

## GERENCIAMENTO ÁGIL DE PROJETOS EM DESENVOLVIMENTO DE SOFTWARE: UM ESTUDO COMPARATIVO SOBRE A APLICABILIDADE DO *SCRUM* EM CONJUNTO COM *PMBOK* E/OU *PRINCE2*

### RESUMO

A gerência de projetos de desenvolvimento de *software*, em ambientes de constante transformação, demanda a utilização de métodos flexíveis de gestão de projetos. Os métodos ágeis surgiram como uma alternativa as abordagens tradicionais de gerenciamento de projetos, sendo baseado em uma série de doze princípios que tem como premissa a rápida resposta ao desenvolvimento, deixando documentação e o planejamento não essencial em segundo plano. Essa mudança de visão popularizou diferentes *frameworks*, todos em busca do desenvolvimento com qualidade e agilidade. O *Scrum* se destaca entre os gerentes de projeto ágeis por ter uma linguagem clara e passos definidos de implementação, podendo ser utilizado sozinho ou em conjunto com outros métodos. Nesse sentido o presente estudo busca comparar a compatibilidade entre o *Scrum* e dois métodos de gestão de projetos: *PMBOK* e o *PRINCE2*. Ambos métodos são originados do conhecimento empírico de inúmeros gerentes de projeto, e têm como objetivo serem guias de boas práticas para todo tipo de projeto, entretanto possuem peculiaridades que engessam o desenvolvimento na área da tecnologia da informação. A pesquisa tem como foco principal a gestão ágil de desenvolvimento de *software*, sendo o *Scrum* o ponto central da análise.

**Palavras-chave:** *Scrum*; Desenvolvimento Ágil de Software; *Prince2*; PMI; Gestão de Projetos

### PROJECT AGILE MANAGEMENT FOR SOFTWARE DEVELOPMENT: A COMPARATIVE STUDY ON THE APPLICABILITY OF *SCRUM* TOGETHER WITH *PMBOK* AND / OR *PRINCE2*

### ABSTRACT

The manages software development projects, in constantly changing environments demand the use of flexible methods of project management. Agile methods have emerged as an alternative to traditional approaches to project management, being based on a series of twelve principles is premised on rapid response to development, leaving documentation and nonessential planning in the background. This change in vision popularized different frameworks, all seeking development with quality and speed. *Scrum* stands out among the agile project managers to have clear language and defined steps of implementation and may be used alone or in conjunction with other methods. In this sense the present study seeks to compare the compatibility between *Scrum* and two project management methods: *PMBOK* and *PRINCE2*. Both methods are originated from the empirical knowledge of numerous project managers, and are intended to be guides to good practice for all kinds of projects, but have quirks that imprison development in the area of information technology. The research has mainly focused on agile management of software development, *Scrum* being the focal point of analysis.

**Keywords:** *Scrum*; Agile Software Development, *PRINCE2*; PMI; Project Management.

Letícia Marques Vargas<sup>1</sup>

---

<sup>1</sup> Mestranda do Programa de Pós-Graduação em Desenvolvimento Territorial e Sistemas Agroindustriais da Universidade Federal de Pelotas - UFPel. Brasil. E-mail: [le.mvargas@gmail.com](mailto:le.mvargas@gmail.com)

## 1 INTRODUÇÃO

Um dos desafios na área de desenvolvimento de *software* está na complexidade da compreensão das expectativas do cliente através da análise dos problemas e da criação de soluções que efetivamente atendam aos anseios e agreguem valor. Sendo o *software* “um produto desenvolvido por meio de padrões de qualidade e que possui uma documentação agregada, podendo ser composto por um ou vários programas<sup>2</sup>” (Miletto & Bertagnolli, 2014, p.4), a busca por ferramentas que proporcionem o desenvolvimento eficiente e de qualidade perpassa, obrigatoriamente a área de gestão de projetos.

Segundo Meredith e Mantel (2000), a literatura da área de engenharia de *software* trás diversas discussões a respeito dos fracassos resultantes da omissão ou má aplicação de metodologias em torno do desenvolvimento de *software*. Nesse sentido, os autores constataram que uma considerável parcela dos problemas relacionados aos projetos são derivados de má administração ou de mau gerenciamento do processo de desenvolvimento do *software*.

Atrelado as questões de gerenciamento de projeto, há uma forte demanda mercadológica por agilidade, qualidade e eficiência dos *softwares*. Segundo Vitiello (2001), essa busca constante por rapidez demanda da equipe de desenvolvimento uma organização dos projetos de forma a atingir os objetivos de produção no menor tempo e com a melhor qualidade possível.

Nesse sentido, Schwalbe (2014) afirma que a gestão de projetos consistira no bom emprego de um conjunto de conhecimento, de habilidade e de ferramentas com o objetivo de organizar as atividades de um projeto e desta forma atender as expectativas do cliente final. Assim, destaca-se que o gerenciamento de projetos não é uma tarefa simples, pois deve atender não só as especificações do escopo do projeto, mas também as expectativas dos envolvidos e a qualidade esperada (Graham, 2009).

Frente a complexidade da gestão de projetos, Maximiliano (2008) ressalta que projetos são iniciativas transitórias, com atividades claras e definidas, criadas para atender/resolver algum problema, oportunidade ou interesse de uma pessoa ou organização. Esta transitoriedade faz parte da natureza de um projeto, visto que a incerteza está presente na unicidade de cada projeto e, em alguns casos, na dificuldade de determinação dos objetivos, dos prazos de execução e/ou dos custos do sistema como um todo. Neste contexto, a gestão de um

projeto auxilia na prevenção de situações de risco, permitindo um controle maior do que está sendo desenvolvido e antecipando as situações desfavoráveis. De acordo com Vargas (2000), os benefícios da boa gestão de projetos são notados em diferentes setores, pois a organização permite a tomada de decisões mais ágeis e acertadas sobre o negócio e o mercado.

Na área da tecnologia, a gestão de projetos deve considerar que o desenvolvimento de *software* é um processo essencialmente criativo, que reúne profissionais com talentos e conhecimentos diferenciados tendo como objetivo a construção de um produto de valor (Sommerville, 2011). As dificuldades da gestão de *software* surgiram na década de 60 quando o fracasso de inúmeros projetos através da entrega atrasada e de produtos de baixa qualidade denunciaram as falhas existentes nas abordagens tradicionais utilizadas (Sommerville, 2003). Frente as especificidades da área de tecnologia, a gestão de projetos foi desafiada a ampliar e a adaptar suas práticas para melhoria da produtividade e da qualidade dos produtos.

Na literatura, há o embate entre duas grandes vertentes, a primeira ancorada nos princípios clássicos de gerenciamento de projetos e a segunda focada nos métodos ágeis de desenvolvimento. O objetivo de todos é o mesmo: garantir a conclusão 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.

