

Proposta de implantação de práticas do *Scrum* em microempresa de Pato Branco/Paraná

Lucas Rodrigo Martinello

Universidad San Carlos (USC)
Asunción, Paraguay

lucas.martinello@gmail.com

Resumo. O *Scrum* tem se tornado uma das vertentes ágeis mais empregadas no ramo de desenvolvimento de software, seja por sua flexibilidade seja por sua estabilidade. Suas entregas rápidas e sua metodologia de desenvolvimento enxuto tem atraído cada vez mais os olhares dos empreendedores. Em Pato Branco, no sudoeste do Paraná, as empresas estão gradativamente implantando suas práticas com sucesso. Assim, considerando a possibilidade de implantação para empresas com metodologia própria, este trabalho apresenta uma proposta para implantação em uma microempresa da cidade, líder nacional em seu segmento, com o mapeamento das práticas ágeis nos processos atuais da empresa sugerindo adaptações. A validação da proposta foi realizada por meio de um questionário on-line com seus colaboradores visando investigar as necessidades atuais e a aceitação dos mesmos para com as práticas ágeis. Os resultados demonstram a viabilidade da implantação, com adaptações, do *Scrum* e uma ampla concordância por parte dos colaboradores.

1 INTRODUÇÃO

Segundo o Standish Group International, em pesquisa realizada com 40 mil projetos de Tecnologia da Informação no mundo, 18% dos projetos fracassaram, 43% foram concluídos com algum sucesso, mas falharam em custo/prazo/qualidade, e apenas 39% tiveram êxito (THE STANDISH GROUP, 2013).

Visando reduzir essas margens de falhas e de resultados comprometidos, empregam-se metodologias de gestão de projetos para administrar as variáveis concernentes a um projeto e aumentar a probabilidade de seu sucesso (CAMILLERI, 2011).

As metodologias ágeis fornecem esse suporte de gestão e empresas do sudoeste do Paraná estão voltando esforços para construir processos estáveis que garantam qualidade, custo e prazo aceitáveis. Essas empresas estão visualizando novos horizontes para o desenvolvimento de *software*. Nos

últimos anos o número de implantações de metodologias ágeis aumentou pela necessidade de gerenciar os processos de desenvolvimento. Dentre as metodologias que elas têm empregado destacam-se o *Scrum* e o XP (*Extreme Programming*), o primeiro para gestão e o segundo com técnicas de engenharia de *software*.

Esse trabalho tem como objetivo principal analisar a viabilidade e apresentar uma proposta de implantação das práticas do *Scrum* na Empresa utilizada para o estudo de caso realizado. Foi desenvolvido em uma microempresa de *software* de Pato Branco aqui denominada “a Empresa”, nome fictício para resguardar a organização. A Empresa atua desde 1997 no ramo de desenvolvimento e é referência nacional para *softwares* de fonoaudiologia e fisioterapia, porém no momento não aplica técnicas das metodologias ágeis em seus processos.

2 O SCRUM

O *Scrum* é uma metodologia ágil de gestão de projetos. Ele pode ser aplicado em diversas áreas, porém neste trabalho o foco de utilização é *software*. Muitos conceitos do *Scrum* são antigos e remetem à metodologia *Lean*, de 1948. Essa metodologia, ou sistema Toyota de produção, surgiu no Japão na fábrica automotiva Toyota, depois da Segunda Guerra Mundial. Era um tempo difícil para a economia japonesa, que vivia com baixa produtividade e falta de recursos, logo havia dificuldades para produzir em massa.

Lean foi criada para sanar esses impedimentos com conceitos e valores vinculados a: construir apenas o que é necessário, parar algo se está errado, eliminar tudo que não agregue valor, respeitar os trabalhadores, utilizar toda a capacidade da carga produtiva, incentivar os trabalhadores pela atribuição de responsabilidade e autoridade, eliminar o desperdício, aumentar o *feedback*, ter comprometimento com o tempo, produzir entregas rápidas, construir a integridade do sistema, dar poder ao time e enxergar o todo (POPPENDIECK; POPPENDIECK, 2003).

