBOK (2013), o sucesso de um projeto depende de seu gerenciamento que é realizado através da aplicação e da integração de cinco grupos de processos: iniciação, planejamento, execução, monitoramento e controle, e encerramento. Projetos bem gerenciados finalizam no prazo, dentro do orçamento e atendendo ao que foi definido em seu escopo.

Também devem incluir: identificação de requisitos, diferentes abordagens no seu planejamento e execução, comunicação ativa e eficaz entre as partes interessadas, atender aos requisitos do projeto e realização de suas entregas, equilibrar as restrições conflitantes (escopo, qualidade, cronograma, orçamento, recursos e risco).

O gerenciamento de projetos é utilizado em todos os tipos de organização privadas ou governamentais com ou sem fins lucrativos, e dentro deste contexto, também se encontra o desenvolvimento de *software*.

Para o desenvolvimento de um *software* são necessárias técnicas, tecnologias, práticas de gerenciamento e ter qualidade. Todas estas necessidades formaram a área da computação denominada Engenharia de *Software*, que define três itens básicos para desenvolvimento de um *software* com qualidade: produtos entregues no prazo e atendendo ao orçamento,

requisitos do cliente atendidos e deve ter fácil usabilidade (PINTO, VASCONCELOS e LEZANA, 2014).

Sistemas de *software* evoluem continuamente. Suas evoluções são muitas vezes baseadas em resposta à evolução de seus ambientes: novas funcionalidades são adicionadas as existentes ou são removidas devido à necessidade de abordar mudanças nos requisitos de negócios, ou forças econômicas e novas plataformas são desenvolvidas, ou novas tecnologias são apresentadas. Isso requer uma mudança de paradigma na forma como os sistemas de *software* serão desenvolvidos, exigindo agilidade nas mudanças (BOWEN *et al.*, 2014).

O desenvolvimento ágil foi criado com base nos conceitos do Manifesto Ágil, de acordo com Beck *et al.* (2001), onde seus doze princípios tinham como objetivo a satisfação do cliente, entregas frequentes, comunicação face a face, simplicidade, design, equipes auto organizada e as mudanças não configurando um problema.

Existem diferentes modelos de processos ágeis e muitas semelhanças em suas abordagens, onde misturam conceitos de engenharia de *software* e todos satisfazem aos princípios do Manifesto para o Desenvolvimento Ágil de *Software*.

Segundo Dingsøyr *et al.* (2017), sendo a participação do cliente indispensável nos métodos ágeis, ela também é considerada um desafio nos esforços de desenvolvimento de sistemas grandes e o risco de falha aumenta. Por ter um grande número de *stakeholders*, já que os diferentes grupos de *stakeholders* têm prioridades diferentes.

Selecionar o que melhor se adapta ao seu projeto depende de muitos fatores, tais como: tamanho da equipe, criticidade dos defeitos, capacidade de receber dos usuários *feedback*, expectativas dos clientes, orçamento, cronograma e escalabilidade dos requisitos. Como esses fatores são diferentes de projeto para projeto, não devem ser utilizadas sempre as mesmas estratégias (SURDU e PARSONS, 2006). Assim, a revisão de falhas e sucessos, as lições aprendidas em projetos de desenvolvimento ágeis, podem ajudar na identificação dos possíveis fatores críticos de sucesso, já que a compreensão das falhas e ações bem sucedidas podem contribuir para a melhoria da qualidade dos projetos (CHOW e CAO, 2008).

Recente publicação do Guia PMBOK 6ª Edição apresentou um guia específico associando o PMBOK as práticas ágeis, o *Agile Practice Guide* PMI (2017), fazendo parte do escopo deste guia: cobrir as abordagens ágeis mais populares e mapear o ágil para o PMBOK. O mapeamento da aplicação ágil foi feito para as áreas de gerenciamento do PMBOK: gerenciamento de integração, escopo, tempo, custos, qualidade, recursos do projeto, comunicações, riscos, aquisições e partes interessadas.

### 3 Análise dos Resultados do Levantamento Bibliográfico

Em pesquisa realizada em outubro de 2017, conforme Quadro 1, foi realizada uma busca nas bases *web of Science e Scopus*, utilizando os mesmos critérios, conforme os listados a seguir.

Base	Critério Utilizado	Quantidade Recuperada
<i>Web of Science</i>	Tópico: (" <i>Agile Methods</i> ") AND Tópico: (" <i>software development</i> ") AND Tópico: (" <i>Project Management</i> ") Refinado por: Tipos de documento: ( <i>ARTICLE</i> ) – Sem opção <i>Review e in Press</i> .	17
<i>Scopus</i>	( <i>TITLE-ABS-KEY</i> ( " <i>Agile methods</i> " ) AND <i>TITLE-ABS-KEY</i> ( " <i>software development</i> " ) AND <i>TITLE-ABS-KEY</i> ( " <i>Project Management</i> " ) ) AND ( <i>LIMIT-TO</i> ( <i>DOCTYPE</i> , " <i>ar</i> " ) OR <i>LIMIT-TO</i> ( <i>DOCTYPE</i> , " <i>re</i> " ) OR <i>LIMIT-TO</i> ( <i>DOCTYPE</i> , " <i>ip</i> " ) )	44

Quadro 1: Resultado pesquisa Base – *Web Of Science e Scopus*  
Fonte: Elaboração própria

Esta consulta retornou um total de 61 artigos, sendo nove deles descartados por duplicidade. Os artigos restantes foram avaliados, sendo selecionados 18 artigos, que apresentaram aderência ao estudo. Buscou-se identificar as melhores práticas de gerenciamento de projetos de desenvolvimento de *software* ágil e as ações complementares adotadas que agregaram valor a este gerenciamento.

