

# Aplicação do método ágil *scrum* no desenvolvimento de produtos de *software* em uma pequena empresa de base tecnológica

## *Implementation of scrum agile methodology in software product project in a small technology-based company*



Bernardo Vasconcelos de Carvalho<sup>1</sup>  
Carlos Henrique Pereira Mello<sup>1</sup>

**Resumo:** Este trabalho apresenta o resultado de uma pesquisa-ação, realizada em uma pequena empresa de base tecnológica, na qual se aplicou o método ágil *Scrum* em um projeto de desenvolvimento de um produto de *software*. A empresa objeto desta atua em Itajubá/MG e seus principais produtos são sistemas de *software*. Estudos indicam que a indústria de produção de *software* é ineficiente e ineficaz. E as micro e pequenas empresas de base tecnológica (MPEBT) têm um desafio ainda maior devido aos seus recursos restritos. Além disso, os métodos tradicionais de desenvolvimento de produtos de *softwares* demandam muitos custos. Tendo em vista a importância estratégica das MPEBT no desenvolvimento regional, seria importante que o *Scrum* fosse compatível com seus processos, para que elas pudessem se tornar mais competitivas e usufruir de seus benefícios. O objetivo deste trabalho foi analisar a implantação do método ágil *Scrum* nos projetos de desenvolvimento de novos produtos de *software* de uma MPEBT, além de compreender e mensurar o impacto desta implantação na empresa. Concluiu-se que os resultados alcançados sugerem que o método melhorou a comunicação e aumentou a motivação do time, diminuiu o custo, o tempo e o risco do projeto e aumentou a produtividade da equipe. Com esses resultados alcançados, a organização tende a se tornar mais competitiva, pois a bem-sucedida gestão de desenvolvimento de produtos é ponto crucial para o sucesso de uma empresa de base tecnológica.

**Palavras-chave:** *Scrum*. Desenvolvimento ágil de produtos. Desenvolvimento de produtos de *software*. Gestão de projetos de *software*.

**Abstract:** *This study presents the result of an action research that was carried out in a small technology-based company, in which the Scrum agile methodology was applied in software product project. The company object of this research operates in Itajubá/MG, and its main products are software systems. Studies have shown that the software industry is inefficient and ineffective. Micro and small technology-based companies have an even greater challenge, considering their limited resources. Furthermore, traditional methods of software product development are expensive. Considering that the strategic importance of micro and small technology-based companies in regional development, Scrum should be compatible with their processes, so that they could become more competitive and reap its benefits. The objective of this paper is to monitor and assist the implementation of Scrum agile methodology in new software product development in a small technology-based company to understand and measure the impact of the use of this methodology in the company. The final results of the research show that this methodology increased the team's motivation, reduced the cost, time, and risk of the project, and increased productivity. Therefore, with these results, the organization tends to be more competitive since a successful product development management is crucial to the success of a technology-based company.*

**Keywords:** *Scrum*. Agile product development. Software product development. Software project management.

## 1 Introdução

No atual ambiente de desenvolvimento de *software*, os requisitos estão sujeitos a frequentes alterações durante o ciclo de desenvolvimento do produto para atender às alterações da demanda (RISING; JANOFF, 2000). Este fato torna o desenvolvimento de *software*

um desafio, principalmente para as pequenas empresas, tendo em vista seus recursos restritos.

A produção de *software* no Brasil ainda esbarra em barreiras como a pirataria e altas cargas tributárias. Segundo a Associação Brasileira das Empresas

<sup>1</sup> Núcleo de Otimização da Manufatura e de Tecnologia da Inovação – NOMATI, Instituto de Engenharia de Produção e Gestão – IEPG, Universidade Federal de Itajubá – UNIFEI, Av. BPS, 1303, Pinheirinho, CEP 37500-903, Itajubá, MG, Brasil, e-mail: bernardo@b2ml.com.br; carlos.mello@unifei.edu.br

de *Software* (2008), 87% dos *softwares* instalados na base de computadores no Brasil em maio de 2008 eram piratas. Além disso, pode-se dizer que a indústria de *software* de um modo geral é ineficaz e extremamente ineficiente.

Dados do Standish Group, que pesquisou mais de 30 mil projetos norte-americanos de desenvolvimento de produtos de *software* desde 1994, revelam que a taxa de sucesso para projetos acima de US\$ 10 milhões (que envolvem mais de quinhentas pessoas, por pelo menos três anos), é estatisticamente nula (JOHNSON, 1995). Já para pequenos projetos, de até US\$ 750 mil, a taxa de sucesso é 55%. No ano de 1994, esses projetos tinham seus custos 189% e durações 222% maiores que os planejados. Além disso, somente 61% das funcionalidades previstas eram de fato implementadas e a taxa de sucesso (produtos de *software* que são realmente utilizados pelo usuário final) média era de somente 16% (JOHNSON, 1995).

