

FRAMEWORK SCRUM: EFICIÊNCIA EM PROJETOS DE SOFTWARE

RESUMO

Scrum é um *framework* onde os times trabalham como uma unidade altamente integrada com cada membro desempenhando um papel bem definido e elimina controles desnecessários, inadequadas e burocráticas, se concentrando na essência do processo de desenvolvimento de sistemas ou *softwares*. Logo, o Scrum emerge como uma alternativa, principalmente para as organizações de *softwares*. Este artigo tem o objetivo de compreender a maneira como uma organização de tecnologia da informação situada na cidade de São Paulo aplica o Scrum para gerenciar os projetos de *softwares*. Com relação à metodologia, o estudo se caracteriza como um estudo exploratório e pesquisa-ação; no total cinco foram os entrevistados do estudo, que compreendem os principais papéis do Scrum; e foi conduzida uma análise qualitativa para melhor compreensão dos dados. Constatou-se que o Scrum é um *framework* e não um processo para o desenvolvimento de sistemas voltado às pessoas e não à tecnologia. Contudo, as práticas do PMBoK *Guide* podem complementar o Scrum, mesmo que os propósitos sejam distintos.

Palavras-chave: Scrum. Gestão de Projetos. PMBoK *Guide*.

SCRUM FRAMEWORK: EFFICIENCY IN SOFTWARE PROJECTS

ABSTRACT

Scrum is a framework whose members getting work together as integrated team, with each member is performing a specific function as a self-management team to cut out unnecessary, unappropriated and bureaucratic controls and focus on the essential, which is development processes of systems or software. Thus, Scrum arises as an option, mainly for software organisations. This paper aims to understand how a specific information technology enterprise located in São Paulo (capital) are applying Scrum to manage its softwares projects. Regarding to methodology, this study was defined as a exploratory and action research; we carry out five interviews with professionals who develop the main Scrum functions at this organisation; and a qualitative analysis was applied to understand of best way the data of this research. We recognise that Scrum is a framework and not a process for developing systems; and is oriented to people and not technology. Nevertheless, the PMBoK *Guide* practices may be complementary the Scrum, even the goals are distinct.

Keywords: Scrum. Project Management. PMBoK *Guide*.

Edson Coutinho da Silva ¹
Leandro Alvarez Lovato ²

¹ Doutor em Ciências Sociais pela Pontifícia Universidade Católica de São Paulo - PUC/SP. Professor do Centro Universitário da FEI. Brasil. E-mail: coutinho_ed@yahoo.com.br

² Especialista em Gestão de Projetos: PMBOK *Guide* pelo Centro Universitário da FEI. Brasil. E-mail: le.lovato@gmail.com

1 INTRODUÇÃO

Um dos desafios na área de gerenciamento de projetos de desenvolvimento de *software* está na dificuldade de se compreender os reais problemas e criar soluções que efetivamente atendam aos propósitos dos clientes. Não são raras as situações em projetos de engenharia de *software* onde, ao final do projeto, percebe-se que o que foi desenvolvido não é exatamente o que o solicitante estava esperando. Em outras palavras, simplesmente o sistema não gera valor para o qual foi criado. Seja esse fato ocasionado pela grande distância entre o cliente - quem compra e/ou consome o produto do projeto - e o fornecedor do projeto - quem desenvolve e vende o projeto - ou, em alguns casos, devido à dificuldade em se identificar quais os objetivos do sistema logo nas etapas iniciais do projeto.

Por um lado, cada vez mais popularizam-se nessa área a utilização de metodologias ágeis de gerenciamento de projetos. Seu maior destaque se dá pela grande valorização das interações entre as equipes e clientes, bem como a rápida resposta e absorção das mudanças identificadas como sendo necessárias nessas constantes interações. Por outro lado, organizações e clientes, ainda exigem certas formalidades e documentações e, não se sentem confortáveis com o método de planejamento mais dinâmico proposto pelo Scrum. Organizações ainda dão preferências para a utilização de métodos mais tradicionais, como a metodologia do *Project Management Body Knowledge* (PMBOK), onde os processos são mais formais e documentados.

Em virtude da característica flexível às mudanças presentes nas metodologias ágeis, muitos profissionais julgam existir incompatibilidade entre suas aplicações e as práticas recomendadas pelo PMBoK *Guide*. Contudo, pretende-se nesse estudo abordar a utilização do Scrum em uma empresa de tecnologia desenvolvedora de *software*, que já implementou o método há cinco anos. Para tanto, formulou-se a seguinte pergunta: “*de que maneira uma organização de tecnologia da informação situada na cidade de São Paulo aplica o Scrum para a gestão de projetos de softwares*”? Logo, o objetivo desse estudo é: compreender a maneira como uma organização de tecnologia da informação situada na cidade de São Paulo aplica o Scrum para gerenciar os projetos de *softwares*. Ao final do estudo, espera-se demonstrar que se, utilizada para os propósitos específicos, poderá aumentar a performance dos projetos e, conseqüentemente, agregar valor aos clientes e negócios, a partir de resultados concretos.

2 REFERENCIAL CONCEITUAL

2.1 Conceito de Scrum

Hiroataka Takeuchi e Ikujiro Nonaka produziram em 1986 um artigo denominado *The New Product Development Game*, com base em suas experiências no setor automobilístico e de tecnologia. Esta proposta introduzia um novo modelo que aumentava a velocidade e a flexibilidade do desenvolvimento de novos produtos comerciais. Neste estudo eles comparavam o novo método em que as fases possuíam forte intersecção e todo o processo era desenvolvido por times multifuncionais, do qual usou o esporte *rugby* como exemplo, uma vez que neste esporte todo o time trabalha em conjunto para avançar. Em 1991, DeGrace e Stahl, em *Wicked Problems, Righteous Solutions: A Catalog of Modern Engineering Paradigms* foram tidos como os primeiros a mencionar o nome Scrum. No início da década de 1990, Ken Schwaber começou o uso do modelo em desenvolvimento de *software* dentro de sua empresa a *Advanced Development Methods*. No mesmo período, Jeff Sutherland utilizou um modelo semelhante na *Easel Corporation*, sendo o primeiro a utilizar o nome Scrum para o desenvolvimento de *software* (Schwaber, 2004).

O Scrum baseia-se no princípio da objetividade, papéis bem definidos e facilidade de aprendizado. O Scrum é um modelo aberto e de forma alguma é previsível, pois o framework não detalha o que deve ser feito e não resolve os problemas da empresa e, sim, dar visibilidade a estes problemas e servir como guia para a construção de soluções dos mesmos. Ele oferece um conjunto de práticas que tem como objetivo manter o gerenciamento do projeto visível aos usuários do modelo. Alguns elementos são chave para a sua implementação em uma empresa (Carvalho & Mello, 2011; Sabbagh, 2013): equipes autogerenciáveis e organizadas; progresso de desenvolvimento por meio de *Sprints*, que são ciclos de desenvolvimento do Scrum, caracterizado por ter um curto período onde a equipe foca em atingir de uma meta específica; requisitos de produtos organizados em uma lista denominada *Product Backlog*; e o conceito de tempo fechado (*timebox*) onde todas as tarefas dentro do Scrum tem tempos máximos definidos para a sua execução e não devem suplantarem estes tempos.

Iteração é o mote do Scrum. Todo o desenvolvimento tem como base iterações com duração entre 2 a 6 semanas, chamadas de *Sprints*. A primeira etapa no *Sprint* é a reunião de planejamento (*Sprint Planning*), onde a equipe (*Scrum Team*), em conjunto com o cliente (*Product Owner*) define o que será implementado na interação, sendo responsabilidade do cliente realizar a priorização do trabalho a ser feito. A próxima etapa é a de execução, onde o time detalha as tarefas necessárias para implementar o que foi solicitado pelo cliente. Durante o *Sprint* a equipe realiza reuniões

diárias (*Daily Meeting*) para averiguar o acompanhamento do progresso do desenvolvimento, usando para isto o *Burndown Chart*, que é um gráfico usado para acompanhamento das tarefas a realizar, em andamento e as já realizadas. Ao final do *Sprint* é realizada uma reunião de validação da entrega (*Sprint Review*), onde o cliente e outros *stakeholders* do projeto podem verificar se o objetivo do *Sprint* foi atingido. Em seguida, é realizada apenas pela equipe do projeto uma reunião (*Sprint Retrospective*) onde o *Sprint* é avaliado sob a perspectiva do processo, equipe ou produto, quais foram os acertos e os erros com o propósito de melhorar o processo de trabalho (Schwaber & Sutherland, 2013).

Scrum é um arranjo de trabalho (*framework*) dinâmico para o gerenciamento de projetos a partir de práticas interativas e incrementais que buscam propiciar mais valor ao negócio. Utilizado com mais frequência por organizações que desenvolvem *softwares*, o Scrum é um *framework* do qual as pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível (Schwaber, 2007). Cruz (2013) diz que o Scrum não é um processo ou técnica, especificamente, mas sim, uma proposta de trabalho onde podem ser aplicados diversos processos e diversas técnicas com o propósito da eficácia no desenvolvimento de sistemas. Machado e Medina (2009) veem no Scrum uma abordagem empírica que propicia flexibilidade, adaptabilidade e produtividade alinhadas ao desempenho em mudanças constantes, característicos da engenharia de *software*, por exemplo: situações de trocas de membros do time, adaptações de cronogramas e orçamentos e linguagens de programação.