Em 1986 surgiu o primeiro estudo mencionando o *Scrum*. Ele foi publicado por Hirotaka Takeuchi e Ikujiro Nonaka na Harvard Business Review. O artigo, intitulado “*The New New Product Development Game*”, descreve uma

abordagem holística na qual os times são pequenos e multifuncionais, trabalhando com sucesso em busca de um objetivo comum, com flexibilidade e agilidade (CLARK; WHEELWRIGHT, 1995).

O conceito do *Scrum* envolve o desenvolvimento iterativo e incremental trabalhando em diferentes partes do sistema ao mesmo tempo, com objetivos claramente definidos e constantemente atualizados e reorganizados de acordo com suas prioridades, uma mescla de *Lean*, desenvolvimento iterativo e os estudos de Takeuchi e Nonaka.

O *Scrum* se baseia em entregas contínuas, ou seja, a cada período específico de tempo (quatro semanas, por exemplo) tem-se um produto com valor de negócio significativo codificado, testado e aprovado que é entregue ao cliente. Essa não é uma tarefa fácil, exige disciplina e comprometimento, mas é possível (NEELY; STOLT, 2013).

O *Scrum* trabalha com papéis, artefatos e reuniões.

Os papéis são: Product Owner, Scrum Master e Equipe de Desenvolvimento.

Os artefatos são: Product Backlog, Sprint Backlog, Burndown Chart e Quadro Kanban.

Os eventos são: Sprint, Pré-Game, Sprint Planning Meeting, Daily Scrum Meeting, Sprint Review Meeting, Sprint Retrospective Meeting e Release.

O *Scrum* inicia com o *Product Owner* conhecendo o negócio do cliente, a montagem da equipe de desenvolvimento e o *Scrum Master* ensinando *Scrum* para a equipe (se necessário). Em seguida é realizada a primeira reunião com todos. Com um prévio *Product Backlog* construído se definem quantas *sprints* e *releases* serão necessárias e a cada quanto tempo são realizadas entregas para o cliente.

Começa-se a primeira *sprint* com primeira reunião *Sprint Planning Meeting*, as reuniões diárias acontecendo todos os dias e, no último dia da *sprint*, ocorrem as reuniões *Sprint Review Meeting* e *Sprint Retrospective Meeting*. No próximo dia inicia outra *sprint* e, caso dividida por *releases*, pode-se realizar as reuniões de planejamento, de revisão e de retrospectiva para a *release*.

3 DESENVOLVIMENTO DE *SOFTWARE* NA EMPRESA

Nessa parte do trabalho é apresentada a análise de uma microempresa de Pato Branco/PR, aqui denominada a Empresa, e o seu modelo de desenvolvimento de *software*.

3.1 Sobre a Empresa

A Empresa foi fundada em 1997 por dois sócios e conta hoje com mais de oito mil clientes em vinte e seis estados brasileiros e em mais de quinze países.

Conta hoje também com mais de trinta *softwares* para as áreas de fonoaudiologia, fisioterapia, psicopedagogia, educação (aprendizagem) e agências de publicidade. Seus *softwares* são citados frequentemente em artigos e publicações de revistas dessas áreas.

A Empresa tem hoje dez colaboradores, incluindo os dois sócios, sendo eles alojados em setores: direção (2 pessoas), comercial (1 pessoa), suporte técnico (2 pessoas) e análise e desenvolvimento (5 pessoas).

3.2 Modo de trabalho

A Empresa tem seu modo de trabalho orientado por projetos, logo um desenvolvedor pode ser alocado em mais de um projeto simultaneamente, caso necessário.

Os cinco desenvolvedores desempenham três projetos (ano de 2014), sendo que dois trabalham em um projeto *web*, um trabalha em um projeto *desktop* e quatro trabalham em outro projeto *web*.