Frente ao exposto surge a necessidade de conhecer os processos e as práticas ágeis, em específico o *Scrum*, e os métodos tradicionais de gestão de projetos, *PMBOK* e *PRINCE2*, para realizar uma análise comparativa dos pontos de divergência e convergência quando aplicados em conjunto para a gestão de projetos de desenvolvimento de *software*. Através da identificação das características principais do *PMBOK*, do *PRINCE2* e do *Scrum*, pretende-se apresentar a gestão de projetos e o gerenciamento ágil sob uma ótica voltada para as peculiaridades do mercado da tecnologia da informação, ressaltando os pontos de adaptação dos métodos de gestão de projetos ao serem aplicadas com um *framework* ágil (*Scrum*).

## 2 SCRUM

A estrutura do método *Scrum* teve sua origem no estudo publicado por Hirotaka Takeuchi e

programação, cujo objetivo é resolver um problema.” (Miletto & Bertagnolli, 2014, p. 4)

<sup>2</sup> “Programa tem um conceito mais amplo, podendo ser definido como um conjunto de linhas de código (ou algoritmo executável) em determinada linguagem de

Ikujiro Nonaka (1986) no qual são descritos seis elementos que agrupados “produzem um conjunto dinâmico” adotado para o gerenciamento de projetos de empresas referência de mercado. O método de gerenciamento apresentado pelos autores utilizou o *rugby* como analogia para exemplificar a forte relação entre os membros da equipe no desenvolvimento contínuo e cíclico proposto pelo método. De acordo com Silva e Lovato (2016), a denominação *Scrum* aparece em um segundo momento através dos autores DeGrace e Stahl, mas somente é consolidada a partir da década de 1990 nos estudos realizados por Schwaber e Sutherland. Segundo estes autores, o *Scrum* deve ser: leve, simples de entender e extremamente difícil de dominar, visto que é um *framework* cunhado para desenvolver e manter projetos complexos, tais como de desenvolvimento de *software*, permitindo diferentes processos e técnicas durante suas práticas de desenvolvimento adaptativo, produtivo e criativo (Schwaber & Sutherland, 2013).

Nesse sentido Vieira (2014, n.p) afirma que assim como outros *frameworks*, o *Scrum* pode ser compreendido como a fundação e as paredes de um edifício, os quais não podem ser fundamentalmente alterados sem o risco de um colapso, é possível adicionar artefatos e recursos customizados para cada organização e elaborar processos próprios desde que seja mantida a estrutura principal.

Assim, conforme Machado e Medida (2009) a adoção do *Scrum* deve vir acompanhada de uma mudança cultural, de maneira a permitir total implementação das características do *framework*. Nesse sentido, o manifesto ágil (Beck et al, 2001) se apresenta como marco teórico do acultramento da mudança de paradigmas de desenvolvimento de *softwares*. De acordo com os autores do manifesto ágil (Beck et al, 2001), este documento foi escrito com o objetivo de consolidar as práticas ágeis na área de tecnologia e dar suporte a todos os profissionais que veem os princípios do pensamento ágil um caminho para o desenvolvimento com qualidade e rapidez. Segundo Beck et al (2001), os valores, os princípios e as práticas do manifesto ágil devem ser internalizadas em conjunto com o *framework* por todos componentes da equipe, de acordo com Brod (2013, p. 33) o manifesto ágil deve ser “tatuado” pela equipe de maneira a naturalizar o comportamento.

Nesse sentido, o *Scrum* apresenta-se como método essencialmente ágil de gerenciamento de projetos, tendo em sua gênese a busca constante pelo melhoramento das práticas estruturais e pelo encorajamento da adaptação constante dos componentes desde que não prejudique a essência do método (Schwaber & Sutherland, 2013).

### 3 *PMBOK*

O *Project Management Body of Knowledge* é um guia de boas práticas em gerenciamento de projetos elaborado e escrito pelos voluntários do *Project Management Institute (PMI)*. O Instituto de Gerenciamento de Projetos é reconhecido mundialmente como instituição responsável para o desenvolvimento de padrões para sua área de atuação (*PMBOK*, 2013).

Segundo Kardec (2012), o *PMI* é o maior difusor dos processos que envolvem o desenvolvimento de projetos, o padrão estabelecido pelo guia de gerenciamento de projetos da *PMI* descreve normas, métodos, processos e práticas mundialmente reconhecidas. Assim, segundo os autores do guia (*PMBOK*, 2013), as boas práticas contidas no documento são fruto da evolução natural da profissão de gerente de projetos e das contribuições dos mesmos para ao desenvolvimento da área.

Nesse sentido, Cruz (2013) afirma que de acordo com o *PMI*, este guia pode ser aplicado na maioria dos projetos, independente da área ou do tempo de duração, os conhecimentos descritos no guia devem ser aplicados de forma padronizada em todos os projetos de maneira a permitir sua compreensão por todos envolvidos ou não em todo o processo.

Assim, as diretrizes estabelecidas pelo guia têm por objetivo descrever a estrutura, a relação entre os processos e a interação entre agentes em um projeto. De acordo com ao *PMBOK* (2013), os processos são reunidos em cinco grupos com interfaces bem definidas que permitem o relacionamento organizado e documentado conforme as regras do guia. Todos os processos se agrupam em dez áreas de conhecimento que serão utilizadas conforme a necessidade do projeto.

Conforme Fernández, Garrido, Ramírez e Perdomo (2015), esta organização dos processos em grupos e áreas de conhecimento é o que caracteriza do *PMBOK*, tendo sido escrito a partir das experiências positivas e negativas dos voluntários autores do *PMI*, é considerado pela literatura como um método tradicional e burocrático de gerenciamento de projetos.

### 4 *PRINCE 2*

Segundo Fernández, Garrido, Ramírez e Perdomo (2015) o *PRINCE2 (Projects In Controlled Environments - Projetos em Ambiente Controlado)* é um método de gestão de projetos *open source* concebido para se adaptar a diferentes tipos de projeto. A característica *open source* está relacionada a possibilidade de desenvolvimento colaborativo e incremental de *softwares* de código

aberto sendo uma alternativa a indústria de tecnologia (Canaltech, 2015, n.p.)

De acordo com o documento oficial do *PRINCE2* (2009), a flexibilidade do método agregada aos princípios constituintes do *framework*, permite a aplicação do método de gerenciamento independente da escala ou amplitude geográfica de trabalho, além do crescimento da maturidade organizacional. Essa abordagem visa integrar custos, tempo, qualidade, escopo, riscos e benefícios permitindo o gerenciamento focado na obtenção dos melhores resultados para o projeto.

Assim, o *PRINCE2* apresenta-se como um método de gestão de projetos universal, auto validado pelos anos de prática dos colaboradores e capaz de se adaptar as necessidades dos gestores de projeto ao fornecer ferramentas flexíveis de gestão (Fernández, Garrido, Ramírez, & Perdomo, 2015).

De acordo Cruz (2015), o gerenciamento oferecido pelo método *PRINCE2* considera a temporalidade dos projetos como base para a necessidade do uso de um ambiente controlado e orientado a entregas, diminuindo assim as incertezas geradas pelas peculiaridades de cada *Business Case*.

Segundo Jamali e Oveisi (2016), o *PRINCE2* e o *PMBOK* possuem os mesmos fundamentos estruturais, buscando diminuir riscos e aumentar a qualidade através de uma série de regras e processos. Entretanto o *PRINCE2* tem a peculiaridade de ter sido elaborado com o propósito de uso em projetos voltados para o ambiente de tecnologia, além de ter uma prevalência de uso nos países europeus.