Segundo Rising e Janoff (2000), implementar o Scrum requer a interação com um processo empírico de gerenciamento de projetos baseados em evoluções e aprendizados contínuos, uma vez que em cada interação sejam detectadas falhas e possíveis melhorias do processo a serem aplicadas na próxima interação, são necessários três componentes para sua implementação: (a) transparência, onde todos os aspectos que afetam o resultado do processo devem ser visíveis e conhecidos por todos os responsáveis por esse resultado, e definidos a partir de um padrão para que todos os observadores compartilhem de um mesmo entendimento; (b) inspeção, onde todo o processo e artefatos do Scrum devem ser inspecionados frequentemente para que os desvios e variações sejam rapidamente detectados e então modificados ou corrigidos; (c) adaptação, se durante a inspeção são detectados variações fora dos limites aceitáveis, de forma que o produto resultante será inaceitável, deve-se realizar as adaptações necessárias ao processo para que o produto retome o caminho desejado. Os ajustes devem ser feitos o mais rápido possível a fim de minimizar novos desvios.

Terlizzi & Biancolino (2014) acreditam que o Scrum é considerado uma metodologia ágil e é um *framework* estrutural dentro do qual é possível empregar vários processos ou técnicas. O Scrum vem se fazendo

presente no cotidiano das organizações desenvolvedoras de *softwares* e/ou sistemas de informação. Lasing, Kishnah & Pudaruth (2012) apontam que a opção de uma organização pelo uso de metodologias ágeis já fornece aos desenvolvedores a sensação de que a organização confia em suas competências, somando-se a isso práticas do Scrum, que focam suas iterações entre os membros da equipe e como estes interagem, tem-se um clima favorável ao crescimento profissional com a troca de conhecimentos pela interdisciplinaridade da equipe.

2.2 Desenvolvimento de um Projeto Scrum

No início do projeto o *Product Owner* reúne-se com o cliente e demais partes interessadas para definir o produto e, em seguida, o objetivo e as necessidades de negócio do cliente. O *Product Owner* e os seus membros devem identificar as necessidades dos clientes e criar um plano de como se espera que o produto evolua no transcorrer do tempo, denominado *Roadmap* do Produto. Este plano contempla todas as metas e etapas para o desenvolvimento do produto do projeto ao longo de um tempo definido. Na sequência, inicia-se a criação de uma lista única e ordenada do trabalho a ser feito (escopo) para garantir a entrega do produto do projeto, denominada *Product Backlog*. Nessa lista são registrados todos os requisitos, funcionalidade e características que irão compor o produto a ser entregue. Essa lista é gerida pelo *Product Owner* e são ordenadas por ele as etapas em prol de entregar o maior valor ao cliente do projeto no menor espaço de tempo possível. Logo, o próximo passo está em estabelecer o Scrum *Team* que irá desenvolver as etapas do projeto e iniciar o desenvolvimento do produto por meio dos ciclos de trabalho (*Sprints*) (Sanders, 2007).

O *Sprint* deve ser definido por duração fixa e constante, onde o incremento do produto é gerado pela equipe de desenvolvimento a partir do escopo do *Product Backlog*, priorizados pelo *Product Owner*. Durante o *Sprint* há uma série de reuniões para acompanhamento do projeto. Todo o trabalho durante um projeto é realizado dentro dos *Sprints*. Os *Sprints* ocorrem sucessivamente a partir do *Product Backlog* até o trabalho ser realizado e finalizado. Cada *Sprint* deve possuir um objetivo claro e definido pelo *Product Owner*, de modo que, ao final de sua duração, sejam entregues pela equipe de desenvolvimento as funcionalidades e requisitos que agreguem o valor desejado ao produto final do projeto. Os *Sprints* permitem previsibilidade que garante a inspeção e adaptação do progresso em direção à meta pelo menos a cada mês corrido (Schwaber & Sutherland, 2013). Para Vijayarathy e Turk (2008), o *Sprint* assegura a identificação dos pontos positivos e traçar estratégias para melhorar os pontos negativos. Esta é uma alternativa para se ter o feedback do desenvolvimento do projeto e evoluir os membros do time.

Para Hicks e Foster (2015), a cada início de *Sprint* do projeto deve ser realizada uma reunião de

Sprint Planning, onde é planejado todo o trabalho a ser realizado durante o período. Nessa reunião, o *Product Owner* apresenta à equipe de desenvolvimento quais os itens do *Product Backlog* deverão ser entregues durante o *Sprint*, de acordo com a meta de negócio a ser alcançada, juntamente com os critérios de aceitação de cada item a ser desenvolvido. Uma vez apresentado aos itens a serem desenvolvidos, a equipe de desenvolvimento é responsável por estimar o esforço necessário para entregar cada um dos itens, a variável de medida mais utilizada é a *Story Points*. Na visão destes autores, a média de *Story Points* entregues nos últimos *Sprints* e na quantidade de pontos atribuídos aos itens atuais, determina-se a conclusão de quais poderão ser entregues dentro do período fixo de tempo do *Sprint* que começará. Na sequência, caberá à equipe definir um plano de execução dos itens. Uma vez definidas as tarefas, elas podem ser representadas e organizadas em um *Sprint Backlog*.

Com vistas a facilitar a organização da equipe de desenvolvimento durante a execução dos *Sprints*, são realizadas as *Daily Meeting*, visando promover a comunicação sobre o trabalho que está sendo executado e servir de oportunidade para decisões rápidas com relação ao progresso do *Sprint*. Configuram-se em reuniões diárias com duração de 15 minutos onde cada membro da equipe de desenvolvimento deve informar o que fez desde a última *Daily Meeting*, o que pretende fazer nas próximas 24 horas e quais obstáculos estiveram ou estão em seu caminho que possam ser resolvidos para a realização do trabalho planejado. Essas reuniões diárias elevam a probabilidade de a equipe de desenvolvimento atingir o objetivo do *Sprint*, uma vez que destaca diariamente quais os pontos de atenção que podem impedir a conclusão do trabalho a ser realizado, permitindo que a equipe atue na contenção dos eventuais contratempos. Tais reuniões podem soar como uma perda de tempo ou desconfiança do *Product Owner* para com a equipe; longe disso, o papel das reuniões é de interagir com a equipe com o intuito de traçar alternativas e soluções ao inesperado para atingir o objetivo do *Sprint* e ser um incremento para servir de fonte para futuros *Sprints* (Kardec, 2012).

Ao final de cada *Sprint*, após a equipe de desenvolvimento ter gerado o incremento do produto que representa o valor visível aos clientes e demais *stakeholders*, devem ser realizadas ao menos duas reuniões consecutivas de inspeção e adaptação, ambas facilitadas pelo *Scrum Master*. Uma é a *Sprint Review*, onde estão presentes a equipe, o cliente, os *stakeholders* e o *Product Owner*. Durante o *Sprint Review* o incremento do produto é apresentado ao cliente e partes interessadas para que ambos avaliem se estão de acordo com os critérios de aceitação definidos e que opinem a respeito do que foi feito. As informações obtidas pelo *Product Owner* serão fundamentais para a manutenção e priorização do *Product Backlog* e para *Sprints* futuros, em vistas do produto ter uma gestão mais eficiente e eficaz para atender às necessidades dos clientes. Outra é

a *Sprint Retrospective* é somente para as equipes do Scrum, sem a presença de equipes externas ao projeto (Carvalho & Mello, 2012). É a oportunidade de os membros apontarem aspectos positivos e negativos no desenvolvimento de um *Sprint*. Em seguida, é criado um plano para implementar possíveis melhorias para a elaboração dos *Sprints* futuros.

2.3 Artefatos do Scrum

Pode-se afirmar que há 4 grupos de artefatos no Scrum: *Product Backlog*, *Sprint Backlog*, incremento e gráficos. O *Product Backlog* consiste em uma relação de itens de todo o trabalho que será realizado ao longo do projeto, com vistas a garantir a entrega de um produto do projeto. Esta lista apresenta todas as características, funções, requisitos, melhorias e correções que englobam as mudanças que devem ser conduzidas no produto em futuras versões. Essa lista é ordenada a partir de itens mais aos menos prioritários do projeto seguindo a perspectiva do objetivo do projeto. Por essa razão, os itens do topo da lista exigem descrições mais detalhadas que busquem justificar suas prioridades e, possivelmente, estimativas mais precisas de tempo, para que se possam ser incluídas aos próximos *Sprints* e desenvolvidas pela equipe (Sabbagh, 2013). Esta lista é gerenciada pelo *Product Owner* e a priorização considera a necessidade do cliente o *Return on Investment* (ROI) (Guntamukkala, Wen & Tarn, 2006).

O *Sprint Backlog* é o segundo artefato. O *Sprint Backlog* se refere a um conjunto de itens selecionados do topo do *Product Backlog* que devem ser entregues pela equipe de desenvolvimento durante um *Sprint*. No *Product Backlog* estão relacionadas todas as atividades e/ou tarefas e suas estimativas de tempo que a equipe de desenvolvimento julga necessárias para a entrega dos itens ou funcionalidades que foram determinadas e definidas durante a reunião de *Sprint Planning* e, conseqüentemente, o incremento do produto ao final do *Sprint*. (Sabbagh, 2013). No transcorrer do *Sprint* a equipe de desenvolvimento é responsável pela manutenção do *Sprint Backlog*, adicionando novas tarefas que um novo trabalho é necessário e atualizando a estimativa do trabalho restante ao concluir uma atividade (Schwaber & Sutherland, 2013). Assim sendo, o trabalho concluído e restante do *Sprint* fica visível a todas as partes interessadas do projeto.