Os autores Al-Sakkaf, Hashim e Omar (2017), acreditando que a adoção completa do método ágil para desenvolvimento de *softwares* pode ser um excesso ou demandar muitos recursos, as equipes ágeis precisam de ajuda para escolher a combinação correta de suas práticas, assim, buscaram identificar práticas correlatas para maximizar seus benefícios e identificaram quatro grupos: Gerenciamento de projetos (contendo as práticas: reunião diária, iterações curtas, retrospectivas, planejamento das iterações); Garantia da qualidade (contendo as práticas: teste de unidade, padrões de codificação, integração contínua, desenvolvimento orientado por teste); Comunicação da equipe (contendo as práticas: *backlogs* prioritários, equipe única, refatoração e área de trabalho aberta); e Grupos incrementais e iterativos (contendo as práticas: estimativa baseada na equipe, avaliações das iterações, proprietário do produto dedicado, mapeamento das histórias, código coletivo, desenvolvimento contínuo, planejamento de release, *user story*).

Dingsøyr *et al.* (2017) questionaram como os métodos ágeis poderiam ser adaptados à projetos de grandes escalas, em relação a organização do programa, envolvimento do cliente, arquitetura de *software* e coordenação de equipes. Utilizando um projeto que havia sido

desenvolvido e usando uma combinação de métodos ágeis e tradicionais, identificaram que houve o envolvimento do cliente e das equipes na descrição da solução e planejamento contínuo, garantindo o alinhamento entre as necessidades do negócio e o desenvolvimento do *software*. O ambiente aberto proporcionou o compartilhamento de conhecimento na formação da cultura organizacional, com mais confiança e relacionamentos colaborativos, melhorando a integração e aprendizagem, com comunicação rápida em reuniões informais, porém houve deficiências na divulgação da informação. Identificaram a necessidade na melhor compreensão para elaboração da arquitetura em sistemas mais complexos.

Kulkarni *et al.* (2017) utilizaram entrevista qualitativa em uma equipe com experiência nas práticas de desenvolvimento nas metodologias tradicionais e ágeis, para avaliar como as práticas de desenvolvimento ágil afetam o desempenho dos projetos de desenvolvimento de *software*. Alcançando mais de 80% de concordância nos seguintes pontos: Especificação (comunicação cara-a-cara e aprendizado contínuo); Cronograma (divisão do projeto em ciclos / iterações curtas, orientado a testes e centrado nas pessoas); Velocidade. Trabalho em equipe (os membros da equipe devem estar fisicamente próximas; discutir regularmente seus erros); A equipe de negócios e os desenvolvedores devem trabalhar juntas; devem ser auto organizadas, competentes e experientes, refletir sobre como se tornar mais eficaz. Colaboração do cliente (os clientes devem colaborar com os membros da equipe de desenvolvimento; utilizar priorizações); Atenção contínua à excelência técnica e ao bom *design*; Desafios / Riscos (diferenças de produtividade entre os membros da equipe); A integração entre as equipes desenvolvimento tradicionais e as equipes ágeis; Problemas com a seleção da metodologia ágil apropriada e as ferramentas.

Os autores Al-Aidaros e Omar (2017) avaliaram diferentes abordagens para o monitoramento de *Work in Progress* (WIP) no desenvolvimento de projetos de *software* diferentes. Os métodos ágeis são reconhecidos em desenvolvimento de *software* por sua flexibilidade e eficácia, sendo o *Scrum* e o *Kanban* considerados os dois métodos mais poderosos, porém os estudos confirmaram que as falhas nos projetos que utilizam estes métodos se devem à falta de monitoramento, comunicação deficiente e falta de relatório sobre o status do projeto, assim, foi identificada a necessidade de mecanismos de monitoramento. O monitoramento do processo de desenvolvimento do projeto de *software* é essencial para garantir que o projeto seja realizado dentro do orçamento, prazo e qualidade.

Farid, Abd Elghany e Helmy (2016) compararam o *Scrum* para simplificar a implementação das práticas específicas do CMMI-DEV Versão 1.3, mapeando as práticas

específicas entre CMMI e o *Scrum*. Apesar das duas abordagens serem aparentemente diferentes, concluiu-se que podem ser combinadas trazendo mais benefícios do que quando utilizados isoladamente. O *Scrum* forneceu boas práticas para atender aos níveis de maturidade 2 e 3 do CMMI onde foram identificados suas cinco melhores práticas: *Sprint Planning*, reunião diária, retrospectiva, planejamento das *releases* e *product backlog*, porém não atendeu a uma parcela do nível 4 do CMMI, não oferecendo práticas de planejamento e controle do orçamento do projeto, cálculo dos custos, práticas para gestão de riscos, planejamento e rastreamento do gerenciamento de dados e práticas para a aquisição de produtos de fornecedores.

Os autores Nuottila, Aaltonen e Kujala (2016) estudaram os tipos de desafios que uma organização pública enfrenta ao adotar práticas ágeis. O método ágil adotado foi o *Scrum* ligeiramente modificado. Foram identificados vários desafios importantes, alguns similares aos que já foram discutidos e outros exclusivos de uma organização pública. A análise foi realizada através de dados qualitativos e, após a análise, os desafios foram agrupados em sete categorias: documentação; educação, experiência e compromisso; comunicação e envolvimento dos *Stakeholders*; funções nas configurações ágeis; localização das equipes ágeis; legislação; complexidade da arquitetura de *software* e integração de sistemas.