Os desenvolvedores são divididos em duas equipes: equipe de desenvolvimento *web* com dois colaboradores e equipe de desenvolvimento de jogos com três colaboradores.

No Projeto 1 estão alocados os dois desenvolvedores *web*, no Projeto 2 está um desenvolvedor de jogos e no Projeto 3 está um desenvolvedor *web* e os três desenvolvedores de jogos.

Dessa forma, a Empresa consegue ter vários projetos em paralelo e maximizar o tempo de cada desenvolvedor alocando ele no projeto em que há mais urgência no momento.

Cada projeto passa pelas seguintes fases: levantamento de requisitos, projeto e análise, desenvolvimento, testes e implantação.

4 PROPOSTA DE IMPLANTAÇÃO DO SCRUM NA EMPRESA

4.1 Proposta com mapeamento

Após analisar o modelo de desenvolvimento da Empresa, mudanças que venham a somar para aperfeiçoar o processo de trabalho e reduzir os problemas são sugeridas. A Empresa utiliza modelo próprio, mas não emprega nenhuma metodologia ágil. No Quadro 1 são apresentadas as sugestões de utilização das práticas ágeis do *Scrum* em comparação com o modelo atual de desenvolvimento da Empresa.

Item	Modelo atual	Sugestão com <i>Scrum</i>
Número de membros por equipe.	De 1 a 4.	Sem alteração, pois <i>Scrum</i> suporta equipes de um membro a nove membros (recomenda-se equipes de sete a nove pessoas cada).
Eventos: <i>sprints</i> e <i>releases</i> .	É desenvolvido por ciclos, porém sem duração fixa ou preestabelecida.	Definir as durações das <i>sprints</i> (ciclos) e das <i>releases</i> (entregas ao cliente). Sugestões: Projeto pequeno (6 meses): <i>sprints</i> de 1 semana e <i>releases</i> de 1 mês; Projeto médio (12 meses): <i>sprints</i> de 2 semanas e <i>releases</i> de 2 meses; Projeto grande (18, 24 meses ou mais): <i>sprints</i> de 4 semanas e <i>releases</i> de 4 meses.
Evento: <i>pré-game</i> ou <i>pré-planejamento</i> .	Diretores se reúnem com líder do projeto e cliente para levantar os requisitos.	Utilizar a reunião de levantamento de requisitos para isso, mas reunir todos os envolvidos (equipe, <i>Product Owner</i> , <i>Scrum Master</i> , direção e cliente) e definir as datas das entregas, construir o <i>Product Backlog</i> preliminar (lista de requisitos) e priorizá-la, levantar os riscos e ver a viabilidade.
Papel: <i>Product Owner</i> .	O líder da equipe e os diretores colhem o conhecimento com o cliente.	Designar uma pessoa na empresa, como é pequena, para ser o <i>Product Owner</i> de toda a empresa e conhecer as necessidades dos clientes dos projetos executados simultaneamente. Essa pessoa se encarregaria somente desta função e estaria sempre disponível às equipes.
Papel: <i>Scrum Master</i> .	O líder da equipe hoje faz esse papel conciliando atividades técnicas e	Embora o <i>Scrum Master</i> não seja um gerente, a empresa precisa de uma pessoa para resolver os impedimentos e