## 5 METODOLOGIA

Podemos definir pesquisa como sendo um processo reflexivo, ordenado e crítico que auxilia na descoberta de novos fatos e/ou soluções, em diferentes áreas do conhecimento, sendo uma atividade realizada por meio do processo do método científico (Rampazzo, 2005). Dessa forma, percebemos que toda pesquisa possui uma intenção construindo o conhecimento a respeito de uma realidade-objeto do pesquisador. Dito isso, esse capítulo tem como proposta demonstrar os caminhos metodológicos dessa pesquisa.

A natureza do presente estudo é qualitativa, com o objetivo de melhor responder as questões levantadas acerca do tema de pesquisa que não é quantificado e não pode ser reduzido à operacionalização de variáveis (Chizzotti, 2003). O presente estudo se configura como qualitativo pela flexibilidade na interpretação, tradução e aprofundamento das questões referentes ao tema (Miles & Huberman, 1994).

Este estudo ainda pode ser classificado como bibliográfico e documental de caráter

exploratório e descritivo, uma vez que a pesquisadora busca compreender as dimensões de cada método e realizar associações e comparações sobre a aplicabilidade e adaptabilidade dos métodos tradicionais ao método ágil (Severino, 2007).

O processo de coleta de dados ocorreu em um primeiro momento nas documentações oficiais de cada um dos métodos escolhidos para a pesquisa. Logo após foram realizadas pesquisas de literatura sobre o tema de maneira a conhecer o campo de estudo. Cabe ressaltar que durante o processo de coleta de dados algumas análises já eram realizadas devido ao caráter comparativo da pesquisa.

Para fins de análise, optou-se por uma comparação qualitativa entre os métodos de gestão de projetos de desenvolvimento de *software*, tendo como objetivo principal verificar a compatibilidade da aplicação do *Scrum* como *PMBOK* e *PRINCE2*.

## 6 COMPARANDO PADRÕES DE GESTÃO DE PROJETOS

Retomando as palavras de Sommerville (2011, p.2) “os sistemas de *software* são abstratos e intangíveis”, e demandam um exercício criativo para seu desenvolvimento. Segundo o autor, deve existir uma preocupação no sentido de não bloquear a equipe criativa com a utilização de métodos de gerenciamento de projetos que burocratizam o andamento da atividade limitando e prejudicando a criatividade esperada na área de desenvolvimento de *software*.

Nesse sentido, Schwaber & Sutherland (2013) afirmam que o *Scrum* estabelece um conjunto de práticas e conceitos que se encaixam em projetos que necessitam de flexibilidade, adaptabilidade e agilidade. Segundo os autores, estas características tornam o *framework* um grande aliado durante o desenvolvimento de *softwares*, possibilitando a entrega de valor em tempos menores e dentro dos padrões de qualidade desejados. Como fragilidade do *Scrum*, os mesmos autores evidenciam o foco quase exclusivo nos processos de desenvolvimento do projeto, deixando áreas complementares ao projeto enfraquecidas quando não administrado pelos gestores.

Assim Kardec (2012) escreve que as lacunas existentes no *Scrum* podem ser supridas ao se utilizar outro método de gestão de projetos que equilibre as fraquezas e as qualidades do método, sem perder o foco quando a flexibilidade e adaptabilidade dos processos de desenvolvimento de *software*. Frente a esta necessidade de busca pelo equilíbrio entre o *Scrum* e outros métodos de gerenciamento de projetos, a comparação entre o *Scrum* e os métodos *PMBOK* e *PRINCE2* se torna essencial para análise da possibilidade de união de

um ou de outro (ou de ambos) ao *framework* para a gestão do desenvolvimento de *software*.

**a. Scrum x PMBOK**

De acordo com o livro *PMBOK* (2013), este método trabalha com vários grupos de processos que abrangem todas as fases de um projeto, escrito de forma genérica, descrevendo as boas práticas de gerenciamento sem determinar a área de conhecimento para sua aplicação.

Nesse sentido, Cruz (2013) afirma que o guia *PMBOK* não descreve “como fazer” projetos e sim descreve “o que deve ser feito” através de processos que abrangem diversas áreas de conhecimento. Este descritivo de boas práticas tem

por objetivo flexibilizar os processos de gerenciamento permitindo a equipe determinar o grau de profundidade a ser atingido conforme a complexidade do projeto, dessa forma as peculiaridades de cada área de conhecimento é alcançada pelo método. O autor ressalta a importância de compreender o *PMBOK* como um método e não uma metodologia, de maneira a desmistificar errônea ideia de burocratização de projetos com o uso deste método.

Desta forma, a perspectiva de análise adotada no presente estudo considera que o guia *PMBOK* se propõe a ser um manual de boas práticas a ser executado em conjunto com uma metodologia de gestão, sendo a última responsável pelo engessamento ou não dos projetos.

**Tabela 1 - Comparação *PMBOK* x *SCRUM***

<i>PMBOK</i>	<i>SCRUM</i>
Padrão de gerenciamento de projetos	<i>Framework</i> de trabalho ágil
Foco no planejamento	Foco no produto
Documentação é o principal instrumento de controle. Toda etapa do projeto deve ser documentada é fator impeditivo para a mudança de etapa. A documentação é extensa e desenvolvida pelo gerente de projetos.	Documentação produzida conforme os ciclos são realizados. O foco do trabalho está na implementação do produto com qualidade, a documentação é secundária. Existe uma flexibilidade na produção da documentação. A documentação é simples, resumida e com linguagem amigável tanto para o cliente quanto para o time de desenvolvimento.
Necessita de um número maior de certezas para estabelecer as métricas de trabalho (custo, dimensionamento de equipe e tempo).	Trabalha com incertezas e com prioridades durante o desenvolvimento do produto.
O conceito utilizado é de escopo claro e detalhado do que deverá ser realizado no projeto. Devem ser descritos os elementos principais que servem de base para as estimativas de custo e tempo.	Utiliza-se um conceito de abstração de funcionalidades, estabelecendo um escopo inicial que servirá de base para o detalhamento no decorrer dos ciclos de vida.
Mudanças são consideradas riscos ao sucesso do projeto, não são impedidas, mas são permitidas apenas se, após análise de impacto e custos, os benefícios forem superiores aos custos de alteração de escopo.	As mudanças são consideradas inerentes ao processo de amadurecimento da equipe e devem ser implementadas no próximo ciclo iterativo possível.
Os custos são estimados com base do escopo e análise dos requisitos do projeto. Existem processos específicos que gerenciam os custos do projeto. A inexistência do escopo inviabiliza a estimativa de custo do projeto.	Os custos são estimados através do <i>product backlog</i> , agilizando o início da execução pois os detalhamentos são realizados a cada iteração. Não há um processo específico que trate do assunto.
Orientado a processos: processos bem definidos devem ser executados obrigatoriamente para garantir o desenvolvimento de qualidade. O gerente deve distribuir as atividades e montar o cronograma conforme os talentos das pessoas envolvidas no projeto.	Orientado a pessoas: as pessoas são tratadas como indivíduos e não recursos. O ritmo de trabalho é respeitado, cada membro da equipe escolhe quais e estimam o tempo a ser gasto das tarefas que irá desenvolver na próxima iteração.
Burocrática: exigem o desenvolvimento da solução completa, gera sobrecarga da equipe e compromete a velocidade de desenvolvimento.	Simplicidade: parte do princípio que realizar algo simples e útil é melhor do que gastar tempo em algo complexo que poderá não ser utilizado.