O terceiro artefato é o incremento. Incremento de produto é o resultado do trabalho de um *Sprint*. Em outras palavras, é a soma de todos os itens do *Product Backlog* entregues durante uma iteração: funcionalidades, melhorias ou correções. Cruz (2013) acredita que o incremento do produto seja entregue na perspectiva em que o *Product Owner* possa decidir por fazer uma versão para os clientes do projeto ao final do *Sprint*.

Os gráficos se referem ao quarto artefato. Os gráficos de acompanhamento do trabalho são ferramentas para a visualização do processo do

cumprimento de uma quantidade mensurável e estimada de trabalho em um determinado tempo. Os gráficos fornecem visibilidade a todas as partes interessadas do projeto sobre a produtividade e desempenho da equipe, assim como uma visão do trabalho ainda pendente para a entrega de um determinado *Sprint* ou *release* (versão). Logo, é possível identificar se o trabalho restante tende a ser entregue no prazo desejado para o projeto considerando o esforço e a velocidade de desenvolvimento. Dessa forma, pode-se tomar decisões e realizar ações para aumentar as chances de o objetivo ser alcançado. Os gráficos mais comuns e utilizados pelas equipes de Scrum são os gráficos *Burndown*. Há dois modelos comumente utilizados (Schwaber, 2007):

- *Release Burndown*: um gráfico administrado pelo *Product Owner* que representa a quantidade de *Sprints* necessários para que uma determinada entrega planejada para o cliente seja finalizada. Ao analisá-lo, o *Product Owner* pode realizar o planejamento das entregas de incremento dos produtos aos clientes, reordenação dos itens do *Product Backlog* a partir dos prazos de entrega ou, até mesmo, mudar o escopo de uma determinada entrega já pré-definida.
- *Sprint Burndown*: é um gráfico que visa monitorar o progresso e desempenho da equipe de desenvolvimento em direção à entrega final de um *Sprint*. Quer se dizer que o gráfico mostra a soma total das horas (ou prazos) restantes de todas as tarefas que devem ser realizadas e concluídas em um *Sprint* e o consumo dessas horas durante dos dias planejados de iteração. O gráfico deve ser atualizado diariamente pela equipe de desenvolvimento e fornece uma linha de tendência de consumo das horas restantes para que se possa identificar se o objetivo do *Sprint* será alcançado.

2.4 Os Papéis do Scrum

De modo geral, há três papéis (ou nomenclaturas de funções) em um projeto de Scrum: *Product Owner*, Equipes de Desenvolvimento e *Scrum Master*. O *Product Owner* – dono do produto – é a pessoa responsável por garantir e maximizar a partir do trabalho da Equipe de Desenvolvimento, o retorno sobre o investimento no produto aos clientes do projeto. É o *Product Owner* o responsável pelas interações com o cliente e *stakeholders* para entender as suas necessidades e seus problemas de negócios e estabelecer a visão, a missão e o objetivo do projeto. Logo, o *Product Owner* deve estar alinhado com os envolvidos no projeto em vista a aprimorar relações e reduzir conflitos frequentes.

O *Product Owner* é o responsável por gerenciar o *Product Backlog*. Portanto, ele é quem prioriza os itens, deve expressar de forma clara e transparente o que vem sendo realizado no projeto. Ele é um profissional que deve estar acessível aos membros da Equipe de Desenvolvimento para esclarecimento de dúvidas sobre o produto. O *Product Owner* está na linha de frente do projeto, uma vez que ele é a figura do gerente, que interage com grande frequência e diariamente com todos os envolvidos no projeto e ninguém, além dele tem autoridade para interferir e priorizar quais serão as tarefas que irão requerer ajustes. É ele que faz o primeiro aceite dos itens que foram desenvolvidos e concluídos ao final de cada *Sprint* respeitando os critérios estabelecidos pelo cliente e/ou *stakeholders* e, somente após o aceite do *Product Owner* que é direcionado ao aceite deles. Logo, o *Product Owner* é gestor, tomador de decisões e deve liderar os membros da equipe em um projeto (Sanders, 2007).

Por outro lado, a Equipe de Desenvolvimento é um grupo multidisciplinar que organiza e realiza todo o trabalho e resultado planejado em um *Sprint*. Recomenda-se que cada equipe seja composta por 3 a 9 membros. É a Equipe de Desenvolvimento quem planeja, nas reuniões de início do *Sprint*, todo o trabalho necessário para a entrega dos objetivos estabelecidos pelo *Product Owner*. A Equipe de Desenvolvimento tem autoridade e propriedade para determinar tecnicamente como o produto será concebido e planejar e estimar o tempo necessário de suas próprias tarefas a serem executadas durante o *Sprint*. Ao mesmo tempo, ele é responsável e responsabilizado por todos os resultados (Kardec, 2012).

Para Salo e Abrahamsson (2008), por se referir a uma equipe multidisciplinar, logo ela deve possuir membros com habilidades para executar todo o trabalho necessário para entregar o Incremento do *Sprint*, evitando dependências de recursos externos à equipe. Uma Equipe de Desenvolvimento em Scrum deve ser autogerenciável, isto é, a equipe deve definir e acompanhar o andamento de suas tarefas rumo ao objetivo do *Sprint*. Para auxiliar a autogestão, a Equipe de Desenvolvimento dispõe de ferramentas do Scrum, como a reunião *Daily Scrum*. Nessa reunião, os membros da equipe expõem suas dúvidas, soluções e realizam o monitoramento do andamento das tarefas a fim de assegurar que tudo seja concluído conforme o acordado com o *Product Owner*.

O último papel é o *Scrum Master*. É o profissional responsável por garantir que as práticas, regras e teorias do *framework* Scrum sejam seguidas e aplicadas no transcórre do projeto. Este profissional deve agir como um facilitador do processo, garantindo que todos os eventos e reuniões sejam realizados quando necessários, assim como a utilização dos artefatos. Ele garante que todos os membros tenham conhecimento de quais são seus papéis e responsabilidade dentro da equipe. Ele também é responsável por assegurar com que quaisquer interferências externas ao projeto cheguem à

Equipe de Desenvolvimento com vistas à produtividade, agindo como um facilitador na resolução de problemas (Hicks & Foster, 2015).

O *Scrum Master* possui autoridade direta sobre o processo Scrum, mas não sobre os membros das equipes. Ele deve treinar a Equipe de Desenvolvimento em termos de autogestão e que os membros possam tomar suas próprias decisões de ordem técnica na direção dos objetivos para o *Sprint*. Um *Scrum Master* é o porta-voz da organização com relação ao Scrum, normatizando a comunicação para que as partes interessadas possam compreender o que vem sendo realizado. Este papel pode ser exercido por qualquer membro da equipe, mas, deve possuir características de comunicador e líder (Carvalho & Mello, 2012)

2.5 PMBoK versus Scrum

O Corpo de Conhecimento da Gestão de Projetos – PMBoK *Guide* – é um manual que apresenta as melhores práticas da disciplina de gestão de projetos, a partir de uma definição do PMI. Esta instituição estruturou este manual em cinco grandes grupos de processos: iniciação, planejamento, execução, monitoramento e controle e encerramento; em dez áreas do conhecimento: integração, escopo, tempo, custo, qualidade, recursos humanos, comunicação, riscos, aquisição e partes interessadas; além, de 47 processos – levando em consideração a quinta edição de 2012, pois a versão anterior, a quarta edição havia 42 processos, com nove áreas do conhecimento; na quinta edição foi incrementada a décima área do conhecimento: “*partes interessadas*” (*stakeholders*). Não é objeto deste artigo detalhar cada um dos 47 processos: (a) iniciação: 2 processos; (b) planejamento: 24 processos; (c) execução: 8 processos; (d) monitoramento e controle: 11 processos; e (e) encerramento: 2 processos. Caso haja interesse em conhecer cada um dos processos, recomenda-se o contato com o manual do PMBoK (2013).

É possível observar a partir do quadro, que todos os projetos são divididos em fases e, sejam grandes ou pequenos, têm um ciclo de vida definido. No mínimo, um projeto terá um estágio inicial, uma, ou várias fases

intermediárias, e uma etapa final. O término de cada estágio representa para o gerente de projeto, o patrocinador e os demais stakeholders, uma oportunidade de avaliar se o projeto deve avançar para a fase seguinte, que só deve ser iniciada quando as “*entregas*” da fase anterior tiverem sido revisadas e aprovadas. Entende-se por “*entrega*” tudo o que deve ser produzido para que a fase ou o projeto sejam encerrados; são elementos tangíveis, que podem ser avaliados e comprovados, ou seja, o escopo do projeto. O conjunto das fases coletivas atravessadas pelo projeto é denominado ciclo de vida do projeto. Os estágios encadeados – *handoff* – são as fases do projeto que percorrem o ciclo de vida do projeto. Assim sendo, pode-se afirmar que cada fase pode ser reconhecida pela apresentação de uma entrega específica (ou várias), marcando o final daquela etapa (Bomfim, Nunes & Hastenreiter, 2012; Silva & Gil, 2013).