Tripp, Riemenschneider e Thatcher (2016) avaliaram como o uso de diferentes práticas ágeis e quais destas práticas afetam a satisfação no trabalho. A pesquisa não considerou as diferenças culturais e geográficas nem *outsourcing* e *offshoring*, também não explicou teoricamente nem analisou empiricamente como as práticas ágeis de desenvolvimento se relacionam com os sentimentos dos membros da equipe sobre seu trabalho, sendo identificados cinco características que influenciam a percepção dos trabalhadores: significado da tarefa; identidade da tarefa; variedade de habilidades; autonomia do trabalho; e *feedback*. Sendo identificado que os gerenciamentos de projetos ágeis impactam principalmente a autonomia e *feedback* do trabalho, enquanto as abordagens de desenvolvimento de *softwares* ágeis impactam principalmente a tarefa (variedade de habilidades, identidade da tarefa e significado da tarefa), onde a satisfação aumenta quando as práticas são combinadas. Assim, sugere-se que gerentes devem apoiar e encorajar as práticas ágeis de desenvolvimento de *software* ágil e gerenciamento de projetos ágil para maximizar a satisfação no trabalho dos membros da equipe de desenvolvimento. Esta pesquisa permitiu relacionar métodos ágeis a satisfação no trabalho, afetando positivamente a qualidade do produto.



Para Vargas (2016), o gerenciamento ágil surgiu para suprir a necessidade de gerência sem burocracia, com desenvolvimento com qualidade e rapidez. Porém o foco quase exclusivo nos processos de desenvolvimento do projeto deixam áreas complementares ao projeto enfraquecidas quando não administradas pelos gestores, gerando problemas na administração ou gerenciamento do processo de desenvolvimento do *software*, surgindo assim a necessidade de controle. Nesse contexto, a compatibilização do *Scrum* com um método de gestão de projetos tradicional pode facilitar a adoção do ágil e trazer benefícios para a gestão de desenvolvimento de *software*. A autora, através de pesquisa qualitativa e bibliográfica, avaliou o equilíbrio entre o *Scrum* e os métodos de gestão de projetos: PMBOK e PRINCE2. Concluindo que, apesar de poder utilizar o *Scrum* sozinho, o sucesso de um projeto depende de seu planejamento e utilização de práticas compatíveis com a filosofia da organização, assim, o método ágil deve se tornar parte da cultura organizacional e a gerência deve facilitar o desenvolvimento realizando a compatibilização do *Scrum* independentemente do método de gerenciamento adotado, PMBOK ou PRINCE2.

Laanti (2014) estudou quais os princípios deveriam ser usados com o SAFe para gerar agilidade e sugeriram como estes princípios poderiam ser construídos. Como resultado da pesquisa encontraram uma lista contendo 21 princípios revisados por pares e acrescido de algumas sugestões de especialistas, onde a origem de cada um deles foi apresentada no artigo. A seguir a lista dos princípios levantados: (1) Conteúdo é a chave: usar o *feedback* do usuário e o conhecimento intrínseco baseado na experiência; (2) Co-criação: um problema é resolvido melhor em grupo do que individualmente ; (3) *Feedback* é o combustível da aprendizagem: avalie o *feedback* e faça mais o que gera melhores resultados; (4) Agilidade dos negócios: o interesse do usuário define o modelo de negócios e a frequência na entrega das releases; (5) Uso da automação: a automação alivia o esforço manual melhorando a produtividade; (6) Escalar utilizando Fractais: use níveis de abstração mais altos e sistemas aninhados, como loops de controle aninhados; (7) Evite Explosões Combinatória: a complexidade é melhor resolvida quando dividida; (8) Sequência para o rendimento máximo: encontrar o rendimento máximo do portfólio, equilibrando o que pode ser feito em paralelo e o que deve ser feito em sequência; (9) Valorizar o conhecimento: use os melhores especialistas para resolver os problemas mais complexos e importantes e dê espaço para a criatividade; (10) Nível de trabalho: priorização, distribuição e eliminação de trabalho desnecessário, utilização do Kanban; (11) Simplicidade: buscar a simplicidade nas soluções; (12) Situacionalidade: nem todos os casos precisam ser tratados da mesma forma, use princípio de Pareto para evitar

processos excessivamente complexos; (13) Controle os Processos não os itens: criar regras simples para a tomada de decisões, em vez de controlar cada decisão individualmente; (14) Mentalidade de crescimento: faça mais e melhore o que criou sucesso; (15) Escute os funcionários, eles sabem todos os problemas: o valor é criado na linha de frente; (16) Descubra e use padrões : seus problemas já foram resolvidos por alguém em algum lugar; (17) Inovação nos custos: apresente solução com menores custos, valorize a flexibilidade nos investimentos; (18) Utilize o conhecimento tácito: use o conhecimento das pessoas; (19) Aprendizagem entre as equipes: crie conhecimento coletivo que compartilhem a mesma visão e ambição; (20) Velocidade é melhor do que a perfeição: tolerar pequenas imperfeições. O ótimo é o inimigo do bom; (21) Prevenir problemas enquanto são pequenos: o sucesso esconde pequenos problemas, não os ignore.

Os autores Bowen *et al.* (2014) acreditam que os desenvolvedores ágeis poderiam se beneficiar com um treinamento de métodos formais. A formalidade no desenvolvimento de um sistema de *software* pode torná-lo lento, porém a quantidade de erro é reduzida, tornando seus custos atraentes. Nos casos onde a segurança é um fator importante, a formalidade é a abordagem indicada. Uma especificação formal também fornece bases para realização de testes rigorosos, porém esta abordagem contrasta com a abordagem ágil de desenvolvimento de *software*, sendo mais iterativa com colaboração dos membros da equipe e a evolução dos requisitos com respostas rápidas às mudanças. A evolução de *software* e a conversão dos sistemas legados devem preservar suas funcionalidades sem afetar seriamente o desempenho. Para atender a preocupação com a evolução dos sistemas legados, foi desenvolvida uma abordagem ágil combinada com as técnicas formais de especificações, melhorando a compreensão precoce, orientando o programador e auxiliando os testes adequados. Neste contexto, é mais provável o sucesso, com uso combinado do desenvolvimento ágil e o julgamento da engenharia experiente.