Equipes maiores, selecionados conforme a estrutura da empresa. Há obrigatoriamente um gerente de projetos como líder. Os membros são alocados no projeto conforme a necessidade organizacional, podendo ter dedicação parcial ou total dos envolvidos, além de ser modificada durante a execução do projeto. Não há restrição quanto a formação de equipes geograficamente separadas.	Equipes pequenas, multifuncionais, sem distinção de títulos profissionais, e, enquanto for economicamente viável, devem permanecer iguais durante todo o projeto. Não há liderança designada, são auto organizáveis. O <i>Scrummaster</i> e o <i>product owner</i> não são considerados no número de membros da equipe pois podem participar de mais de um projeto. Preferencialmente não utiliza equipes virtuais.
São definidos três modelos de ciclo de vida: preditivo, iterativo/incremental e adaptativo. Os dois últimos são mais utilizados e permitem o planejamento detalhado em ondas, o escopo é planejado em uma visão macro no início do projeto e pormenorizado conforme as etapas de desenvolvimento se repetem.	Ciclo de vida baseado em <i>sprints</i> , o projeto tem seu escopo detalhado ao longo do desenvolvimento. As reuniões de planejamento e revisão da <i>Sprint</i> definem o detalhamento do escopo do período e as possíveis mudanças necessárias. O ciclo tem duração fixa de 2 a 4 semanas e é um evento <i>timeboxed</i> .
O processo de gerenciamento de qualidade do processo deve ser aplicado a todo o projeto e garantir que os requisitos sejam cumpridos e validados com a melhor qualidade possível. Os padrões de qualidade devem ser compatibilizados com as normas ISO.	A qualidade está em atender o cliente através do cumprimento dos requisitos e do <i>feedback</i> constante do cliente ao longo do projeto. A cada ciclo são retornados os pontos de melhoria que devem ser trabalhados o mais breve possível. O foco está no produto com a melhor qualidade possível.
A comunicação é planejada através da identificação das necessidades das partes e determinação dos meios adequados. Devem ser gerados documentos formais de comunicação para registrar e divulgar as conclusões as partes interessadas.	A comunicação é interpessoal, realizada nas reuniões de planejamento e revisão da <i>Sprint</i> , e reuniões diárias. São reuniões objetivas, rápidas e com pauta preestabelecida.
O cliente está presente em poucos momentos de planejamento do escopo ou de gerenciamento de riscos.	Participação ativa do cliente ( <i>product owner</i> ) como parte do projeto.
Ao final do projeto é realizada reunião para arquivamento de documentos e retomada das lições aprendidas durante o projeto.	O processo de retomada das lições aprendidas é realizado ao final de cada <i>Sprint</i> durante a reunião de retrospectiva.
Gerenciamento de aquisições	Não há

*Nota.* Informações compiladas pela autora, 2015.

O primeiro ponto de distinção apresenta-se na classificação dos métodos pela literatura, segundo o guia *PMBOK* (2013) o mesmo é um padrão para gerenciamento de projetos, que reúne as boas práticas de gestão baseadas na experiência de profissionais renomados da área, enquanto Schwaber e Sutherland (2013) classificam o *Scrum* como um *framework* para o desenvolvimento de projetos.

O segundo ponto de distinção que é destacado pela literatura é relativo a documentação. De acordo com o *PMBOK* (2013), a documentação e o controle são elementos essenciais para o bom andamento de um projeto. Em contrapartida, o *Scrum* (Schwaber & Sutherland, 2013) visa um trabalho menos documentado que registre apenas os elementos realmente importantes para a compreensão do projeto. Nesse sentido ao trabalhar com o método *PMBOK* temos como obrigatório o termo de abertura e o escopo do projeto, os quais quando ausentes impedem o início das atividades de desenvolvimento (Cruz, 2013). Já o *Scrum* pode se valer deste documento para uma pequena formalização e registro para comparação ao final do projeto, Cruz (2013) destaca que o *Scrum* presa pela

comunicação verbal e documentação simples. O escopo e planejamento são elaborados ao longo do processo, e o projeto pode ser iniciado utilizando os requisitos desenvolvidos na análise inicial em conjunto com o cliente.

Seguindo as distinções, Kardec (2012) comenta que o custo do projeto no *PMBOK* deve ser calculado mediante a formalização do escopo e análise dos requisitos, em contrapartida o *Scrum* trabalha com o *product backlog* para a estimativa de custos e distribuição das atividades e responsabilidades. São diferenças que afetam diretamente o cronograma de execução, e em projetos de cunho tecnológico, prazo é muito importante. Cruz (2013) alerta que um projeto de desenvolvimento de *software* demanda adaptações constantes e a exigência demasiada de documentos pode atrasar o cronograma de execução, entretanto a falta desta também pode ser negativa e levar a desorganização do projeto. O autor ainda destaca a flexibilidade da documentação para iniciar o projeto como característica seminal do *Scrum* e permite a organização da equipe e das responsabilidades do projeto de maneira mais ágil, enquanto de acordo

com o guia *PMBOK* (2013) o termo de abertura é elemento obrigatório para o início das atividades.

Segundo Cruz (2013), o escopo é um dos elementos centrais e mais difíceis de serem gerenciados em projetos de *software* devido à natureza dinâmica da área da tecnologia. Assim, de acordo com o *PMBOK* (2013), o escopo deve ser definido durante o planejamento do projeto e aprovado pelo cliente, mudanças no escopo só podem ser realizadas através de um processo de gerenciamento integrado de mudanças que verifica o binômio custo x benefício para determinar sua implementação. Em contrapartida, o *Scrum* utiliza um controle iterativo de escopo. A cada Sprint são revisados os requisitos e definidas as funcionalidades que serão desenvolvidas. O planejamento em ciclos permite a adequação do escopo e implementação das mudanças de forma ágil. A flexibilidade no planejamento do escopo apresenta-se como característica necessária na área de desenvolvimento de *software*, e nesse sentido o método ágil considera a mudança como elemento natural no processo de amadurecimento da equipe (Schwaber & Sutherland, 2013). Para o *PMBOK* (2013), o escopo deve sofrer o mínimo de mudanças possíveis, e caso seja necessário estas devem ser planejadas e implementadas após uma análise do impacto financeiro realizada por um, confirmação de que os benefícios serão maiores do que os custos e replanejamento do projeto.

Outro ponto de diferenciação entre os métodos está relacionado a formalização das alterações realizadas ao longo do projeto. No método do *PMI*, as mudanças devem ser registradas por escrito no sistema de gerenciamento de mudanças, a documentação é analisada de forma integrada com o projeto por um comitê de controle de mudanças. Toda mudança deve ser acompanhada de “novas ou revisadas estimativas de custos, sequenciamento de atividades, datas de cronograma, requisitos de recursos e análise de alternativas de resposta aos riscos” (*PMBOK*, 2013, p. 96). O processo é descrito no guia *PMBOK* com fases bem definidas e necessárias para permitir a alteração de algum elemento do escopo. O foco *PMBOK* é entregar o produto planejado no início do projeto.