Com relação às áreas do conhecimento, cada uma delas cumpre uma função na gestão de projetos: (a) integração - refere-se à coordenação de todos os aspectos do plano de projeto e envolve um elevado nível de interação; (b) escopo - diz respeito à definição de todas as atividades do projeto necessárias ao cumprimento das metas; (c) tempo - visa assegurar que o projeto termine dentro do prazo previsto; (d) custo - busca assegurar que o projeto termine dentro do orçamento aprovado; (e) qualidade - tem o intuito de assegurar que o projeto atenda aos requisitos com os quais se comprometeu com o cliente; (f) recursos humanos - abrangem todos os aspectos do gerenciamento e da interação das pessoas envolvidas no projeto; (g) comunicação - está relacionada com as habilidades gerais de comunicação entre os envolvidos no projeto; (h) riscos - referem-se, ao mesmo tempo, às ameaças e às oportunidades do projeto; (i) aquisição - diz respeito à compra de bens ou serviços de fornecedores externos e prestadores de serviços; e (j) partes interessadas – tem relação com aqueles que possuem interesse no projeto ou que são afetados positiva ou negativamente por seus resultados (Carvalho & Rabechini Jr, 2015; Maximiano, 2010; 2002; PMBoK, 2013).

Tabela 1 - PMBoK versus Scrum

Características	PMBoK	Scrum
Ter definido a priori	Escopo	Tempo (<i>Sprints</i>)
Responsável pela organização para atingir os objetivos do projeto	Gerente de Projeto	Scrum Master
Frequência de reuniões de status	Dependendo da complexidade, necessidade do projeto, alinhar a frequência	Diárias
Escopo	Bem definido nas fases iniciais do projeto e formalização através WBS (Work Breakdown Structure)	Escopo é definido em alto nível e os requisitos são priorizados e definidos de forma iterativa. Necessita de maior controle de planejamento.
Tempo	Cronograma detalhado para realização de todo o projeto.	Cronograma orientado a produto com entregas incrementais de 2-4 semanas.
Custo	Monitoração das alterações para que não altere o custo planejado.	Maior controle em função da rapidez na incorporação de alterações.
Qualidade	Processos de verificação, validação e plano de testes.	Programação em pares e testes incrementais.
Riscos	Análise de riscos durante todo o ciclo de vida do projeto.	Aplica-se o mesmo conceito do gerenciamento tradicional.
Comunicação	Documentação formal	Implícita interpessoal e colaborativa.
Recursos Humanos	Papéis claros e bem definidos.	Confiança nos membros da equipe e ambiente colaborativo.
Aquisição	Controle por contrato e escopo bem definido e documentado.	Presença do cliente, volatilidade de requisitos e pouca documentação.
Integração	Plano de projeto detalhado e controle total do projeto pelo gerente.	Plano do projeto evolutivo e gerente do projeto atuam como facilitador.

Fonte: Framework de Maturidade para mais Agilidade nas Empresas (2011, p. 44)

3 METODOLOGIA DE PESQUISA

Este estudo pode ser classificado como sendo exploratório, uma vez que estes pesquisadores buscaram compreender o uso a metodologia de gerenciamento de projetos Scrum por meio de uma organização desenvolvedora de *software* situada na Capital de São Paulo (Cassel & Symon, 1994). Este estudo ainda se configura como uma pesquisa ação e participante (Thiollent, 2005); porque um dos autores ter desenvolvido atividades profissionais na equipe de projetos Scrum dessa organização de tecnologia, logo, suas observações e registros foram considerados para fins de análise. Este estudo não se configura em um estudo de caso, pois estes pesquisadores não efetuaram um estudo profundo na organização pesquisada, mas

sim, na compreensão do contato com o *framework* de profissionais que aderiram há 5 anos o uso dessas práticas, já que anteriormente eles faziam uso, tão somente, da metodologia do PMBoK *Guide* (2013) do *Project Management Institute* (PMI).

Com relação ao campo de estudo, a organização pesquisada possui aproximadamente 100 funcionários em toda a organização, dois quais 70 atuam diretamente com atividades técnicas e os demais em gestão da organização e área administrativa. A equipe de desenvolvimento é composta por duas grandes equipes de desenvolvimento de *softwares*, uma responsável pelos projetos de evolução de produto e outra com projetos de curto prazo e customização aos clientes. Este estudo ter atenção à visão dos funcionários para

compreender a aplicação do Scrum na organização. Tendo como base os papéis do Scrum desempenhados em um projeto, conforme destaca a teoria, os autores selecionaram 5 entrevistados, respeitando tais papéis: 1 *Scrum Master* (SM), 1 *Product Owner* (PO), 1 Analista de Qualidade (AQ) – que busca garantir as entregas das etapas aos clientes –, e 2 Membros da Equipe de Desenvolvimento (MED 1 e MED 2). Todos foram escolhidos a partir das experiências, funções e acessibilidade. Para fins de análise, todos os entrevistados serão mencionados e citados no corpo do texto a partir das siglas em parênteses.

Por se tratar de um estudo qualitativo, os autores elaboraram um roteiro de entrevistas com 5 eixos: (a) o contato dos funcionários com o Scrum, em termos de interação; (b) o planejamento de um projeto; (c) a execução por meio dos *Sprints*; (d) o papel dos membros das equipes; (e) e encerramento do projeto. Estes eixos serão os tópicos da análise e Discussão dos Resultados. O processo de coleta de dados ocorreu em três momentos: primeiro, o contato de um dos pesquisadores com o ambiente de projetos da organização pesquisada, como membro da Equipe de Desenvolvimento; segundo, por intermédio de entrevistas semiestruturadas que foram conduzidas com os 5 informantes-chave a partir do roteiro com 15 questões abertas para compreender o objetivo deste estudo; cada entrevista teve a duração de 45 – 50 minutos em média e todas ocorreram entre outubro e novembro de 2014; e, por fim, pelas documentações e registros internos dos projetos que foram elaborados mediante metodologia Scrum. Cabe ressaltar que todas essas cinco entrevistas foram registradas e gravadas com a anuência e consentimento dos informantes da pesquisa para o uso exclusivo deste estudo.

Para fins de análise optou-se pela análise qualitativa a partir de análise de conteúdo, onde a partir da transcrição, foram elencados questões, aspectos e conteúdos que seriam utilizados para fins de análise. Não se optou por um método quantitativo de análise, uma vez que não havia um volume de pessoas significativas para se desenhar e generalizar conclusões a partir de um questionário; e também, porque estes pesquisadores estavam interessados em ouvir dos entrevistados toda a dinâmica de gerenciamento de projetos a partir da metodologia Scrum. Logo, pela fala e explicação dos entrevistados, é alcançar o objetivo da pesquisa, que é compreender a dinâmica de gerenciamento de projetos a partir da metodologia Scrum e uma pequena organização de tecnologia.

4 ANÁLISE E DISCUSSÃO DOS DADOS

4.1 O Contato com o Scrum

Todos 5 entrevistados, mais um dos pesquisadores deste artigo, têm em comum o fato de o contato com o framework Scrum ter ocorrido pela

primeira vez na organização de tecnologia pesquisada. Eles nunca haviam administrado e desenvolvido projetos por meio de Scrum em organizações anteriores. Era uma experiência nova a todos os entrevistados. Dentre os entrevistados, o SM e PO eram os que tinham mais vivência em Scrum antes de ingressar na organização – os demais obtiveram conhecimento do Scrum quando ingressaram na organização –, cuja implementação da metodologia da organização contou com suas contribuições. Foi o *Scrum Master* quem sugeriu o uso do Scrum pela organização, onde segundo ele a ideia nasceu a partir de uma palestra de desenvolvimento de *software*. Sabedor dos problemas que a organização de tecnologia vivia há 5 anos, ele identificou no Scrum uma alternativa para buscar resolver os problemas que a organização tinha com seus clientes. Os clientes criticavam a organização pela ausência na flexibilidade de requisitos e a proximidade com eles na construção do produto gerado pelo projeto.

Segundo o PO, que também participou da implementação do Scrum, “*nós não tínhamos expectativas do sucesso do uso dessa ferramenta na época*”, para ele a dificuldade residia “*na adaptação da equipe com a nova forma de trabalho e na ausência de experiência com o Scrum*”. Outro problema estava em alinhar o Scrum às expectativas dos clientes, na visão do SM “*os clientes não estavam acostumados com a ideia de estarem tão presentes durante todas as etapas do projeto*”. No início da implementação e adaptação, o planejamento em curto prazo e as entregas constantes – em vez da entrega completa ao final do projeto – geraram estresses nos clientes, de acordo com os dois entrevistados. O Scrum trouxe transparência aos clientes na forma de gerir projetos. Após alguns projetos bem-sucedidos, a desconfiança acerca da metodologia por parte do cliente foi diminuindo e fez aflorar a perspectiva de que a relação entre organização e cliente poderia evoluir com o uso das práticas do Scrum.

O MED I argumenta que o Scrum propiciou a ele e aos colegas o conhecimento de todas as etapas do projeto. Há 3 anos utilizando o *framework* Scrum, ele diz que aprendeu a lidar com a participação dos clientes nas etapas do projeto, onde ele destaca que evoluiu na competência de comunicação, quando comparada a sua experiência em outros projetos em organizações anteriores. Para o MED 2 e Analista de Qualidade, o Scrum permitiu com que a Equipe de Desenvolvimento tivesse mais autonomia acerca das decisões que são tomadas na execução das tarefas do projeto, além da validação constante das entregas contribuir para mensurar se os objetivos estão sendo alcançados ou não e ter oportunidades de promover mudanças exigidas pelos clientes.