	gerenciais.	fazer o <i>Scrum</i> acontecer na empresa, ou seja, ele deve trabalhar com execução e qualidade. Como ela é pequena sugere-se que o <i>Product Owner</i> acumule a função de <i>Scrum Master</i> , porém precisará ter discernimento para não misturar as funções. Futuramente, quando for possível treinar duas pessoas, as funções se dividiriam.
Papel: equipe de desenvolvimento.	Hoje a equipe se divide entre os projetos.	Sugere-se que durante uma <i>sprint</i> cada membro permaneça focado em um só projeto. Após isso poderá alternar para outro projeto e fazer a próxima <i>sprint</i> nele. Caso seja necessário ter um membro parcial (um período do dia ou um dia sim e outro não) é necessário haver uma excelente comunicação entre a equipe (todos precisam conhecer tudo sobre o sistema) e, em casos de falta de informações, o <i>Scrum Master</i> precisa estar atento e conhecer tudo o que está acontecendo para auxiliar.
Artefato: <i>Product Backlog</i> .	Há uma lista de requisitos de posse do líder da equipe a partir do que o cliente solicitou.	O <i>Product Owner</i> (nova função) deverá gerenciá-lo, alterar diretamente com o cliente observando as necessidades dele e não somente adicionar o que é solicitado, repassar à equipe nas reuniões e solucionar as dúvidas ao longo do dias.
Artefato: <i>Sprint Backlog</i> .	Há uma lista de funcionalidades para cada ciclo, porém esse ciclo não é fixo e a lista sofre alterações.	Criar uma lista de funcionalidades para o ciclo (<i>sprint</i>) em comum acordo entre o <i>Product Owner</i> e a equipe e essa lista não poderá sofrer alterações durante o ciclo.
Artefato: <i>Burndown Chart</i> .	Não é construído. Diretores contatam o líder da equipe para acompanhar o que está acontecendo, porém isso limita a visão da direção sobre o andamento do projeto em comparação com o planejado e facilmente sai fora do estimado, gerando atrasos.	Construção em um quadro pendurado na parede em um lugar visível para todos. Sua atualização pode ser realizada na reunião diária (<i>Daily Scrum Meeting</i>).
Artefato: Quadro <i>Kanban</i> .	Não é construído. Os diretores contatam o líder da equipe para acompanhar o que está acontecendo, porém isso limita a visão da direção sobre o andamento do projeto	Construção em um quadro pendurado na parede como o <i>Burndown Chart</i> . Sua atualização pode ser feita conforme a equipe desenvolve os itens do <i>Sprint Backlog</i> .

	e gera frequentes interferências da direção na equipe de desenvolvimento.	
Evento: <i>Sprint Planning Meeting</i> .	Existe uma reunião antes de cada fase para definir o que será desenvolvido.	Após definir as datas e durações das <i>sprints</i> , fazer a reunião na data com toda a equipe mais membro <i>Product Owner</i> (e <i>Scrum Master</i>). Na primeira parte da reunião, pela manhã, o <i>Product Owner</i> explicará as funcionalidades e entrará em comum acordo com a equipe de quais itens serão desenvolvidos na <i>sprint</i> e, na tarde, a equipe dividirá os itens em sub-funções até o ponto em que consiga definir tudo o que será realizado nas próximas duas semanas.
Evento: <i>Daily Scrum Meeting</i> .	Não é feito.	Reunir toda a equipe no início da manhã, das 9h às 9h15, para cada um expor o que fez no dia anterior, o que fará no atual e se há impedimento para fazer. O <i>Scrum Master</i> estará junto para auxiliar e resolver os impedimentos. Se houver necessidade de uma conversa prolongada sobre algum item, os desenvolvedores responsáveis podem ficar debatendo após esse período, enquanto os outros retornam às suas atividades.
Evento: <i>Sprint Review Meeting</i> .	Não é feito.	Fazer uma reunião no último dia da <i>sprint</i> , pela manhã, com a presença do <i>Product Owner</i> (e <i>Scrum Master</i>) e representantes do cliente (opcional) com a conferência de funcionalidades. Se o <i>Product Owner</i> e <i>Scrum Master</i> forem a mesma pessoa recomenda-se utilizar representante do cliente para ele conferir com mais liberdade. Se forem pessoas diferentes pode ser o <i>Product Owner</i> .
Evento: <i>Sprint Retrospective Meeting</i> .	Não é feito.	Fazer uma reunião no último dia, pela tarde, com a presença do <i>Scrum Master</i> e a equipe. Se o cargo do <i>Product Owner</i> for outra pessoa (não for o <i>Scrum Master</i>) ele pode estar, mas apenas como observador. Nessa reunião a equipe criará um documento de melhores práticas com o que foi bem na <i>sprint</i> e pode ser continuado, o que pode ser melhorado e o que deve ser descartado.