Ao trabalhar com *softwares* a agilidade deve prevalecer, a mudança faz parte do amadurecimento dos envolvidos com o projeto. Conforme o projeto avança novos requisitos são conhecidos e novas soluções são. O excesso de formalismo tende a tolher a criatividade da equipe e limitar a expansão do *software*.

A equipe de desenvolvimento é um dos elementos centrais para o sucesso de um projeto *Scrum*. Ao trabalhar com o auto-organização dos membros o *framework* permite que naturalmente surjam lideranças nos momentos em que esta for necessária, assim fica a cargo dos membros do time

a definição de um líder ou não para o projeto. O *PMBOK* trabalha com a designação de liderança desde o início do projeto, independente da estrutura hierárquica da organização o gerente de projetos é determinado como líder do time e deve se manter nesta posição durante todo o período de execução do projeto.

Em relação ao tamanho e formação da equipe, no *Scrum* preza-se pela manutenção dos membros enquanto for economicamente viável, ao se iniciar um projeto deve-se procurar manter o time de desenvolvimento até o término do projeto. Essa prática favorece o amadurecimento dos envolvidos e melhora a prática diária de trabalho, melhorando os tempos de desenvolvimento e diminuindo os prazos de entrega. Já no *PMBOK*, o time de desenvolvimento é dinâmico, as pessoas são alocadas conforme sua necessidade ao longo do projeto. Esta modificação é realizada com base na ociosidade dos envolvidos, um profissional especialista pode ser necessário apenas em um determinado ponto do projeto, desta forma será alocado para o trabalho de forma transitória. Esta prática visa utilizar especialistas em várias frentes simultaneamente e diminuir custos com ociosidade.

O ciclo de vida de um projeto *Scrum* é um evento *timeboxed* com um horizonte de tempo entre duas e quatro semanas. O time de desenvolvimento define este prazo no início do projeto e deve mantê-lo até o final. O ciclo é iterativo, com eventos sucessivos e ininterruptos de planejamento, desenvolvimento, revisão e retrospectiva.

No *PMBOK* são apresentados três modelos de ciclo de vida, sendo um primeiro preditivo utilizado quando se conhece o produto a ser entregue e cada etapa do projeto é realizada apenas quando a anterior é finalizada, o segundo iterativo que tem algumas fases do projeto repetidas conforme a equipe compreende melhor o projeto e um terceiro ciclo tido como adaptativo que se assemelha ao *Scrum* por utilizar um envolvimento maior do cliente com entregas pequenas a cada duas ou quatro semanas. O ciclo de vida adaptativo foi adicionado ao *PMBOK* na última edição do guia (2013), e vai ao encontro do modelo de ciclo de vida do *Scrum*. Importante ressaltar que não são todas as etapas do *PMBOK* que iterativas e sucessivas, o planejamento do escopo poderá ser em ondas desde que o escopo macro do projeto tenha sido planejado ao início do projeto para permitir os processos de gestão de custos e cronograma (Cruz, 2013).

A qualidade é elemento importante para o *PMBOK* e para o *Scrum*. O cumprimento dos requisitos do escopo com a melhor qualidade possível é buscado por ambos métodos, o foco de trabalho é que diferencia. Para o *PMBOK* o planejamento das ações de desenvolvimento e a realização de testes contínuos permitem a entrega de um produto com a qualidade esperada. O padrão ISO

é utilizado pelo *PMBOK* como padrão a ser atingido pelo projeto. No *Scrum*, o *feedback* do cliente é o foco da busca pela qualidade do produto. Ao longo do desenvolvimento os retornos do cliente são os responsáveis pela melhoria contínua do produto. O foco do *Scrum* está em suprir as necessidades do cliente com qualidade, mesmo que isso signifique uma mudança de escopo.

A comunicação é essencial para o sucesso do projeto. No *PMBOK* toda comunicação deve ser transformada em documento formal para controle e registro dos assuntos tratados. O *Scrum* preza pela objetividade e agilidade nas comunicações, sendo a personalidade um elemento de muita força. Em se tratando de equipes geograficamente próximas, o ato de “levantar e ir até a mesa do colega” deve ser uma constante, evitar e-mails quando se pode falar pessoalmente e aproveitar ao máximo as reuniões de equipe, sempre em busca da comunicação efetiva e clara (Cruz, 2013).

Outro ponto importante de comparação é a figura do cliente do projeto. No *Scrum* temos o cliente representado pelo *product owner*, tendo papel ativo nas reuniões de planejamento e revisão da *Sprint*. O *feedback* constante é atividade primordial para o sucesso do projeto e fica a cargo do *product owner*, sem a participação do cliente um projeto baseado em *Scrum* tende a não atingir a qualidade esperada. No *PMBOK* esta interação é mais restrita, sendo as expectativas gerenciadas pela equipe do projeto e comunicadas ao cliente em momentos específicos. O escopo e o termo de abertura do projeto depois de elaborados em conjunto com o

cliente são os documentos que nortearão o desenvolvimento e busca pela qualidade do produto.

O aprendizado é importante para ambos métodos. No *Scrum* vemos a reunião de retrospectiva de *Sprint* como principal momento de retomada das lições aprendidas. No *PMBOK* esta retomada é realizada ao final do projeto, durante os processos de encerramento. A retomada a cada *Sprint* permite a melhoria de forma mais rápida e dentro do mesmo projeto, agilizando processos e focando na qualidade do produto final. Ao se trabalhar com essa retomada apenas no final do projeto pode-se perder oportunidades de melhoria durante o projeto e gerar um retrabalho. São posicionamentos que devem ser avaliados a cada projeto, mas que afetam o desenvolvimento e a agilidade da resolução de problemas e melhoria contínua.

O *PMBOK* prevê o gerenciamento das aquisições necessárias ao projeto, devem ser planejadas as aquisições de serviços e produtos necessários para o bom andamento do desenvolvimento. O *Scrum* não menciona como deve-se proceder no caso de ser necessária a aquisição de algum elemento.

#### b. *Scrum* x *PRINCE2*

O *PRINCE2* tem uma abordagem que se mescla com outros métodos pois sua estrutura não discorre sobre como serão os métodos de entrega do produto. Dessa maneira as diferenças entre o *PRINCE2* e o *Scrum* em muitos momentos se tornam complementos para uma gestão de projetos com qualidade.

Figura 1 - Posição do *Scrum* na estrutura *PRINCE2*



Fonte: De “Agile Business Case Management” de W. Martin, 2010, [Blog]. Retrieved from <http://www.borselaer.org/index.php/2010/03/whitebook-PRINCE2-combined-with-Scrum>

Segundo Martin (2010), através da figura 1 é possível verificar o ponto de convergência entre os métodos *Scrum* e *PRINCE2* como sendo na fase de entregas do projeto. Assim, ao realizar a integração entre os métodos, todo gerenciamento das entregas passa a ser realizado utilizando-se o *Scrum* enquanto

as demais áreas de gerência ficam sob responsabilidade do *PRINCE2*.