Na visão do SM o Scrum ainda requer muitos ajustes na organização, contudo, ele aponta que é quase que um senso comum de que a agilidade de trabalho fez com que os projetos fossem conduzidos de maneira mais organizada e planejada. Na organização, apenas o *Scrum Master* e *Product Owner* vêm buscando aprimorar a

gestão de projetos mediante Scrum, os demais se concentram em aprimorar mais o desenvolvimento de competência de *software* do que de gestão. Os dois profissionais são responsáveis pela gestão dos projetos na organização que estão distribuídos em duas categorias: (a) aprimoramento de funcionalidades no produto principal, um *software* de atendimento ao consumidor utilizado por 200 clientes que são afetados sempre quando ocorrem mudanças; (b) e o projeto que trata das customizações realizadas no produto para atender necessidades específicas de clientes.

4.2 Pontapé Inicial e o Planejamento do Projeto

Em projetos concebidos por intermédio do Scrum, não há necessidade de formalizar um termo de abertura, como ocorre com quem utiliza a metodologia do PMBoK *Guide* (2013). O SM aponta o Scrum tem menos documentações para gerir. Ele salienta que anualmente os diretores representantes das diferentes áreas da organização reúnem-se para definir o planejamento estratégico onde são determinados os objetivos e metas do ano e, conseqüentemente, os produtos e/ou serviços que serão ofertados para atender à demanda de um dado mercado. Uma vez estabelecida os objetivos para o ano, os responsáveis pela área de desenvolvimento do produto escolhem os projetos necessários para atingi-los e determinam o orçamento e cronograma macro, que novamente serão submetidos aos diretores para aprovação. O PO argumenta que *“a aprovação é conduzida de uma forma mais dinâmica e menos formal (...) é possível, por exemplo, que um projeto seja iniciado e, logo após uma pequena entrega inicial (...)”, e “(...) ele pode ser paralisado se essa entrega não está realmente aderente ao mercado desejado”*. Logo, o desafio na iniciação dos projetos internos de evolução do produto está no levantamento de requisitos da funcionalidade que está para ser desenvolvida.

Toda e qualquer alteração no sistema ou nova funcionalidade incluída pode afetar toda a base de clientes da empresa que utiliza o *software*. O PO como responsável por este processo de análise de requisitos, explica que *“uma vez decidido realizar o novo desenvolvimento no produto, são feitas diversas visitas aos maiores clientes e estudadas novas maneiras com que o sistema é utilizado, como eles utilizam as novas funcionalidades e quais impactos elas teriam na atual operação”*. Uma vez levantadas informações consideradas suficientes para a construção de uma solução comum aos clientes, são criados protótipos de baixo custo que possam ser utilizados para demonstrar aos clientes, ou aos mercados desejados, como seriam as funcionalidades e se elas atenderiam suas necessidades e excederiam suas expectativas. Tão somente após essa confirmação é iniciado um planejamento mais completo sobre o projeto, bem como a execução das tarefas de desenvolvimento propriamente ditas.

Este processo no Scrum, do qual a empresa tem diversos clientes que podem ser afetados pelos seus resultados gerados pelo projeto é semelhante ao Gerenciamento dos *Stakeholders* do PMBoK *Guide* (2013). Tal processo tem o propósito de identificar todos os envolvidos que serão afetados pelo projeto, analisar suas expectativas e o impacto dos resultados e, em seguida, desenvolver um plano de ação para engajar as partes interessadas na execução e tomadas de decisões durante o projeto. Para o PO *“identificar os stakeholders e gerenciá-los (...) poderia ajudar não somente na organização das necessidades e expectativas dos clientes, como também diferenciá-los, uma vez que cada um dos clientes irá requerer ações distintas”*. É a inserção de uma visão de Negócios, Estratégia e Marketing em um ambiente de projetos. Em projetos menores, com um único cliente o levantamento de requisitos é rápido e pouco complexo devido ao número de envolvidos no projeto. Geralmente as necessidades são claras e os requisitos necessários para atendê-las são levantados imediatamente nas reuniões iniciais. Mas, o MED 2 aponta que para projetos com mais envolvidos, a adaptação e uso do processo Coleta de Requisitos, presente na área de conhecimento Escopo é útil e adequada.

Definidos os objetivos do projeto, e que a primeira visão do produto desejado esteja especificada e alinhada com os *stakeholders*, em todos os projetos da empresa são realizadas estimativas macro de custo e prazo do projeto. Para o SM essa estimativa tem objetivos distintos em cada tipo de projeto. Para os projetos de entrega de personalizações do produto para clientes específicos, o custo e o prazo dos projetos são estimados para fins comerciais e contratuais, para que se possa conhecer quanto o projeto custará a cada cliente e informá-lo quando ele receberá a nova funcionalidade. Mas, há também os projetos de investimento da empresa em grandes evoluções de produto, esse orçamento de custo e tempo é realizado principalmente para que os diretores tenham uma ideia de quanto será gasto até que haja uma versão básica que possa ser demonstrada aos clientes e, então, validada sua aderência ao mercado desejado. O SM faz uma observação acerca dessa questão: *“quanto maior o custo dessa primeira versão básica do projeto, maiores serão os prejuízos do projeto à organização se foram abortados (...) e isso ocorre quando o produto do projeto não foi o esperado”*.

O responsável pela etapa de planejamento do projeto – que inclui o orçamento – é o *Product Owner*. Logo quando a ideia inicial do produto do projeto é concebida, o *Product Owner* começa a decompor o produto do projeto em componentes menores entregáveis (*deliverables*) para serem mais facilmente gerenciáveis, geralmente definidos por funcionalidades do produto e que, ao serem concluídos, gerem um valor perceptível ao cliente. Cada uma dessas partes do produto recebe o nome de *User Story*, e todas elas são registradas no *Product Backlog*, uma lista mantida e ordenada pelo *Product Owner*, contendo todo o tipo de trabalho que

será realizado durante o projeto. Segundo o PO “*este é o principal documento gerado durante a etapa de planejamento do projeto (...) e é baseado nele que são definidas e organizadas as entregas do projeto - durante os Sprints – aos clientes*”. Essa etapa é similar à Criação da WBS (*Work Breakdown Structure*) no PMBoK Guide (2013), conhecida em português como Estrutura Analítica de Trabalho (EAP). No PMBoK Guide (2013) essa é a etapa onde se descreve tudo que será realizado – inclusive os pacotes de trabalho, que configuram o último nível da WBS – em um projeto, ou seja, a Gestão de Escopo em Projetos.

Após a definição das *User Stories*, elas são levadas até a Equipe de Desenvolvimento para que seja definido o esforço necessário para a entrega de cada um dos componentes do produto. Na visão do MED 1 é essencial a participação de toda a equipe técnica que estará envolvida no desenvolvimento para que todos sintam-se responsáveis pelas tarefas e orçamento de cada entrega. A unidade de medida utilizada para definir o esforço das entregas é *Story Point*. A responsável por transformar esses Story Points em unidades de medidas comuns de tempo e custo é o *Scrum Master*, considerando informações históricas sobre a velocidade de trabalho da equipe e quantidade de pontos entregues nos *Sprints* anteriores. Com a posse dos esforços necessários para cada uma das *User Stories*, o *Product Owner* tem informações suficientes para realizar o planejamento e definição do que será realizado e quanto custará a primeira entrega básica do projeto, que levará até a primeira onda de valor. Na opinião deste entrevistado, a participação do cliente nesta definição é essencial para que a primeira entrega atinja logo as necessidades iniciais e possibilite a utilização do produto e geração de *feedbacks* para que os eventuais erros ou equívocos sejam prontamente corrigidos e/ou solucionados. O SM diz que “*neste momento do planejamento, é possível realizar um cronograma macro contendo alguns marcos de entrega de valor ao cliente*”, mas, que esse cronograma pode ser facilmente alterado conforme os *Sprints* vão evoluindo e, assim, o “*próprio cliente pode decidir trocar a ordem das entregas de acordo com as informações que vão sendo obtidas com a evolução do projeto*”.

Definidas as entregas de valor, inicia-se o planejamento do *Sprint* a partir de uma reunião de *Sprint Planning*, da qual a Equipe de Desenvolvimento planeja como cada *User Story* será entregue durante o ciclo de desenvolvimento. Para isso, cada uma delas é decomposta em tarefas que são estimadas em horas de desenvolvimento, também pela própria equipe. No PMBoK Guide (2013) essa abrange os Gerenciamento de Tempo e Custos em Projetos.