Quadro 1 – Mapeamento do *Scrum* na Empresa
Fonte: o autor.

Conforme ilustrado no Quadro 1 é possível implantar as práticas ágeis do *Scrum* no atual modelo da Empresa com algumas adaptações.

4.2 Pesquisa

A fim de validar esse mapeamento fez-se uma pesquisa junto aos colaboradores da Empresa para levantar o nível de aceitação das práticas ágeis. Foram realizadas 17 questões sobre o modo atual de trabalho e sugestão de aplicação das práticas ágeis. O questionário foi aplicado aos colaboradores por meio do envio de *links* eletrônicos. Os resultados são exibidos na sequência.

Todos os colaboradores destacaram que, na maioria das vezes, conseguem entregar as tarefas dentro do prazo estipulado. A Tabela 1 apresenta as respostas.

Tabela 1 – Tarefas dentro do prazo

Tarefas dentro do prazo	Número de respostas
Sim, sempre	0
Sim, na maioria das vezes	5
Não, geralmente não dá tempo	0

Fonte: o autor.

A maioria dos colaboradores aponta alterações de escopo com muita frequência e alguns apontam com alterações com frequência. A Tabela 2 apresenta as respostas.

Tabela 2 – Alterações de escopo

Alterações de escopo	Número de respostas
Acontecem com muita frequência, a cada pouco precisamos parar o que fazemos para alterar funcionalidades	3
Acontecem com frequência, toda semana há alterações de funcionalidades nas quais estávamos trabalhando	2
Temos um pacote de trabalho. As mudanças só acontecem quando terminamos o pacote de trabalho	0
O escopo de trabalho é bem definido e gerenciado. As mudanças são controladas e acontecem em marcos do projeto	0

Fonte: o autor.

A empresa não tem mecanismos de cronograma e cada colaborador faz da sua maneira, como ilustram as respostas obtidas que constam na Tabela 3.

Tabela 3 – Cronograma de tarefas

Cronograma de tarefas	Número de respostas
Sim, tenho um cronograma de tarefas realizadas, em andamento e pendentes	2
Sim, sei o que preciso fazer, mas não tenho cronograma do que foi feito	1
Não, faço conforme me é requisitado, mas não tenho controle do que foi feito e do que falta fazer	2

Fonte: o autor.

As entregas para a maioria não são planejadas, acontecendo conforme o ritmo da equipe em desenvolver as funcionalidades (Tabela 4).

Tabela 4 – Entregas de *software*

Entregas de <i>software</i>	Número de respostas
As entregas de <i>software</i> funcionando são planejadas e ocorrem dentro do prazo estabelecido	0
As entregas de <i>software</i> funcionando são planejadas, mas geralmente não conseguimos entregar dentro do prazo	1
As entregas de <i>software</i> funcionando não são planejadas, entregamos conforme conseguimos fazer	4
As entregas de <i>software</i> funcionando ocorrem em longos períodos (meses), não são planejadas e geralmente o cliente pede muitas alterações quando elas acontecem	0

Fonte: o autor.

Todos afirmam que, embora haja explicação das funcionalidades, as dúvidas permeiam a equipe gerando mudanças constantes (Tabela 5).

Tabela 5 – Resolução de dúvidas

Resolução de dúvidas	Número de respostas
Todas as nossas dúvidas são sanadas rapidamente	0
Geralmente sim, em poucos dias temos todas as dúvidas resolvidas	0
Geralmente não, ficamos com dúvidas sobre as funcionalidades e isso acarreta mudanças constantes	5
Não, existem explicações, mas geralmente ficam dúvidas não sanadas	0

Fonte: o autor.

Os dados da Tabela 6 mostram que a maioria dos colaboradores concorda que a equipe ora tem capacitação, mas sem disponibilidade e ora disponibilidade, mas sem capacitação adequada, logo geralmente há gargalos.