Sundararajan, Bhasi e Vijayaraghavan (2014) exploraram as práticas de gerenciamento de riscos em um grande projeto de desenvolvimento de *software* ágil *Scrum offshore* e terceirizado, que foi executado por equipes geograficamente distribuídas e com fornecedores diferentes, sendo selecionados 20 fatores. Algumas práticas do método de desenvolvimento tradicional foram integradas ao método ágil formando um modelo híbrido. Incluído *Sprint* zero, prova de conceito, processos e procedimentos de teste, testes por unidade e testes finais. Os requisitos de negócio e funcionalidades foram definidos de forma detalhada, a arquitetura do sistema, o *design* do banco de dados, as interfaces do *software* e os critérios de aceitação

para entrega foram definidos na inicialização do projeto. O projeto teve papéis como arquiteto, *Scrum master*, *designer*, desenvolvedor, testador, analista de negócios, especialista em testes, coordenador no local e facilitadores e alguns membros da equipe desempenharam mais de um papel. Houve uma combinação de estrutura plana adequada à cultura ágil e estrutura hierárquica necessária para a coordenação de grandes projetos. Contrariando à prática ágil, foi dado maior importância ao desempenho individual. Medidas adicionais para a retenção de conhecimento foram adotadas como documentação utilização dos processos CMMI nível 5, uso de diretrizes de documentação UML e utilização de repositório centralizado de documentos para compartilhamento. Todos os membros da equipe foram treinados para entender os valores e princípios ágeis. Concluindo que os projetos ágeis são bons candidatos para o estudo realizado, se os riscos específicos a este tipo de projeto puderem ser identificados, avaliados e gerenciados.

Papatheocharous e Andreou (2014), motivados pelo fato de que poucas pesquisas se concentraram na identificação dos fatores críticos que afetam a adoção ágil, utilizaram pesquisa *online*, incluindo cinco seções: Informações pessoais, papéis, e experiência; Informações da empresa / organização; Membros da equipe e comunicação; Conhecimento e estratégias; e Adoção ágil, visando avaliar a adoção da abordagem ágil nas organizações de *software* em todo o mundo e suas implicações práticas no desenvolvimento e nos aspectos organizacionais e metodológicos. Com relação a localização das equipes, foi identificado que não existe um padrão, elas podem estar no mesmo ambiente ou distantes que irão trabalhar da mesma forma, o *Scrum* foi identificado como a metodologia preferida, seguido por abordagens customizadas, ou seja, adaptações específicas para o ambiente das empresas. Os principais objetivos identificados foram a produção de *software* em todas as iterações/*sprints*/fases, o *feedback* imediato dos clientes, simplicidade, discussão dos objetivos do projeto em grupo e entregas *software* dentro do prazo. Identificado também que o paradigma ágil é adotado por equipes pequenas e grandes, a comunicação é realizada por diversas formas sendo que as mais utilizadas são as conversas face-a-face, *e-mail* e reuniões curtas. O desenvolvimento ágil também adota diversas estratégias de testes, reuniões *stand-up* e utiliza ferramentas para acompanhar o progresso através de relatórios de status. Com relação aos padrões de qualidade de *software*, foi identificado que apenas metade das equipes ágeis declararam que seguiram algum padrão como exemplo ISO9000, SPICE, CMM e CMMI. Os principais benefícios apontados foram a velocidade e a capacidade de gerenciar e priorizar mudanças de requisitos, e os principais motivos que geraram falhas de projetos ágeis estavam

relacionados a falta de conhecimento ou experiência com abordagens ágeis e a resistência à mudança dos métodos de desenvolvimento tradicionais para o ágil.

Wautelet e Kolp (2012) elaboraram um jogo para ensinar aos gerentes de projetos de *software* e aos estudantes de informática as habilidades necessárias para lidar com os processos de desenvolvimento. Existem poucas abordagens sobre este aprendizado, apesar de existirem diversas metodologias de desenvolvimento de *software*, a grande maioria aprende nos cursos e livros ou nos projetos reais, onde as abordagens apresentam falhas. A teoria não destaca a dimensão prática e a aprendizagem empírica expõe a organização a erros e prejuízos. O jogo utilizou métodos *Unified Process* (UP) e ágil onde são avaliadas as habilidades no gerenciamento: planejamento, custo, tempo, recursos, risco, qualidade, através de simulações de ameaças e oportunidades aleatórias não previstas, onde o jogador deve realizar análise de impacto e tomar decisões para que suas habilidades sejam avaliadas. O jogo habilita o indivíduo no gerenciamento do processo completo do *software* de forma adequada, possibilitando o aprendizado das principais dimensões e conceitos de gerenciamento de projetos de *software*.

Mahnic e Zabkar (2012) relatam a preocupação com a falta de controle de gestão na adoção de métodos de desenvolvimento de *software* ágeis, porém a medição deve ser realizada de forma transparente não interferindo na agilidade do processo de desenvolvimento. Os autores utilizaram um projeto *Scrum* para aplicar um modelo de monitoramento de desempenho, envolvendo diferentes *stakeholders*. As medições foram disponibilizadas em um repositório de medição, e para as considerações dos requisitos utilizaram a área de processo de medição do CMMI, e a para avaliação da conformidade, utilizaram o modelo COBIT. As medidas de progresso observadas foram: velocidade, quantidade de trabalho restante e índices de desempenho e custo. As medições geraram aprendizagem com os ciclos de planejamento anteriores proporcionando melhoria nas estimativas. Assim, concluíram que houve grande vantagem em se medir o progresso dos projetos de *softwares* baseados em *Scrum* e que não houve trabalho administrativo adicional para realização das medidas, sendo a coleta de dados um processo valioso.