Seguindo esta linha de pensamento de Martin (2010), outros pontos de convergência e de divergência são encontrados entre os métodos *PRINCE2* e *Scrum*, e apresentados na sequência.

**Tabela 2 - Comparação *PRINCE2* x *SCRUM***

<i>PRINCE2</i>	<i>SCRUM</i>
Método de gerenciamento de projetos	<i>Framework</i> de trabalho ágil
Foco em questões mais amplas do projeto como riscos, organização, equipe. Por ser um método de gerenciamento de projetos, seus princípios norteiam todos aspectos envolvidos.	Foco no desenvolvimento incremental do produto e nas entregas eficientes.
O conjunto de benefícios percebidos pelo cliente após a conclusão do projeto é o que define o sucesso de um projeto.	O sucesso de um projeto é definido pela entrega com qualidade, cumprindo todos os requisitos e necessidades do cliente.
Abordagem preditiva (baseada no plano)	Abordagem adaptativa
Crítérios de aceitação são fixados ao iniciar o projeto e devem permanecer independente das mudanças realizadas ao longo do desenvolvimento.	As mudanças são consideradas naturais ao processo de desenvolvimento. A flexibilidade e a colaboração do cliente ao longo do desenvolvimento são próprias do <i>framework</i> .
Equipe organizada pelo gerente de projetos, com atividades distribuídas e monitoradas pelo gerente.	Auto-organização da equipe, as atividades são disponibilizadas a todos e alocadas de acordo com o desejo do membro.
Utiliza três papéis com responsabilidades diferentes como representante do cliente: patrocinador de negócio, usuário final e fornecedor	Utiliza a figura do <i>product owner</i> como representante do cliente
Gerente de projetos responsável pela estrutura e aplicação do método ao projeto e possui autoridade sobre a equipe	<i>Scrummaster</i> é responsável pela aplicação do <i>framework</i> mas não tem autoridade sobre a equipe.
Inicia o projeto após uma fase de preparação e de início que estabelecem a base para o entendimento comum de todos os envolvidos sobre o <i>business case</i> , métodos e procedimentos a serem adotados.	Inicia o projeto com o <i>Product backlog</i> , uma abordagem de iniciar imediatamente e gerir obstáculos e mudanças ao longo dos ciclos.
Gerenciamento de riscos com a participação ativa dos representantes do cliente.	Os riscos são tratados durante as <i>sprints</i> .
Projeto planejado, executado e controlado por etapas.	Projeto desenvolvido em ciclos iterativos e sucessivos.
Princípio aprender com a experiência, na qual o time de desenvolvimento deve buscar aprender com os erros e acertos realizados em projetos anteriores e durante o projeto em execução.	Reunião de retrospectiva de <i>Sprint</i> que tem como objetivo resgatar os pontos positivos e negativos ocorridos no ciclo anterior para manter os acertos ou corrigir os erros realizados.
O projeto deve focar na entrega com qualidade do produto. E para isso a justificativa do negócio e os requisitos do produto devem ser bem delineados e compreendidos por todos envolvidos no projeto.	O projeto foca na entrega sucessiva e iterativa de um produto com qualidade. Para isso as reuniões de planejamento a cada <i>Sprint</i> devem ser realizadas com a presença do <i>product owner</i> para definições de requisitos e andamento do projeto.

*Nota.* Informações compiladas pela autora, 2015.

O *PRINCE2* apresenta um método de organização para a gestão e controle de projetos. São diretrizes voltadas para os gerentes de projeto para garantir a qualidade do produto ou serviço que será entregue. Em contrapartida, o *Scrum* é um

*framework* que orienta como desenvolver o produto ou serviço, através da definição e priorização dos requisitos, *feedback* contínuo e do ciclo iterativo e sucessivo.

No tocante ao foco principal, o *PRINCE2* baseia seu método em um conjunto de sete princípios que abrangem todos os aspectos de um projeto. Não está contemplado no manual a forma de trabalho das equipes, e sim um meio de comunicação entre o projeto e a equipe através de termos de elaboração de relatórios, prestação de contas e do trabalho em si. O *Scrum* tem o foco no desenvolvimento e entrega com qualidade, os ciclos incrementais e sucessivos de constante priorização têm o objetivo de desenvolver o que o cliente deseja, testar e realizar *feedback*.

O *Scrum* trabalha com a ideia que sucesso ao entregar produto com qualidade e atendendo os requisitos/necessidades do cliente. O sucesso no *PRINCE2* envolve não só a entrega com qualidade, mas também os benefícios percebidos pelo cliente.

A abordagem do *PRINCE2* é considerada preditiva, existe a tendência de projetos que utilizem este método se manterem mais focados nos objetivos de negócio originais não realizarem mudanças significativas no plano definido. Já o método ágil tem a adaptabilidade como essência. O *Scrum* trabalha com prazos curtos e planejamentos incrementais independentes, sendo sensíveis as mudanças do ambiente e as necessidades do cliente. Ressalta-se que a pré-disposição do *PRINCE2* em manter-se preso ao projeto inicial não significa que é uma abordagem fechada, apenas em comparação ao *Scrum* este método é menos suscetível a mudanças.

As modificações nos requisitos ao longo do projeto são tratadas como fruto do amadurecimento do time de desenvolvimento para o *Scrum*, ao iniciar um projeto sob a óptica ágil, a equipe sabe que haverá um relacionamento muito estreito com o cliente para alinhamento das necessidades ao início de cada *Sprint*. Em contraposição, o *PRINCE2* necessita que sejam fixados os critérios de aceitação do produto ao iniciar o projeto, qualquer alteração deve ser passar por um processo de análise e controle de mudanças visando resguardar o desenvolvimento de clientes inconsistentes. Entretanto, um relacionamento restrito entre time e cliente priva o projeto de atualizações rápidas frente a novidades no ambiente.

Em métodos orientados para planos, como no caso do *PRINCE2*, as equipes são organizadas pelo gerente de projetos tendo como critério as habilidades e/ou pontos fortes e/ou fracos de cada profissional. No *Scrum* a auto-organização é elemento crítico para o sucesso do *framework*. Todas as atividades do ciclo são disponibilizadas para o time e qualquer membro pode trabalhar em qualquer uma das tarefas utilizando como critério suas aptidões e tempo disponível. A auto-organização do *Scrum* favorece uma integração maior entre os membros do time e um maior investimento pessoal no projeto, enquanto o enfoque estruturado do

*PRINCE2* foca nas metas de eficiência e organização controlada.

Ambas abordagens trabalham com a figura do cliente envolvido no projeto. O *Scrum* determina o *product owner* como o responsável pela definição dos requisitos do produto do ponto de vista do cliente. No *PRINCE2* esta definição de requisitos tem a responsabilidade dividida em três papéis: patrocinador de negócio, usuário final e fornecedor. Esta divisão realizada pelo *PRINCE2* se adapta ao papel do *product owner* no sentido de que as reuniões de planejamento serão realizadas tendo apenas a presença do papel necessário no momento necessário de desenvolvimento.