Tabela 6 – Capacitação e disponibilidade de recursos humanos

Equipe de trabalho tem capacitação e disponibilidade para realizar todas as tarefas	Número de respostas
Sim, temos toda a capacitação necessária e sempre tem alguém disponível para realizar a tarefa	0
Geralmente sim, temos capacitação, mas falta disponibilidade	3
Geralmente sim, temos disponibilidade, mas nem sempre pessoal qualificado para a demanda, o que faz as tarefas demorarem para serem concluídas	2
Geralmente não, a equipe tem capacitação suficiente e sempre faltam recursos humanos para realizar as tarefas dentro do tempo	0
Não, a equipe necessita urgentemente de capacitação e novos membros para dar conta da demanda de tarefas	0

Fonte: o autor.

Parte dos colaboradores apontou (dados na Tabela 7) que não há reuniões de planejamento e monitoramento de atividades, mas parte indica que sim, mesmo não ocorrendo em datas estipuladas. Isso se explica pelo fato de haver mais de um projeto, então uma equipe realiza as reuniões e outra não, porém não é padrão da empresa.

Tabela 7 – Reuniões de planejamento e monitoramento

Reuniões de planejamento e monitoramento	Número de respostas
Acontecem com frequência semanal ou mensal	0
Acontecem eventualmente, mas sem datas estipuladas	2
Há reunião de planejamento, mas não há reunião de monitoramento das atividades	1
Não há reunião de planejamento e também não há reunião de monitoramento das atividades	2

Fonte: o autor.

Em relação ao tamanho das entregas, entre optar por um pequeno conjunto de funcionalidades e entregas de *software* em tempo curto ou por um grande conjunto de funcionalidades e entregas em longos períodos, a maioria preferiu entregas eventuais e algumas entregas rápidas, mas nenhum respondeu entregas longas ou entrega única, mostrando a preferência por ciclos curtos, como mostram os dados da Tabela 8.

Tabela 8 – Tamanho das entregas de *software*

Tamanho das entregas de <i>software</i>	Número de respostas
Entregas rápidas (2 a 4 semanas)/conjunto pequeno de funcionalidades	2
Entregas eventuais (4 a 8 semanas)/conjunto médio de funcionalidades	3
Entregas longas (8 a 16 semanas)/conjunto grande de funcionalidades	0

Sobre se a existência de uma pessoa responsável pelas funções de *Scrum Master* seria bom para sua equipe com as funções de:

- a) Remover os impedimentos do dia-a-dia (adquirir/trocar equipamentos, conseguir ajuda técnica para dificuldades, fazer contato com gerentes e clientes, etc.);
- b) Proteger a equipe da influência externa/pressão do cliente ou distrações;
- c) Resolver todas as necessidades não relacionadas ao trabalho de desenvolvimento que gastam o tempo dos membros.

Todos concordaram que seria relevante a existência desse papel, como indicam os dados da Tabela 9.

Tabela 9 – Relevância do *Scrum Master*

Relevância do <i>Scrum Master</i>	Número de respostas
Sim, facilitaria muito nosso trabalho	5
Sim, poderia ajudar em alguns momentos	0
Não, não temos necessidade disso	0

Fonte: o autor.

A existência de responsável pelas funções de *Product Owner* é de total concordância ser relevante, como ilustram os dados da Tabela 10. As funções desse papel são:

- a) Fazer o contato com o cliente e conhecer a fundo as necessidades dele;
Transmitir informações detalhadas sobre todas as funcionalidades do sistema para a equipe;
- b) Criar a lista de funcionalidades que o sistema deverá conter e repassar/discutir ela com a equipe;
- c) Sanar todas as dúvidas sobre as regras de negócio do sistema;
- d) Acompanhar a equipe e testar as funcionalidades após codificadas.