Sutharshan (2011) alerta para problemas que surgem quando os desenvolvedores de diferentes culturas trabalham juntos. Os fatores humanos são componentes significativos que determinam o sucesso de um projeto, sendo a cultura importante em vários aspectos do negócio. Uma empresa deve interagir com pessoas: clientes, funcionários, fornecedores, *stakeholders* e a participação do usuário final também são fatores importantes no sucesso no

projeto. Os Métodos ágeis foram projetados para melhorar o gerenciamento de equipe de projetos de *software*, sendo comum em grandes projetos, existirem equipes de diferentes localidades com diferenças culturais, estas diferenças geram expectativas diferentes e reações diferentes, o que pode gerar impacto no sucesso do desenvolvimento de *software* afetando a adoção de uma metodologia. O estudo correlacionou os atributos culturais e os princípios do método ágil, definidos no manifesto ágil, auxiliando a adequação do trabalho com equipe de projeto multicultural na realização de projetos de desenvolvimento de *software* com métodos ágeis.

Lee e Yong (2010), através de estudo de caso de sucesso onde foram utilizadas práticas integradas de desenvolvimento de *software* ágil e desenvolvimento de *software* distribuído, examinaram os desafios enfrentados por equipes de *software* distribuídas globalmente e pesquisaram as melhores práticas para apoiar o gerenciamento de projetos usando uma abordagem ágil distribuída. A maioria dos desafios enfrentados por estas equipes está relacionada à comunicação e à confiança. Construir uma base de confiança e respeito através das diferenças culturais e linguísticas é difícil como também o compartilhamento do conhecimento tácito. Portanto, é importante garantir uma participação justa e acessível para todos os membros da equipe, manter uma visão compartilhada e proporcionar transparência ao progresso para as partes interessadas, tanto dentro como fora da organização. O grupo de serviços globais implementou o *Scrum* de alto desempenho com as práticas de engenharia XP onde foram abordados: Flexibilidade do processo; Empoderamento; Espaços de trabalho compartilhados; Conferência; e Coaching. Estas práticas foram adotadas para melhoria do processo solucionando problemas identificados na implementação anterior, incorporando o *Scrum* ao processo de implantação.

Chow e Cao (2008) se basearam na literatura para classificar os fatores críticos de sucesso de projetos ágeis em cinco categorias divididas em 55 fatores entre falhas e sucessos. Após análise os autores realizaram uma redução para 12 fatores principais e os associaram a quatro categorias: organização, pessoas, processos e técnicas, avaliando-os nas dimensões: Qualidade, Escopo, Tempo e Custo. O resultado da pesquisa, realizada através de coleta de dados com abordagem quantitativa e realizada em 109 projetos ágeis de 25 países ao redor do mundo, permitiu que os autores concluíssem que apesar da extensa lista preliminar apenas 10 hipóteses foram suportadas e os únicos fatores que podem ser chamados de fatores críticos de sucesso para projetos de desenvolvimento de *software* ágil são: estratégia de entrega; técnicas de engenharia de *software* ágil e capacidade da equipe. Outros três fatores podem ser

considerados críticos, porém apenas em algumas dimensões: processo de gerenciamento de projetos ágil de qualidade; ambiente amigável e forte envolvimento do cliente.

Surdu e Parsons (2006) relatam sobre o desenvolvimento do *software* OneSAF - *One Semi-Automated Forces*, que foi desenvolvido para apoiar pesquisa, desenvolvimento e treinamento de futuros líderes militares dos EUA, utilizando técnicas ágeis, XP e tradicionais, foi desenvolvido através de 26 empresas diferentes, onde processos, ferramentas e procedimentos foram estabelecidos, mantendo-se a preocupação com a melhoria contínua e busca de novas ideias. Assim, três fatores contribuíram para o sucesso do gerente do projeto: o envolvimento da equipe no processo de desenvolvimento; a flexibilidade para estabelecer processos de desenvolvimento, integração e teste a partir das melhores práticas do setor e pessoas tecnicamente qualificadas. Foi utilizado um ambiente integrado onde todas as equipes foram colocadas em uma única instalação. Como definido nos métodos ágeis, adotaram reuniões informais e deram ênfase a volatilidade dos requisitos adotando o desenvolvimento espiral, com entregas frequentes, assim os clientes evoluíam os requisitos e desejos à medida que visualizavam os protótipos de trabalho. A colaboração do cliente e a capacidade de resposta às mudanças gerou nos usuários confiança no processo e na equipe de desenvolvimento. Toda mudança confirmada no sistema de gerenciamento de configuração era submetida a testes automatizados e o desenvolvedor e seu supervisor recebiam e-mails em caso de falhas para que pudessem adotar as medidas corretivas necessárias. Para cumprir as exigências do CMMI nível 5, foi necessária uma forte documentação, onde não só os processos eram documentados, mas também seus resultados, os artefatos que documentaram a adesão dos desenvolvedores, as pastas de desenvolvimento de *software on-line*, ferramentas de rastreamento, relatórios de problemas, defeitos, avaliações de pares e riscos, tudo foi disponibilizado na *web*, o que reduziu o peso sobre os desenvolvedores para o cumprimento do processo de documentação e permitiu flexibilidade na comunicação entre equipes. A mistura de técnicas ágeis, XP e tradicionais foram fundamentais no programa permitindo alcançar metas e objetivos técnicos.

O Quadro 2 exibe o resumo dos artigos.