O gerente de projetos, no *PRINCE2*, possui autoridade sobre a equipe de desenvolvimento sendo responsável pela organização e distribuição de atividades durante a execução do projeto. No *Scrum*, o *Scrummaster* é responsável apenas pela implementação das regras do *framework* e manutenção do ambiente favorável para o desenvolvimento do projeto. Os níveis de autoridade no *PRINCE2* são compartilhados (gestão por exceção), enquanto no *Scrum* a auto-organização deve prevalecer, e por este motivo é importante equilibrar os limites de controle e comando em ambas abordagens. O gerente de projetos e *Scrummaster* são responsáveis pelo processo e não pelos resultados.

Para o *Scrum* o início do projeto deve ser imediato, impedimentos e incertezas são gerenciadas ao longo do desenvolvimento do projeto. É considerado um movimento comum dentro de ciclos iterativos e sucessivos. Para o *PRINCE2* a fase de preparação e planejamento formam a base para a compreensão de todos os envolvidos sobre o *business case* do projeto, os métodos e os procedimentos.

Os riscos de um projeto são tratados de forma diferentes por cada uma das abordagens. Para o *Scrum* todo impedimento e mudanças são administrados a cada ciclo durante as reuniões de planejamento ou revisão. Toda *Sprint* é um ponto de controle para as mudanças dentro da esfera de influência do projeto. O *PRINCE2* possui uma abordagem de gestão de risco com participação ativa dos representantes do cliente, os esforços da equipe são seccionados conforme o grau de responsabilidade e esfera de competência do risco.

O ciclo de vida do projeto tem abordagens semelhantes no *PRINCE2* e no *Scrum*. No primeiro o princípio gerenciar por estágios determina que todo projeto deve ser dividido em pequenas etapas para melhor execução e gestão. No *Scrum*, o desenvolvimento através de *sprints* é a base do *framework*. Este é um ponto de convergência, havendo apenas uma diferença de nomenclatura.

Outro ponto de convergência é em relação ao aprendizado da equipe a cada projeto. O princípio

“Aprender com a experiência” do *PRINCE2* trabalha a ideia de que a curiosidade do grupo deve ser constante, a busca pela melhoria contínua só é completa se houver empenho de todos os envolvidos em aprender com os erros e sempre crescer a cada novo projeto. No *Scrum* temos este aprendizado incorporado nas reuniões de retrospectiva da *Sprint*, na qual o time de desenvolvimento retoma todos os pontos que foram positivos e negativos no ciclo anterior para manter ou corrigir o curso do projeto.

O foco na entrega de um produto com qualidade é fundamental para ambas abordagens. Para o *PRINCE2*, a delimitação correta da justificativa de negócio em conjunto com os requisitos de qualidade permitem que seja realizado um desenvolvimento de qualidade e consequentemente entregas que superem as expectativas dos clientes. No *Scrum* o produto é o foco do trabalho, através das reuniões cíclicas são definidas e executadas as atividades prioritárias para o cliente, prezando pela qualidade e agilidade.

## 7 CONSIDERAÇÕES FINAIS

O desenvolvimento de *software* enseja um processo diferente de gestão. A natureza complexa e inovadora de projetos ligados a tecnologia exige a utilização de ferramentas que acompanhem o mesmo nível de inovação sem burocratizar o trabalho ou inibir a criatividade inerente à área de tecnologia.

O gerenciamento ágil surge para suprir essa necessidade de gerencia sem burocracia, permitindo desenvolver com qualidade e de forma rápida. Os diferentes *frameworks* são derivados do Manifesto Ágil, tendo, portanto, o mesmo esqueleto e os mesmos objetivos. O *Scrum* se sobressaiu aos demais pela estrutura e eficiência dos resultados, tornando-se o mais popular no mundo.

A índole ágil do *Scrum* o torna naturalmente compatível com a complexidade dos projetos de *software* e a necessidade de planejar conforme o desenvolvimento caminha. Entretanto tal atitude pode não ser bem vista pela organização pois existe uma necessidade de controle enraizada em muitos gestores e consequentemente existir uma resistência ao uso do *Scrum* puro.

Nesse contexto, a compatibilização do *Scrum* com um método de gestão de projetos tradicional pode, além de facilitar a adoção do ágil, trazer benefícios importantes para a gestão de desenvolvimento de *software*. A escolha por uma das abordagens de gestão de projetos deve levar em consideração os pressupostos de cada uma.

A nova edição do *PMBOK* (5ª edição, 2013) trabalha elementos que convergem para o gerenciamento ágil, mas sua integração com os processos e áreas de conhecimento ainda necessitam

de refinamento e adaptação para plena utilização do método.

O *PMBOK* trabalha com um planejamento mais robusto e apesar de possibilitar o detalhamento do plano ao longo de ciclos iterativos, ainda peca ao restringir as mudanças a um processo de análise e processos próprios de implementação pois apenas parcela seu detalhamento. O desenvolvimento de *software* demanda um processo orientado a mudanças e planejamentos mais simples e rápidos que permitam manter a atualidade das tecnologias e, consequentemente, do produto final.

O controle centralizado das atividades é outro ponto no qual o *PMBOK* apresenta-se como entrave para o desenvolvimento em ambientes de constante transformação os quais exigem comunicação clara, rápida e eficiente. O *Scrum* promove esta comunicação entre os envolvidos, preza pela comunicação pessoalmente e reuniões rápidas e efetivas. Os registros e documentações exigidas pelo *PMBOK* tendem a extinguir a comunicação simples e pessoal e tornando a comunicação burocrática e impessoal com registros, aprovações e documentações de comunicação entre membros e liderança do projeto.

Ao analisar as abordagens *PRINCE2* e *Scrum* percebe-se que ambas trazem lacunas quando se trata de gestão ágil do desenvolvimento de *software*. Enquanto o *PRINCE2* foca no gerente de projetos para o planejamento, organização e controle, o *Scrum* trabalha com uma liberdade maior dos membros envolvidos visando tornar o ambiente mais ágil e criativo.

Entretanto a compatibilização entre o *PRINCE2* e o *Scrum* aparenta ser mais simples por se tratar de um método de gestão baseado em princípios, o qual trabalha questões necessárias para a implementação de um projeto, mas não explicita como se deve fazer esta gerencia, nesse ponto o *Scrum* traz o passo a passo de como proceder no desenvolvimento.

Através da literatura abordada e do estudo realizado que a complexidade da gerencia de desenvolvimento de *software* demanda um olhar diferenciado e adaptável. O sucesso de um projeto depende de seu planejamento, não só do projeto em si, mas em um nível organizacional com a adoção de práticas compatíveis com a filosofia da organização.

O método ágil deve se tornar parte da cultura organizacional, de forma a existirem times auto organizáveis de forma natural a cada projeto, com conhecimento técnico e experiência no trabalho em equipe. A gerência deve ser no sentido de facilitar o desenvolvimento, independentemente do método adotando: *PMBOK* ou *PRINCE2*.

Por se tratar de uma pesquisa qualitativa e bibliográfica há uma limitação natural frente ao universo literário existente sobre cada uma das abordagens. Em relação ao *PRINCE2* adiciona-se a

limitação da bibliografia em português, ainda pequena e com problemas de tradução em alguns casos.