Os cinco entrevistados foram unânimes em afirmar que o principal ponto forte do Scrum é o alto nível de envolvimento do cliente em todas as etapas de especificação do planejamento. O PO acredita que “*essa proximidade eleva as chances de ter-se, ao final do projeto, um resultado que realmente satisfaça as*

necessidades do cliente, pois fornece oportunidade de identificar desvios entre o que foi planejado e entregue enquanto eles ainda podem ser corrigidos em tempo e custos mais baixos”. O MED2 coloca que “*embora alguns clientes sejam relutantes e resistentes em participar no início das atividades (...) ao longo do projeto eles acabam cedendo e percebendo os benefícios de atuar ao lado daqueles que estão desenvolvendo às soluções a eles*”. Constata-se que ao longo das atividades os *stakeholders* são convidados com frequência para as reuniões principais com os *Sprints*. Os *stakeholders* se fazem presentes nas reuniões de *Sprint Planning*, onde são definidas as entregas para aquele ciclo de desenvolvimento e como elas serão construídas. Já a segunda participação fica para a reunião de *Sprint Review*, onde o incremento do produto é apresentado ao cliente e os *stakeholders* do projeto, para que eles avaliem se o que foi produzido está alinhado com os critérios de aceitação estabelecidos e que forneçam *feedbacks*. O AQ é de opinião que essas reuniões que contam com participação do cliente são cruciais, uma vez que as informações obtidas enriquecem a perspectiva da realização de um projeto adequado ao cliente e que o produto será o que eles desejam.

4.3 Ciclo Sprints

Os projetos da organização pesquisada têm ciclo de três semanas de duração cada. Essa duração é sempre estabelecida e fixa e o trabalho realizado durante esse tempo é definido nas reuniões de *Sprint Planning*, onde as *User Stories* que serão entregues são escolhidas por meio do critério do tamanho orçado pela Equipe de Desenvolvimento. Logicamente que a determinação de três semanas foi determinada com base na experiência da organização em projetos e com *Sprints* – produtividade. Segundo o SM, inicialmente eram, por um lado, realizados *Sprints* de uma ou duas semanas, mas o período mostrou-se curto para entregar algumas tarefas, porém, por outro lado os *Sprints* maiores fariam com que os clientes demorassem para ver o resultado de cada iteração, diminuindo a quantidade de *feedbacks* e podendo aumentar a quantidade de retrabalho. O *Sprint* de 3 semanas é uma das principais características do Scrum para o desenvolvimento de *softwares*, como apontam Leite e Lucrédio (2014) em uma pesquisa na Universidade Federal de São Carlos (UFSCar). Em desenvolvimento de *softwares*, como afirma MED 1, é comum que as expectativas do cliente na entrega do sistema, se eles não participarem da construção do mesmo. Para o entrevistado, a divisão em ciclos de desenvolvimento eleva as chances de tudo correr como o esperado pelo cliente, pela proximidade e participação do cliente durante do projeto, dando a ele a oportunidade de validar o que foi feito ao final de cada *Sprint*, emitir suas opiniões e solicitar mudanças que seriam mais custosas no futuro.

Outra vantagem destacada pelo PO acerca do uso de *Sprints* para a execução de projeto é a

possibilidade de entregar rapidamente valor para os clientes antes do final do projeto. Segundo ele “*um bom planejamento de entregas e Sprints pode, por exemplo, permitir que o cliente utilize o sistema antes dele ter todas as suas funcionalidades desenvolvidas, mas que já seja capaz de atender suas principais necessidades*”. Ele ainda destaca que essa é a característica preferida dos clientes, uma vez que permite que eles, durante a utilização do sistema tenham informações para priorizar as próximas funcionalidades que serão desenvolvidas de forma que elas também entreguem o máximo do valor possível e, eventualmente, até removam do *Backlog* algumas funcionalidades que ela perceba que não serão necessárias, assim “*isso evita uma série de trabalhos desnecessários e gera uma economia considerável de tempo e custo nos projetos*”.

Durante os *Sprints*, a própria Equipe de Desenvolvimento é responsável por gerenciar suas tarefas que devem ser concluídas para entregar o que foi combinado para o ciclo. O SM relata que seu principal papel durante os *Sprints* é blindar a equipe e evitar que problemas externos ao projeto, novos pedidos, ou qualquer outra interferência que possa influir na produtividade e velocidade da equipe chegue até ela. Para ele “*no início da utilização do Scrum, foi um grande desafio conscientizar as demais áreas da organização de que a equipe precisava de foco durante os Sprints*” e que quaisquer outras demandas que surgisse “*seria tratada com prioridade menor e alocada e planejada somente no próximo ciclo*”. Contudo, com a maturidade do *framework* de Scrum na empresa, todo o processo está alinhado de modo que a organização se tornou uma grande aliada da produtividade além de facilitar a previsão de quando cada entrega será realizada, aumentando a satisfação de todos os envolvidos no projeto.

Sobre o acompanhamento do progresso das tarefas do *Sprint*, o MED 1 vê que a principal vantagem do Scrum está na comunicação do Scrum, e um de seus principais fatores de sucesso é a *Daily Scrum*. Ele argumenta que o *Daily Scrum* auxilia: (a) na disseminação do conhecimento sobre o escopo diário do projeto; (b) na identificação dos impedimentos; (c) e na priorização do trabalho a ser realizado no dia que se inicia. Contudo, observa-se que o *Daily Scrum* não deve ser utilizada com o propósito de resolver problemas. É o *Scrum Master* e a Equipe de Desenvolvimento que monitoram o progresso do *Sprint* a ser realizado no *Daily Scrum*, por intermédio de reuniões curtas e realizadas diariamente. MED 2 ainda complementa que a reunião permite identificar diversos problemas que poderiam ser encontrados somente em um período mais crítico do projeto, quando seria mais caro e demorado resolvê-lo. Ele reforça que o objetivo da reunião não é efetivamente resolver todos os problemas, mas sim, identificá-los e definir os responsáveis por resolvê-los fora da reunião; envolvendo todas as outras pessoas que forem necessárias.

Ainda no que se refere ao monitoramento e controle dos *Sprints*, o PO mencionou o uso de gráficos *burndown* para o acompanhamento do desempenho da Equipe de Desenvolvimento na conclusão das tarefas, baseado no que foi planejado durante o *Sprint Planning*. Em suma, esse gráfico expõe a quantidade de horas planejadas que ainda podem ser consumidas pela equipe e a distribuição do consumo entre os dias que ainda restam de *Sprint*. Este gráfico analisado diariamente pela Equipe de Desenvolvimento durante o *Daily Scrum*, para que ela identifique se o ritmo atual é suficiente para concluir as entregas planejadas ao final do ciclo e, ainda se está atrasada e, portanto, serão necessárias atitudes corretivas para atingir o objetivo.

Por mais que seja uma característica forte do Scrum o gerenciamento das tarefas do *Sprint* pela própria equipe, alguns *stakeholders* dos projetos exigem relatórios de acompanhamento de progresso mais formais e completos para sentirem-se mais seguros ou até mesmo para reportar essas informações para outras áreas superiores. Cabe mencionar que os processos do PMBoK (2013) que tratam do Monitoramento e Controle do Trabalho do Projeto estão presentes em todas as etapas do desenvolvimento de projetos de sistemas no Scrum, é inerente às práticas das metodologias ágeis. Logo, os entrevistados reconhecem que o seu benefício é permitir que os *stakeholders* entendam o real *status* do projeto, quanto dinheiro e tempo já foram gastos e o que pode ser desenvolvido em relação ao escopo total do projeto. O *Burndown* não é parte integrante do Scrum, mas auxilia o *Scrum Master* e o *Product Owner* a suprir as expectativas de informações dos clientes e demais áreas por intermédio de gráficos que apresentam o andamento dos *Sprints*.

Se o *Scrum Master* identificar a necessidade de um controle mais formal devido aos riscos encontrados na fase de planejamento do projeto, ou nas reuniões diárias de acompanhamento. O PMBoK *Guide* possui uma área de conhecimento dedicada somente aos riscos, com processos, tais como: planejamento, análise e controle dos riscos. Mas, o Scrum surge como um *framework* para mitigar riscos frequentes em projetos tradicionais, porque são mais dinâmicos e ágeis, algo que não se observa no PMBoK *Guide*. O AQ argumenta que os testes técnicos e funcionais são realizados juntamente com os desenvolvedores durante a execução das tarefas. Segundo este entrevistado a comunicação é essencial para a organização da equipe. Cada vez que os desenvolvedores julgarem terminarem uma *User Story*, os analistas de qualidade imediatamente iniciam os testes e, caso identifiquem eventuais inconformidades ou falhas, comunicam os desenvolvedores que direcionam à funcionalidade para corrigi-la. Por essa razão, este entrevistado acrescenta que se trata de um processo dinâmico e não estático, porque a equipe ganha agilidade e diminui as chances de uma funcionalidade que gera valor não chegar ao cliente devido a formalidade dos processos de qualidade.

Após o final de cada *Sprint*, também há uma validação da qualidade e dos objetivos funcionais realizados pelos clientes na reunião de *Sprint Review*. Todas as inconformidades encontradas aqui, bem como as alterações sugeridas pelos clientes, são imediatamente incluídas no próximo *Sprint* para desenvolvimento. O PMBoK *Guide* (2013) possui também uma área de conhecimento direcionada à Gestão de Qualidade em Projetos, mas acredita-se que a proximidade do cliente com as entregas de uma forma ágil é a maneira de assegurar a qualidade em projetos gerenciados pelo Scrum. Salvo algumas exigências pontuais dos *stakeholders* ou problemas recorrentes que surgiram a necessidade de um processo mais formal.