Referência	Achados	Itens identificados	Itens Atenção
Al-Sakkaf, Hashim e Omar (2017)	Identificar grupos de práticas ágeis que possam maximizar os benefícios se adotados através de práticas correlatas	Identificados quatro grupos de práticas ágeis que podem maximizar o resultado dos projetos: Gerenciamento de projetos; Garantia da qualidade; Comunicação da equipe; Grupos incrementais e iterativos. Visando a maximização dos resultados.	A adoção completa do método ágil pode ser um excesso ou demandar muitos recursos
Dingsøyr et al. (2017)	Estudo para utilizar o método ágil em grandes projetos.	Utilização de combinação de métodos ágeis e tradicionais. Envolvimento do cliente e equipes garantiu o desenvolvimento do <i>software</i> em conformidade com as expectativas do negócio. A utilização de ambiente aberto e reuniões informais favoreceu o compartilhamento do conhecimento e a formação da cultura organizacional.	Identificado problema de comunicação e necessidade de compreensão de como as metodologias ágeis afetam os projetos arquitetônicos em sistemas complexos.
Kulkarni et al. (2017)	Avaliaram como as práticas de desenvolvimento ágil afetam o desempenho dos projetos de desenvolvimento de software.	Identificado que as práticas ágil que afetam o desempenho do desenvolvimento de <i>software</i> são a especificação, cronograma, trabalho em equipe, colaboração do cliente e risco.	
Al-Aidaros e Omar (2017)	Avaliaram diferentes abordagens para o monitoramento no desenvolvimento de projetos de software utilizando métodos ágeis.	Métodos ágeis possuem flexibilidade e eficácia.	Falhas dos projetos ágeis estão relacionadas a falta de monitoramento, comunicação deficiente e falta de relatório sobre o status do projeto.
Farid, Abd Elghany e Helmy (2016)	Mapearam as práticas específicas entre CMMI e o Scrum.	O Scrum forneceu boas práticas para atender aos níveis de maturidade 2 e 3 do CMMI, sendo apontado como melhores práticas: <i>Sprint Planning</i> , reunião diária, retrospectiva, planejamento das <i>releases</i> e <i>product backlog</i> .	O Scrum não forneceu práticas para atender ao CMMI nível 4 nos itens de planejamento e controle do orçamento do projeto, cálculo dos custos, práticas para gestão de riscos, planejamento e rastreamento do gerenciamento de dados e práticas para a aquisição de produtos de fornecedores.
Nuottila, Aaltonen e Kujala (2016)	Desafios que uma organização pública enfrenta ao adotar práticas ágeis	Identificado sete categorias: Documentação; Educação, experiência e compromisso; Comunicação e envolvimento dos Stakeholders; Funções nas configurações ágeis; Localização das equipes ágeis; Legislação; Complexidade da arquitetura de software e integração de sistemas	

Quadro2: Resumo dos artigos – Vantagens, desvantagens e pontos de atenção

Fonte: Elaboração Própria

<b>Tripp, Riemenschneider e Thatcher (2016)</b>	Avaliaram como e quais práticas ágeis afetam a satisfação no trabalho.	As práticas ágeis afetam positivamente a satisfação no trabalho gerando a qualidade do produto. Identificadas cinco características que influenciam a percepção dos trabalhadores: Significado da tarefa; Identidade da tarefa; Variedade de habilidades; Autonomia do trabalho; e Feedback.	A satisfação do trabalho aumenta quando são combinadas práticas de gerenciamento de projeto ágil e práticas de desenvolvimento ágil.
<b>Vargas, L. M. (2016)</b>	Avaliou a compatibilização do Scrum com um método de gestão de projetos tradicional se proporcionam benefícios para a gestão de desenvolvimento de software.	Identificado qualidade em projetos com utilização do Scrum e práticas de gerenciamento do PMBOK e PRINCE2.	
<b>Laanti (2014)</b>	Estudou quais os princípios deveriam ser usados com o Scaled Agile Framework (SAFe) para gerar agilidade	Identificação de 21 princípios que deveriam ser utilizados com o SAFe para gerar agilidade. Destacando-se o <i>feedback</i> , base de conhecimento, frequência das entregas, velocidade, simplicidade, padronização, automação e prevenção de problemas.	
<b>Bowen et al. (2014)</b>	Desenvolvedores ágeis poderiam se beneficiar com técnicas formais de especificações.	Identificado que o programador tem melhor orientação e os testes são mais adequados quando o ágil é combinado com técnicas formais de especificação.	Quando o fator segurança é importante, o desenvolvimento ágil pode ser uma ameaça.
<b>Sundararajan, Bhasi e Vijayaraghavan (2014)</b>	Exploraram as práticas de gerenciamento de riscos em um grande projeto de desenvolvimento de software ágil Scrum	Identificado que é possível executar projetos ágeis com equipes geograficamente distribuídas utilizando o gerenciamento de riscos com processos do CMMI e UML.	Os riscos específicos do projeto devem ser identificados, avaliados e gerenciados.
<b>Papatheocharous e Andreou (2014)</b>	Buscaram identificar dos fatores críticos que afetam a adoção ágil no desenvolvimento de software.	Identificado que os principais benefícios da metodologia ágeis são a velocidade e a capacidade de gerenciar e priorizar mudanças de requisitos. Para a adoção dos métodos ágil não importa a localização e tamanho da equipe. A comunicação é informal. São adotados diversos testes e ferramentas de gerenciamento.	A falta de conhecimento ou experiência com abordagens ágeis e a resistência à mudança dos métodos de desenvolvimento tradicionais para o ágil podem gerar falhas nos projetos.  Necessário definir padronização.
<b>Wautelet e Kolp (2012)</b>	Elaboraram um jogo para ensinar as habilidades necessárias para o gerenciamento de projetos de desenvolvimento de software.		Os gerentes de projetos de <i>software</i> devem estar capacitados e possuir habilidades de gerenciamento planejamento, custo, tempo, recursos, riscos e qualidade.
<b>Mahnic e Zabkar (2012)</b>	Avaliaram se a medição deveria ser realizada nos processos de desenvolvimento de software ágil.	Identificado grande vantagem ao realizar medições em projetos de <i>softwares</i> baseados em Scrum utilizando processos de medição do CMMI e modelo COBIT.	A realização de medições não deve gerar trabalho adicional ao projeto.