O acesso aos guias oficiais do *PMBOK* e *PRINCE2* também se mostraram como uma limitação negativa ao trabalho. Em ambos os casos apenas os associados têm acesso irrestrito ao material. Em relação ao *Scrum* essa limitação não se aplica, pois, o guia está disponível a todos.

Como trabalho futuro, pretende-se realizar a implementação do *Scrum* em conjunto com uma das duas abordagens apresentadas em empresa de desenvolvimento de *software* que ainda não utiliza nenhuma das abordagens trabalhadas. Será possível verificar através de estudo de caso as peculiaridades da adaptação organizacional ao método e medir a eficiência da produtividade.

## REFERÊNCIAS

- Arnal, J., Del Rincón, D. & Latorre A. (1994) Investigación educativa: fundamentos y metodología. Barcelona: Labor.
- Beck, K., Cunningham, W., Hunt, A., Martin, R.C., Thomas, D., Beedle, M., Fowler, M., Jeffries, R., Mellor, S., Bennekum, A. V., Grenning, J., Kern, J., Schwaber, K. Cockburn, A., Highsmith, J., Marick, B., & Sutherland, J. (2001). Manifesto for Agile Software Development. Recuperado em 12 outubro 2016, de <http://agilemanifesto.org/>
- Brod, C. (2013) *ScrumScrum* Guia Prático para Projetos Ágeis. (2a. ed.) São Paulo: Novatec Editora.
- Canaltech. (2015) O que é *open source*?. Recuperado em 12 novembro, 2015 de <http://canaltech.com.br/o-que-e/o-que-e/O-que-e-open-source>
- Castro, C. M. (1978) A prática da pesquisa. São Paulo: McGraw-Hill do Brasil.
- Cervo, A. L., Bervian, P. A. & Silva, R. (2007) Metodologia científica. (6a. ed.) São Paulo: Pearson Prentice Hall.
- Chizzotti, A. (2003) Pesquisa qualitativa em ciências humanas e sociais. Petrópolis: Vozes.
- Cruz, F. (2013). *ScrumScrum* e Guia *PMBOK* unidos no gerenciamento de projetos. Rio de Janeiro: Brasport.
- Cruz, F. (2015) Introdução do Prince 2. Recuperado em 13 novembro, 2015 de <http://www.fabiocruz.com.br/oprince2>
- da Silva, E., & Lovato, L. (2016). Framework *Scrum*: Eficiência em Projetos de Software. *Revista de Gestão e Projetos - GeP*, 7(2), 01-15. doi:<http://dx.doi.org/10.5585/10.5585>
- Fernandez, K., Garrido, A., Raminez, Y., & Perdomo, I. (2015). *PMBOK* y PRINCE 2, similitudes y diferencias. *Revista Científica*, 23, 111-123. Doi: 10.14483/udistrital.jour.RC.2015.23.a9
- Graham, N. (2009). *PRINCE2* for Dummies. (2a. ed.) EUA: John Wiley & Sons.
- Jamali, G. & Oveisi, M. (2016). A Study on Project Management Based on *PMBOK* and *PRINCE2*. *Modern Applied Science*, 10(6), 142-146.
- Kardec, M. S. (2012). Estudo de Compatibilidade entre *PMBOK* e *Scrum*. *Revista Tecnologias e Projeção*, 3(1), 1-7.
- Machado, M., & Medina, S. G. (2009) *Scrum* – Método Ágil Uma mudança cultural na gestão de projetos de desenvolvimento de software. *Revista Científica Intracência*, 1(1): 58-71.
- Ludke, M., & Andre, M. E. D. A. (1986). Pesquisa em educação: abordagens qualitativas. São Paulo: E.P.U.
- Magalhães, G. (2005). Introdução à metodologia científica: caminhos da ciência e tecnologia. São Paulo: Ática, 2005.
- Martin. W. (2010). Agile Business Case management. [Blog] Retrieved November 25, 2015, from <http://www.borselaer.org/index.php/2010/03/whitebook-prince2-combined-with-scrum>
- Mascarenhas, S. A., (2012) Metodologia científica. São Paulo: Pearson Education do Brasil.
- Maximiano, A.C.A. (2008). Administração de projetos: como transformar idéias em resultados (3a. ed.). São Paulo: Atlas.
- Meredith, J.R., & Mantel, S.J., Jr. (2000) Project Management: a managerial approach (4nd ed.) New York: Wiley.
- Miles, M. B. & Huberman, M. A. (1994) *Qualitative data analysis*. (2nd.ed.) London: Sage.
- Miletto, E. M., & Bertagnolli, S. C. (Org) (2014). Desenvolvimento de software II: Introdução ao desenvolvimento com HTML, CSS, JavaScript e

- PHP (Dados eletrônicos). Porto Alegre: Bookman.
- Minayo, M. C., (1993). O desafio do conhecimento: pesquisa qualitativa em saúde. São Paulo: Hucitec-Abrasco.
- PMBOK *PMBOK*, Guia. (2013). Conjunto de Conhecimentos em Gerenciamento de Projetos (5a. ed.). Pennsylvania: PMI.
- PRINCE2 *PRINCE2*. (2009). *Managing successful projects with prince2PRINCE2*. United Kingdom: TSO.
- Rampazzo, L.. (2005). Metodologia Científica: Para alunos dos cursos de graduação e pós-graduação. São Paulo: Loyola.
- Schwaber, K. & Sutherland, J. (2013). Guia *ScrumScrum*. Recuperado em 12 outubro, 2015, de <http://www.scrumScrumguides.org/docs/scrumScrumguide/v1/ScrumScrum-Guide-Portuguese-BR.pdf>
- Schwalbe, K. (2014) Information technology project management (7a. ed. rev.) EUA: Cengage Learning.
- ScrumScrumAlliance. (2015) *The 2015 State of ScrumScrum Report*. Retrieved October 12, 2015, from <https://www.scrumScrumalliance.org/why-scrumScrum/state-of-scrumScrum-report>
- Severino, A. J. (2007) Metodologia do Trabalho científico. (23a ed.) São Paulo: Cortez Editora.
- Sommerville, I. (2003). Engenharia de Software (A. Ribeiro, trad.). São Paulo: Addison Wesley.
- Sommerville, I. (2011). Engenharia de Software (I. Bosnic & K. Gonçalves). São Paulo: Pearson Prentice Hall.
- Triviños, A. N. S. (1987). Introdução à pesquisa em ciências sociais: a pesquisa qualitativa em educação. São Paulo: Atlas.
- Vargas, R.V. (2000). Gerenciamento de Projetos: estabelecendo diferenciais competitivos. (2a. ed.) Rio de Janeiro: Brasport.
- Vieira, D. (2014) *ScrumScrum: A Metodologia Ágil Explicada de forma Definitiva*. *MindMaster*. Recuperado em 12 outubro, 2015 de <http://www.mindmaster.com.br/scrumScrum>
- Vitiello, J. (2001). Fast Track into Management. *Computerworld*. Retrieved November 29, 2015, from <http://www.computerworld.com/article/2582357/it-skills-training/fast-track-into-management.html>