Além da *Sprint Review*, ocorre também a reunião de *Sprint Retrospective*. O *Scrum Master* atribui à essa reunião uma das características que ele julga mais importante do Scrum, o aprendizado e a evolução contínua do processo de execução. Todos os integrantes dessa equipe participam dessa reunião, frequentemente, onde são elencados todos os pontos positivos que contribuíram para o sucesso – ou não – do *Sprint* e que devem ser mantidos para a próxima interação, bem como os pontos negativos que devem ser corrigidos. Essas informações são semelhantes às lições aprendidas do PMBoK *Guide* (2013), registradas normalmente durante todo o projeto, mas muitas vezes centralizadas no processo Encerrar Projeto ou Fase.

4.4 O Gerenciamento de Equipes

Há de se reconhecer que a forma como a equipe é gerenciada mudou radicalmente na área de desenvolvimento de *software* da empresa com a adoção do Scrum como *framework* de gestão de projetos. O *Product Owner*, que era desenvolvedor na época que ainda eram utilizadas as práticas tradicionais na empresa, lembra que o envolvimento da equipe com o cliente e com o real objetivo do projeto era baixo. Todo o processo de planejamento, considerando estimativas de custos e tempo eram conduzidas pelos gerentes com consultas somente aos desenvolvedores mais experientes. Para o PO, “*essa prática afetava o comprometimento da equipe com as entregas do projeto, uma vez que eles recebiam somente um conjunto de tarefas e funcionalidades para serem desenvolvidas, porém, com a ausência de alinhamento com as necessidades do cliente que deveriam ser atendidas*”.

Na oportunidade a cobrança por desempenho da equipe e atingimento dos objetivos e metas propostos eram realizadas de um modo pouco eficiente antes do Scrum, como destacou o SM. Pelo fato de não haver envolvimento da equipe, a cobrança era realizada pontualmente sobre as tarefas e seus prazos. Similarmente a uma “linha de produção”, logo era possível que o projeto não atingisse seus objetivos, mas ainda sim a equipe atingiria suas metas. No entanto, com o advento do Scrum e outras metodologias de gerenciamento de equipe, a empresa melhorou os

resultados dos projetos, o SM relatou “*com o amadurecimento do Scrum na equipe, o maior envolvimento dela durante todas as etapas do projeto e a adoção de algumas técnicas de autogerenciamento, o comprometimento e a satisfação da equipe se elevou significativamente, os projetos bem-sucedidos*”.

Hoje, conforme citam MED 1 e 2, durante a etapa de concepção do projeto, toda a equipe é frequentemente consultada para sugerir não somente soluções técnicas, como também possíveis alternativas funcionais. Para os dois entrevistados, o principal momento de participação da equipe está nos orçamentos realizados durante o *Sprint Planning*. Como toda a equipe está presente nessa reunião, e os custos e os prazos das *User Stories* são dados pelo consenso de todos os presentes, o comprometimento em atingir o que foi proposto é muito maior. Ademais, a visão clara e o foco em atingir o valor proposto ao cliente proposto para um *Sprint* que é considerado um curto espaço de tempo, ajuda na motivação da equipe e na tomada de decisões.

O AQ destaca a relevância, para toda a equipe, da proximidade e participação constante do cliente durante toda a participação constante do cliente em todo o curso do projeto. Cabe mencionar que os integrantes da equipe de desenvolvimento são convidados a visitar o ambiente de trabalho dos clientes e entender como o sistema será utilizado. Todas as informações aumentam a motivação da equipe por conseguir enxergar exatamente o valor que seu trabalho levará até o cliente. Também fornecem mais segurança para tomadas de decisão da equipe durante o projeto, uma vez que fazem com que a equipe conheça bem as necessidades que devem ser sanadas, e perceberam como essas alterações podem afetá-las.

Nota-se que tanto o AQ quanto MED 1 e 2 caracterizam que a principal contribuição do uso da metodologia Scrum está na autonomia para tomar decisões e programar suas próprias atividades durante um *Sprint*, elevando a responsabilidade da equipe com os resultados a serem alcançados e, em seguida, aumenta a motivação. Observa-se que a equipe é cobrada, bem como avaliada em termos de entregas e padrões de qualidade para com os clientes e com o *Product Owner* conforme estabelecido no início do *Sprint*. Todos os integrantes da equipe têm total flexibilidade e autoridade de distribuição das tarefas, desde que as cumpram. Até mesmo o horário de trabalho é flexível. Mas, conforme aponta o SM que para se alcançar esse estágio a equipe passou por um amadurecimento de autogerenciamento, onde uma série de decisões foram delegadas aos integrantes. O desconforto inicial se transformou em algo confortável à equipe, sem receios e inseguranças. O estabelecimento de uma confiança mútua foi um dos desafios do processo do Scrum na organização.

O MED 2 salienta que a meta hoje não está somente nas entregas das tarefas planejada no início do *Sprint*, cumprir prazos e custos. Hoje a equipe também é responsável pela satisfação dos clientes com o produto, a

diminuição de perdas de clientes, etc. Metas e trabalho ganharam um novo significado, e as decisões tomadas durante o transcorrer do projeto passaram a ser avaliadas como um aspecto relevante para a equipe. Constatou-se que a organização envolveu a equipe nos negócios da organização, onde a competência anterior era limitada às questões meramente técnicas e passou a se considerar também a competência de gestão em negócios.

4.5 O Encerramento do Projeto

O encerramento do projeto possui características distintas para os dois principais papéis em um projeto Scrum, como destaca PO. Os projetos internos e mais longos, de evolução do produto, raramente possuem uma finalização formal. Há situações em que nas etapas iniciais do projeto são detectados, mediante estudos com os clientes, que a funcionalidade não agregará valor desejado aos clientes, ou que seu impacto será negativo para a maior parte dos usuários. Se isto ocorrer, o projeto é abortado logo nos primeiros *Sprints*, um cálculo do valor investido na tentativa de evolução é feito para registro interno e, em seguida, inicia-se um novo projeto.

Casos assim, em que o projeto de evolução segue adiante, após uma quantidade significativa de entregas, começam a “colher” as opiniões dos clientes que as utilizam. Quando alterações são sugeridas, elas são analisadas pelo *Product Owner* e, eventualmente, alocada no *Sprint* para poder ser desenvolvida. Trata-se, portanto, de uma evolução “contínua” de um projeto e do produto, e os valores das *User Stories* vão sendo adicionadas aos centros de custos de seus respectivos. Por sua vez, em projetos curtos, realizados para o desenvolvimento de customizações especificadas para um único cliente, o projeto é finalizado quando todo o escopo acordado no contrato inicial é entregue ao cliente. Como, na maioria dos casos, o cliente esteve presente durante o início e fim de cada *Sprint* e, muitas vezes já está utilizando uma parte da funcionalidade desenvolvida, as “surpresas” ao final do projeto são raras, como justifica o PO. Comumente a apresentação da funcionalidade completa com todas as entregas do *Sprint* é suficiente. Após isso, o cliente deve declarar formalmente que o projeto foi encerrado e a entrega realizada.

Também é comum que, durante uma ou duas semanas após a finalização formal do projeto, exista um acompanhamento da equipe de pós-vendas junto ao cliente. Caso sejam encontradas necessidades de alteração, elas serão analisadas pelo *Product Owner*, juntamente com a Equipe de Desenvolvimento e, então, decidido se a empresa absorverá os custos da mudança ou se um novo desenvolvimento será requerido do cliente.

5 CONSIDERAÇÕES FINAIS

Scrum é um *framework* - e não uma metodologia, porque ele não diz o que fazer -, direcionado à solução de problemas em *software*, que é composto por um conjunto de diretrizes que podem propiciar uma gestão eficiente e eficaz em projetos. O ponto forte do Scrum reside na relação a outros métodos ágeis e parece ser relativamente simples de se implementar, mas não é, pois requer a mudança de comportamento da equipe de desenvolvimento e do cliente. É um método democrático já que as opiniões dos membros da equipe são consideradas para estimar o escopo, os prazos, os custos e os padrões de qualidade em termos de soluções aos clientes. O empirismo é algo constante no Scrum, cuja teoria parte do pressuposto que o profissional desenvolve melhor as atividades com a experiência vivida em outros projetos. O Scrum surgiu com uma abordagem incremental e iterativa para melhorar a previsibilidade e o controle de riscos e busca contemplar valores e princípios em que se valoriza mais a colaboração e interações com ferramentas e menos as documentações.

Este estudo de caráter exploratório e qualitativo buscou compreender a forma como uma organização de tecnologia da informação, mais específico de desenvolvimento de *software*, aplica o Scrum para gerenciar de projetos de *software*. Este estudo surgiu do interesse de um dos pesquisadores que foi membro da Equipe de Desenvolvimento em demonstrar a aplicação do Scrum em conjunto com alguns processos do PMBoK *Guide* (2013). Dentre os 5 entrevistados, o *Scrum Master* e *Product Owner* são os mais experientes com o *framework* Scrum, os outros 3 entrevistados tiveram contato com o Scrum nesta organização pesquisada, aproximadamente 3 anos. Todos os profissionais que participaram da pesquisa aprimoram seus conhecimentos acerca do *framework* em seminários, cursos e treinamentos com certificações nacionais e internacionais. Este investimento realizado por eles próprios e pela organização, vem fomentando a gestão de conhecimento entre as equipes, uma vez que a organização reforça que o conhecimento é algo que merece ser compartilhado, e não ser posse dos profissionais da organização.