Quadro 2 (Continuação): Resumo dos artigos – Vantagens, desvantagens e pontos de atenção  
Fonte: Elaboração Própria



<b>Sutharshan (2011)</b>	Correlacionou os atributos culturais e os princípios do método ágil, auxiliando a adequação do trabalho com equipe de projeto multicultural na realização de projetos de desenvolvimento de software		Equipes com diferenças culturais possuem expectativas e reações diferentes o que pode gerar impacto na implementação da metodologia ágil.
<b>Lee e Yong (2010)</b>	Avaliaram as melhores práticas para apoiar o gerenciamento de projetos usando uma abordagem ágil distribuída	Identificado as melhores práticas para apoiar o gerenciamento de projetos ágil distribuído: Flexibilidade do processo; Empoderamento; Espaços de trabalho compartilhados; Conferência; e Coaching.	Foi identificado falta de planejamento e comunicação entre a equipe. Equipes inexperientes. Gerente de projeto local responsável por múltiplas funções. As diferenças culturais e linguísticas que podem gerar problemas de comunicação e confiança.
<b>Chow e Cao (2008)</b>	Classificaram os fatores críticos de sucesso de projetos ágeis	Identificado como fatores críticos de sucesso para projetos de desenvolvimento de <i>software</i> ágil: Estratégia de entrega, Técnicas de engenharia de <i>software</i> ágil e Capacidade de equipe.	Outros fatores também podem ser considerados críticos: Processo de gerenciamento de projetos ágil, ambiente de equipe ágil amigável e um forte envolvimento do cliente.
<b>Surdu e Parsons (2006)</b>	Relatam sobre o desenvolvimento do software que utilizaram técnicas ágeis, XP e tradicionais,	Identificação dos fatores que contribuíram para o sucesso do gerente do projeto: envolvimento da equipe; flexibilidade, integração e testes.	Necessidade de documentação.

Quadro 2 (Continuação): Resumo dos artigos – Vantagens, desvantagens e pontos de atenção  
Fonte: Elaboração Própria

## 4 CONCLUSÃO

Diversos são os métodos ágeis que podem ser adotados no desenvolvimento de *software*. As utilizações das práticas corretas para o desenvolvimento de *software* ágil podem maximizar seus benefícios, assim como sua utilização combinada com as práticas tradicionais de gerenciamento de projetos.

Alguns autores pesquisados na literatura não identificaram restrições para a utilização da metodologia ágil com projetos de grande porte e com equipes geograficamente distribuídas, entretanto outros discordam e citam que a equipe deve estar fisicamente próxima.

Diversos autores relacionam os principais benefícios na utilização do ágil para desenvolvimento de *software* sendo citados: a velocidade, flexibilidade, entregas constantes, capacidade de aceitar mudanças, diversidade dos testes e o envolvimento da equipe.

Foi também observado que as práticas ágeis afetam positivamente a equipe e geram satisfação com o trabalho. Entretanto, um ponto de atenção deve ser considerado e que pode

causar prejuízos ao gerenciamento do projeto ágil está relacionado com o fator humano e a diversidade cultural da equipe e expectativas diferentes dos diversos *stakeholders*.

Outros itens citados que podem afetar o desempenho dos projetos de desenvolvimento de *software* podem estar relacionados a comunicação, monitoramento, documentação, necessidade de definição das funções, *design* e arquitetura do projeto, falta de práticas para controlar custos, riscos e aquisição e a falta de padronização. Sendo que estes controles são complementados ao gerenciamento de projetos que utilizam o método ágil, utilizando métodos / *frameworks* tradicionais, o que reforça a necessidade da combinação dos dois métodos.

Como proposta para trabalhos futuros, sugere-se a realização de estudos para elaboração de diretrizes que possam gerenciar estes itens na utilização de desenvolvimento de *software* utilizando uma metodologia ágil.

## REFERÊNCIAS

AL-AIDAROS, H. A. A.; OMAR, M. Software project management approaches for monitoring work-in-progress: A review. **Journal of Engineering and Applied Sciences**, v. 12, n. 15, p. 3851-3857, 2017. ISSN 1816949X (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85029124622&doi=10.3923%2fjeasci.2017.3851.3857&partnerID=40&md5=eff47cec7d97b969ff8181d2734b9690> >. Acesso em: 22 out. 2017.

AL-SAKKAF, A. M.; HASHIM, N. L.; OMAR, M. Using hierarchical cluster analysis to generate clusters of agile practices. **Journal of Telecommunication, Electronic and Computer Engineering**, v. 9, n. 1-2, p. 53-56, 2017. ISSN 21801843 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020803914&partnerID=40&md5=b3cd49a886512bff887a56e6321d4c04> >. Acesso em: 22 out. 2017.

BECK, K. et al. Manifesto para Desenvolvimento Ágil de Software. 2001. Disponível em: < <http://agilemanifesto.org/iso/ptbr/manifesto.html> >. Acesso em: 02 abr. 2017.

BOWEN, J. P. et al. Formality, agility, security, and evolution in software development. **Computer**, v. 47, n. 10, p. 86-89, 2014. ISSN 00189162 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84908239815&doi=10.1109%2fMC.2014.284&partnerID=40&md5=8b494d63a3985becaead5a4e4bc022e0> >. Acesso em: 22 out. 2017.

CHOW, T.; CAO, D. B. A survey study of critical success factors in agile software projects. **Journal of Systems and Software**, v. 81, n. 6, p. 961-971, 2008. ISSN 01641212 (ISSN).

Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-42049093568&doi=10.1016%2fj.jss.2007.08.020&partnerID=40&md5=16460679eaedae763c64dc55442110> >. Acesso em: 22 out. 2017.

DINGSØYR, T. et al. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. **Empirical Software Engineering**, p. 1-31, 2017. ISSN 13823256 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85020136655&doi=10.1007%2fs10664-017-9524-2&partnerID=40&md5=85febfd2f043293a7a9b73bacb9b3cd7> >. Acesso em: 22 out. 2017.

FARID, A. B.; ABD ELGHANY, A. S.; HELMY, Y. M. Implementing Project Management Category Process Areas of CMMI Version 1.3 Using Scrum Practices, and Assets. **International Journal of Advanced Computer Science and Applications**, v. 7, n. 2, p. 243-252, Feb 2016. ISSN 2158-107X. Disponível em: < <Go to ISI>://WOS:000371515600035 >. Acesso em: 22 out. 2017.

**GUIA PMBOK**. 5 Edition - Portuguese. PMI, 2013.

KULKARNI, R. H. et al. Investigating agile adaptation for project development. **International Journal of Electrical and Computer Engineering**, v. 7, n. 3, p. 1278-1285, 2017. ISSN 20888708 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85021081669&doi=10.11591%2fijece.v7i3.pp1278-1285&partnerID=40&md5=1fd5b94a3a0d8d5b943c28144ce3ddf1> >. Acesso em: 22 out. 2017.

LAANTI, M. **Characteristics and principles of scaled agile**. Lecture Notes in Business Information Processing: Springer Verlag. 199: 9-20 p. 2014.

LEE, S.; YONG, H. S. Distributed agile: project management in a global environment. **Empirical Software Engineering**, v. 15, n. 2, p. 204-217, Apr 2010. ISSN 1382-3256. Disponível em: < <Go to ISI>://WOS:000275705900004 >. Acesso em: 22 out. 2017.

MAHNIC, V.; ZABKAR, N. Measuring Progress of Scrum-based Software Projects. **Elektronika Ir Elektrotehnika**, v. 18, n. 8, p. 73-76, 2012. ISSN 1392-1215. Disponível em: < <Go to ISI>://WOS:000310422400017 >. Acesso em: 22 out. 2017.

NUOTTILA, J.; AALTONEN, K.; KUJALA, J. Challenges of adopting agile methods in a public organization. **Ijispj-International Journal of Information Systems and Project Management**, v. 4, n. 3, p. 65-85, 2016. ISSN 2182-7796. Disponível em: < <Go to ISI>://WOS:000390947200005 >. Acesso em: 22 out. 2017.

PAPATHEOCHAROUS, E.; ANDREOU, A. S. Empirical evidence and state of practice of software agile teams. **Journal of Software-Evolution and Process**, v. 26, n. 9, p. 855-866, Sep 2014. ISSN 2047-7473. Disponível em: < <Go to ISI>://WOS:000342829400011 >. Acesso em: 22 out. 2017.

PINTO, E. B.; VASCONCELOS, A. M.; LEZANA, Á. G. R. ABORDAGENS DO PMBOK E CMMI SOBRE O SUCESSO DOS PROJETOS DE SOFTWARES. **Revista de Gestão e Projetos - GeP**, v. 5, n. 1, p. 55-70, Janeiro/Abril 2014. Acesso em: 24 jun. 2017.

PMI. **Agile Practice Guide**. 2017.

SUNDARARAJAN, S.; BHASI, M.; VIJAYARAGHAVAN, P. K. Case study on risk management practice in large offshore-outsourced Agile software projects. **IET Software**, v. 8, n. 6, p. 245-257, 2014. ISSN 17518806 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84919639376&doi=10.1049%2fiet-sen.2013.0190&partnerID=40&md5=4d4764f8f036c47feb55199b8afdca6> >. Acesso em: 22 out. 2017.

SURDU, J.; PARSONS, D. J. Army simulation program balances agile and traditional methods with success. **CrossTalk**, v. 19, n. 4, p. 4-8, 2006. Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-33745207562&partnerID=40&md5=b36c16dc1203031bfe2eedcf583fd63e> >. Acesso em: 22 out. 2017.

SUTHARSHAN, A. Enhancing Agile methods for multi-cultural software project teams. **Modern Applied Science**, v. 5, n. 1, p. 12-22, 2011. ISSN 19131844 (ISSN). Disponível em: < <https://www.scopus.com/inward/record.uri?eid=2-s2.0-79956299285&partnerID=40&md5=7e6d6f27545968c788b81f168de9a0e5> >. Acesso em: 22 out. 2017.

TRIPP, J. F.; RIEMENSCHNEIDER, C.; THATCHER, J. B. Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign. **Journal of the Association for Information Systems**, v. 17, n. 4, p. 267-307, Apr 2016. ISSN 1536-9323. Disponível em: < <Go to ISI>://WOS:000378143100002 >. Acesso em: 22 out. 2017.

VARGAS, L. M. PROJECT AGILE MANAGEMENT FOR SOFTWARE DEVELOPMENT: A COMPARATIVE STUDY ON THE APPLICABILITY OF SCRUM TOGETHER WITH PMBOK AND / OR PRINCE2. **Revista De Gestao E Projetos**, v. 7, n. 3, p. 48-60, Sep-Dec 2016. ISSN 2236-0972. Disponível em: < <Go to ISI>://WOS:000399121700004 >. Acesso em: 22 out. 2017.

WAUTELET, Y.; KOLP, M. e-SPM: An Online Software Project Management Game. **International Journal of Engineering Education**, v. 28, n. 6, p. 1316-1325, 2012. ISSN 0949-149X. Disponível em: < <Go to ISI>://WOS:000310724600010 >. Acesso em: 22 out. 2017. Acesso em: 22 out. 2017.