Tabela 10 – Relevância do *Product Owner*

Relevância do <i>Product Owner</i>	Número de respostas
Sim, facilitaria muito nosso trabalho	5
Sim, poderia ajudar em alguns momentos	0

Os colaboradores, em sua maioria (como indicam os dados da Tabela 11), acreditam que se a equipe seguisse as funções segundo o *Scrum* seria bom para seu próprio desempenho. Essas funções são:

- a) Ser auto-organizada (a equipe mesmo decide quem fará cada funcionalidade conforme capacidades individuais);
- b) Ser auto-conduzida (a equipe mesmo decide como e quando fará cada funcionalidade);
- c) Responsável pela entrega do produto: a equipe se responsabiliza por entregar o *software* dentro do período que ela acredita ser suficiente para analisar/codificar/testar;
- d) Possuir as habilidades necessárias para desenvolver o sistema: a equipe contém membros com capacidades para trabalhar em todas as áreas do desenvolvimento (programação, banco de dados, *design*, projeto, análise, etc.).

Tabela 11 – Relevância das funções da equipe

Relevância das funções da equipe	Número de respostas
Sim, teríamos ganhos significativos de desempenho	4
Sim, melhoraria nosso trabalho	1
Não, não faria uma diferença significativa	0
Não, não temos necessidade disso	0

Fonte: o autor.

A maioria concordou que melhoraria as entregas do sistema se a equipe tivesse uma lista de itens priorizados a serem desenvolvidos para o sistema formada a partir dos requisitos do cliente e mantida por uma pessoa com conhecimento da área do cliente (dados da Tabela 12 ilustram as respostas).

Tabela 12 – Relevância do *Product Backlog*

Relevância do <i>Product Backlog</i>	Número de respostas
Sim, conseguiríamos entregar o que o cliente deseja dentro do prazo estipulado	3
Sim, conseguiríamos entregar mais próximo do que o cliente deseja e dentro do prazo estipulado	2
Não, não faria uma diferença significativa	0
Não, não temos necessidade disso	0

Fonte: o autor.

Houve uma equivalência de respostas sobre se à existência de uma forma gráfica para medir o que foi feito *versus* o que deveria ter sido feito estimularia a equipe a completar as tarefas e minimizaria os gerentes irem à equipe cobrar o que está sendo desenvolvido, como mostram os dados apresentados na Tabela 13. Isso indica insegurança da equipe quanto ao seu trabalho ser medido.

Tabela 13 – Relevância do *Burndown Chart*

Relevância do <i>Burndown Chart</i>	Número de respostas
Sim, conseguiríamos medir o desempenho e ter mais tranquilidade para trabalhar	2
Sim, teríamos uma visão mais macro de como está o trabalho	1
Não, não faria uma diferença significativa	1
Não, não temos necessidade disso	1

Fonte: o autor.

Sobre a relevância da *Daily Scrum Meeting*, foi questionado se os colaboradores acreditam que a reunião diária seria proveitosa para a equipe. Pelos dados da Tabela 14, a maioria (4 de 5) acredita que ela seria proveitosa.

Tabela 14 – Relevância da *Daily Scrum Meeting*

Relevância da <i>Daily Scrum Meeting</i>	Número de respostas
Sim, seria muito proveitoso a equipe saber o andamento das atividades individuais e um poder ajudar o outro	2
Sim, facilitaria para um ajudar o outro em suas dificuldades	2
Não, não faria uma diferença significativa em nosso monitoramento de atividades	1
Não, apenas gastaríamos mais tempo com uma reunião	0

Fonte: o autor.

A maioria acredita que uma reunião de planejamento no início da *sprint* seria proveitosa para a equipe, pois poderiam trabalhar sem mudanças de requisitos, como apontam os dados apresentados na Tabela 15.

Tabela 15 – Relevância da *Sprint Planning Meeting*

Relevância da <i>Sprint Planning Meeting</i>	Número de respostas
Sim, a equipe teria um controle das tarefas que se espera dela e o tempo que deve realizar	2
Sim, a equipe conseguiria trabalhar sem mudanças de funcionalidades no ciclo	3
Não, não faria uma diferença significativa em nosso planejamento	0
Não, apenas gastaríamos mais tempo com uma reunião	0

Fonte: o autor.