Foi possível observar que a aplicação conjunta e compatível do Scrum e PMBoK *Guide* (2013) se faz mais presente na etapa de planejamento e, principalmente, em projetos maiores de , com níveis de complexidades elevados - de evolução do produto principal -, onde a quantidade de *stakeholders* e clientes são maiores. Dos processos do PMBoK *Guide* (2013), o Termo de Abertura - o *Project Charter* - e um processo, ao menos, das áreas de conhecimento como Gestão de Escopo, Tempo, Custos, *Stakeholders*, Qualidade e Riscos foram implementados na gestão de projetos na organização. Cabe apontar que a compatibilidade do PMBoK e Scrum somente foi possível, porque os integrantes das equipes de projetos têm conhecimentos

tanto de PMBoK quanto de Scrum, ao ponto de saber o que – em termos de processos – pode ser integrado ou não para os propósitos do projeto. É um sinal da maturidade tanto das equipes quanto da organização, onde a sinergia entre as a metodologia PMBoK *Guide* e *framework* Scrum e os membros da equipe podem produzir atividades bem-sucedidas e agregar valor aos seus clientes.

Constata-se que Scrum é altamente recomendável para empresas que tenham no seu ciclo de projetos, produtos dinâmicos e que possuam alta taxa de mudança de requisitos. Projetos como os da organização pesquisada, tem características de evolução, onde após uma quantidade significativa de entregas, inicia-se um processo dinâmico de interações para ouvir opiniões dos clientes e, sempre que há uma alteração solicitada pelos clientes e/ou *stakeholders*, inicia-se um novo *Sprint*. Projetos menores – ou menos complexos –, que no caso são os projetos de customizações específicas a um único cliente, o projeto é finalizado somente quando todo o escopo acordado é entregue, portanto, o cliente participa somente no início e fim dos *Sprints*. Então, pode-se afirmar que o Scrum requer equipes multifuncionais e multidisciplinares, em projetos tradicionais, são pouco desenvolvidas: visão de negócios, adaptabilidade e flexibilidade, foco em objetivo e metas, gestão participativa e de conhecimentos, proatividade e inovação e, por fim, ser um profissional solucionador de problemas dos *stakeholders*.

Estes pesquisadores puderam, ainda, observar pelas falas dos entrevistados que a eficiência dos membros da equipe do Scrum está relacionada à colaboração e comprometimento deles. É fato que trabalhar em equipe e características proativas são pré-requisitos para compor a equipe de Scrum. Por mais que o ambiente e equipes autogerenciáveis existem atribuições e delegação de atividades. Contudo, a função de controle pertence aos próprios membros da equipe como um todo, que escolhe a melhor maneira de trabalhar para cumprir os objetivos do projeto. Constatou-se que o ganho do Scrum em relação ao PMBoK *Guide* está na agilidade das equipes, na facilidade de comunicação e, também, na produtividade da equipe. O Scrum parte da premissa que o tempo e custos são fixos, porém, o escopo é variável, pois é definido no desenvolvimento do projeto.

Reconhece-se duas limitações neste estudo: uma é não contemplar a perspectiva dos clientes na aplicação do Scrum; e outra a quantidade de entrevistados. Na concepção deste estudo, esperava-se contar com as visões dos clientes para contemplá-las com os membros da equipe para incrementar e afrontá-las, contudo, após a rejeição de cinco clientes que foram convidados; estes pesquisadores optaram, apenas, pelas visões dos membros da equipe. Com relação ao número de entrevistados, a dificuldade estava em encontrá-los, uma vez que grande parte dos membros da equipe não trabalham na unidade da organização em todos os dias da semana, alguns deles desenvolvem suas atividades

Home Office; e os que lá estavam disponíveis, não se sentiram confortáveis para contribuir com este estudo. Diante disso, estes pesquisadores tiveram que contar, apenas, com as falas e explicações dos cinco entrevistados. Como sugestão para futuros estudos acerca do tema, estes pesquisadores veem que um estudo mais amplo e quantitativo poderá trazer novas contribuições para o campo de estudo.

Enfim, a implementação do Scrum requer mudança de cultura de trabalho na organização. A organização pesquisada optou há cinco anos pelo Scrum e, na perspectiva dos entrevistados, foi uma escolha difícil, pois inferiu em mudanças de pessoas, rotina de trabalho e relações com os clientes. O diagnóstico na oportunidade foi: “*nós precisamos encontrar novas maneiras para desenvolver nossas atividades ou trabalho, ou continuaremos a perder nossos clientes*”, como ressaltou o SM. Ou seja, mudanças são oriundas de um processo dinâmico, do qual a pressão do mercado, cliente e concorrente encorajam a organização a buscar novas alternativas. Hoje, a organização, após alguns anos se adaptando, adequando e amadurecendo o método Scrum, demonstra, pelos entrevistados, de que encontrou, a partir das características de negócios, a melhor maneira de gerenciar seus projetos e agregar valor e satisfazer seus clientes. Lidar com equipes, *stakeholders* e clientes é algo complexo, mas, quando a organização entende o que oferece e desenvolve e, simultaneamente, compreende o que as equipes, os *stakeholders* e os clientes desejam, o primeiro passo foi dado, o próximo está em adequar o interesse de ambos.

REFERÊNCIAS

- Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, 21(5), 61-72.
- Bomfin, D. F.; Nunes, P. C. A. & Hastenreiter, F. (2012) Gerenciamento de Projetos Segundo o Guia PMBOK: Desafios para os Gestores. *Revista de Gestão e Projetos – GeP*, São Paulo, 3(3) 58-87.
- Carvalho, B. V. & Mello, C. H. P. (2012) Aplicação do Método Ágil Scrum no Desenvolvimento de Produtos de Softwares em uma Pequena Empresa de Base Tecnológica. *Gestão & Produção*, 19(3), 557-573.
- Carvalho, M. M. & Rabechini Jr., R. (2015). *Fundamentos em Gestão de Projetos: Construindo Competências para Gerenciar Projetos*. 4 ed. São Paulo: Atlas.
- Cassel, C. & Symon, G. (1994). *Qualitative Methods in Organizational Research*. London: Sage Publication.

- Cruz, G. (2013). Scrum e PMBoK Unidos no Gerenciamento de Projetos. Rio de Janeiro: Brasport.
- Framework de Maturidade para mais Agilidade nas Empresas (2011), Revista Mundo Project Management, 36. Disponível em: <http://www.mundopm.com.br/pres_reved36.shtml> Acesso em 04 de agosto de 2016.
- Guntamukkala, V.; Wen, H. J. & Tarn. (2006). Na Empirical Study of Selecting Software Development Life Cycle Models. Human Systems Management, 4), 265-278.
- Hicks, M. & Foster, J. S. (2015) Adapting Scrum to Managing a Research Group. Department of Science, University of Maryland. Disponível em: <http://www.cs.umd.edu/~mwh/papers/score.pdf> Acesso em 08 de agosto de 2015.
- Kardec, M. S. (2012). Estudo de Compatibilidade entre PMBoK e Scrum. Revista Tecnologias e Projeção, 3(1), 1-7.
- Lalsing, V; Kishnah, S. & Pudaruth, S. (2012). People Factors in Agile Software Development and Project Management. International Journal of Software Engineering & Applications, 3(1) 117-137.
- Leite, L. M. & Lucrédio, D. (2014). Desenvolvimento de Software utilizando o Framework Scrum: Um Estudo de Caso. T.I.S. São Carlos, 3(2), 114-121.
- Machado, M. & Medina, S. G. (2009). Scrum – Método Ágil: Uma Mudança Cultural na Gestão de Projetos de Desenvolvimento de Software. Revista Científica Intraciência, Faculdade do Guarujá – UNIEESP, 1(1), 58-71
- Maximiano, A. C. A. (2010). Administração de Projetos: Como Transformar Ideias em Resultados. 4 ed. São Paulo: Atlas.
- PMBoK Guide (2013): A Guide to the Project Management Body of Knowledge. PMI. 5 ed.
- Rising, L. & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. IEEE Software, 17(4), 26-32.
- Sabbagh R. (2013). Scrum: Gestão Ágil para Projetos de Sucesso. São Paulo: Casa do Código.
- Salo, O. & Abrahamsson, P. (2008). Agile Methods in European Embedded Software Development Organisations. IET Software, 2(1), 58-64.
- Sanders, D. (2007). Using Scrum to Manage Student Projects. Journal of Computing Sciences in Colleges, 23(1), 69-79.
- Schwaber, K. & Sutherland, J. (2015). Um Guia Definitivo para o Scrum – As Regras do Jogo. Julho de 2013. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em 06 de agosto de 2015.
- Schwaber, K. (2004). Agile Project Management with Scrum. Redmond: Microsoft Press.
- Schwaber, K. (2007). The Enterprise and Scrum. Redmond: Microsoft Press.
- Silva, E. C. & Gil, A. C. (2013). Inovação e Gestão de Projetos: Os Fins Justificam os Meios. Revista de Gestão e Projetos – GeP, São Paulo, 4(1) 138-164.
- Terlizzi, M. A. & Biancolino, C. A. (2014). Projeto de Software no Setor Bancário: Scrum ou Modelo V. TAC, Rio de Janeiro, 4(1), 46-58.
- Thiollent, M. (2005). Metodologia da Pesquisa-Ação. 14. ed. São Paulo: Cortez.
- Vijayarathy, L. R. & Turk, D. Agile Software Development: A Survey of Early Adopters. Journal of Information Technology Management, XIX(2), 1-8.