A maioria das pessoas (4 de 5, como ilustram os dados da Tabela 16) acredita que a reunião de revisão no final da *sprint* seria proveitosa para a equipe.

Tabela 16 – Relevância da *Sprint Review Meeting*

Relevância da <i>Sprint Review Meeting</i>	Número de respostas
Sim, a equipe poderia verificar o porquê não conseguiu realizar as atividades e as inconformidades em relação ao que o cliente esperava	2
Sim, a equipe poderia encontrar as falhas de escopo e o cliente efetuar testes no sistema	2
Não, não faria uma diferença significativa em nossa revisão	1
Não, apenas gastaríamos mais tempo com uma reunião	0

Fonte: o autor.

Uma reunião de retrospectiva no final da *sprint* seria proveitosa para a equipe na visão da maioria dos colaboradores, pois auxiliaria a encontrar falhas e realizar as devidas correções (Tabela 17).

Tabela 17 – Relevância da *Sprint Retrospective Meeting*

Relevância da <i>Sprint Retrospective Meeting</i>	Número de respostas
Sim, certamente a equipe melhoraria suas práticas de desenvolvimento	2
Sim, provavelmente a equipe poderia encontrar suas falhas e corrigir o rumo	3
Não, não faria uma diferença significativa fazer uma retrospectiva	0
Não, apenas gastaríamos mais tempo com uma reunião	0

Fonte: o autor.

Por fim percebe-se que as respostas do questionário indicam falhas de planejamento, cronogramas mal estipulados e do excesso de alterações nos requisitos durante o desenvolvimento, itens estes que teriam maiores possibilidades de resolução com a implantação do *Scrum*.

Também se percebe que quase todas as práticas do *Scrum* foram bem aceitas pelos colaboradores, com exceção do *Burndown Chart* pelo efeito de medição que ele gera, porém acredita-se que com sua implantação a equipe iria usufruir dos seus benefícios.

5 CONCLUSÃO

Este trabalho dissertou sobre o *Scrum* e teve como objetivo analisar a viabilidade e apresentar uma proposta de implantação das práticas do *Scrum*

em uma microempresa de Pato Branco/PR. O objetivo foi atendido com a apresentação de uma proposta de mapeamento das práticas do *Scrum* de acordo com a realidade da empresa.

Além disso, foi realizada uma pesquisa com os colaboradores de desenvolvimento da Empresa para averiguar a aceitação deles quanto as práticas e quais os principais descontentamentos. Descobriu-se uma ampla aceitação sobre as práticas e que os descontentamentos, conforme citados anteriormente, são solucionáveis com a implantação do *Scrum*.

Desse modo, acredita-se que não somente nesta empresa, mas nas demais da região sudoeste do Paraná que ainda relutam em aplicar um *framework* ágil, devem abrir horizontes e portas para as metodologias que podem auxiliar na resolução de dificuldades de gerenciamento de projetos que são enfrentadas pelas empresas.

REFERÊNCIAS

CAMILLERI, Emanuel. **Project Success: Critical Factors and Behaviours**. Londres: Gower Publishing Company, 2011.

CLARK, Kim B.; WHEELWRIGHT, Steven C. **The Product Development Challenge: Competing Through Speed, Quality, and Creativity**. Estados Unidos da América: Harvard Business Press, 1995.

NEELY, Steve; STOLT, Steve. **Continuous Delivery? Easy! Just Change Everything (well, maybe it is not that easy)**. Agile Conference, p. 121-128, 2013.

POPPENDIECK, Mary; POPPENDIECK, Tom. **Lean Software Development: An Agile Toolkit**. Estados Unidos da América: Addison-Wesley Professional, 2003.

THE STANDISH GROUP. **CHAOS Report**, 2013. Disponível em: <<http://versionone.com/assets/img/files/CHAOSManifesto2013.pdf>>. Acesso em: 10 dez. 2014.