No ano 2000, o aumento médio do custo ficou em 45%, o atraso médio foi de 63%, e 67% das funcionalidades previstas foram entregues. Esta grande melhoria reflete o aumento da preocupação, atenção e da competência mundial em relação ao desenvolvimento de produtos de *software* (JOHNSON, 2001).

Entretanto, em 2003 alguns indicadores pioraram. Os atrasos aumentaram para 82% e as funcionalidades implementadas desabaram para 52%. Mas, a taxa de sucesso aumentou consideravelmente para 34% dos projetos e o atraso médio caiu para 43% (JOHNSON et al., 2003 apud JØRGENSEN; MOLØKKEN, 2006).

Em meados da década de 1990, surgiram técnicas de desenvolvimento ágil de produtos de *software*. Esta disciplina foi fortemente influenciada pelas melhores práticas da indústria japonesa, particularmente pelos princípios da manufatura enxuta implementados pelas companhias Honda e Toyota e pelas estratégias de gestão do conhecimento de Takeuchi e Nonaka (2004) e Senge (1990).

Nesse contexto, destaca-se o *Scrum*, uma abordagem enxuta de gerenciamento de projetos de desenvolvimento de produtos. Este processo foi desenvolvido por Jeff Sutherland em 1993, juntamente com Mike Beedle e Ken Schwaber, baseado num artigo de Takeuchi e Nonaka (1986) sobre as vantagens dos pequenos times no desenvolvimento de produtos. Originalmente, seu foco era somente o desenvolvimento de *software*, embora atualmente ele seja aplicado ao desenvolvimento de produtos de maneira geral (CRISTAL; WILDT; PRIKLADNICKI, 2008).

Apesar de ser uma abordagem relativamente nova, a utilização do método *Scrum* tem aumentado nos últimos anos, impulsionado pelas recentes pesquisas que mostram que seu uso aumenta a satisfação dos clientes e diminui o atraso em projetos em relação aos métodos tradicionais (MANN; MAURER, 2005).

Além do problema quanto ao custo dos projetos de *software*, existe também a necessidade de se melhorar a eficácia dos produtos. Os sistemas de *software* em si estão passando a ser a ferramenta mais básica e fundamental dos processos econômicos e sociais. Atualmente, é improvável que exista alguma atividade econômica que não sofra nenhum efeito de um sistema de informação. Por isso, é de suma importância que eles sejam eficazes.

Esse conjunto de fatores que marca a economia realça também o interesse sobre o papel que as micro e pequenas empresas de base tecnológica podem ter no desenvolvimento de regiões e países.

Pode-se afirmar que ainda são incipientes os estudos sobre o processo de desenvolvimento de produtos em empresas de base tecnológica não só no Brasil, mas em todo o mundo (LEDMITH, 2000; SOUDER; SHERMAN; DAVIES-COOPER, 1998). De acordo com Carvalho et al. (2000) e Gonzalez e Toledo (2012), há uma carência tanto teórica quanto prática na forma com que as empresas de base tecnológica realizam o desenvolvimento de seus produtos. Desta forma, existe uma demanda reprimida de pesquisas com o foco na gestão dos projetos de desenvolvimento de produtos em pequenas empresas de base tecnológica.

Não há atualmente na literatura sobre *Scrum* trabalhos que trazem dados concretos sobre o impacto de sua implantação com relação aos aspectos definidos para o presente estudo. Isto faz com que se considere relevante a contribuição científica deste trabalho.

Visando contribuir com a base de conhecimento para preencher esta lacuna identificada, o presente trabalho tem como objetivos: analisar a implantação do método ágil *Scrum* em um projeto de desenvolvimento de um novo produto de *software* em uma pequena empresa de base tecnológica; e compreender e mensurar o impacto desta implantação com relação à comunicação, colaboração, produtividade e motivação do time, custos e tempo do projeto e gerenciamento dos riscos.

## 2 Indústria de *software* do Brasil

Sob o ponto de vista mercadológico, o mercado brasileiro de *software* e serviços é bastante expressivo. Um estudo da Associação Brasileira das Empresas de *Software* (2008) aponta que em 2007, o País ocupou a 12ª posição no mercado mundial de *software* e serviços relacionados, tendo movimentado aproximadamente 11,12 bilhões de dólares, equivalentes a 0,86% do PIB brasileiro daquele ano. Deste total, foram movimentados US\$ 4,19 bilhões em *software*, o que representou perto de 1,6% do mercado mundial. Os restantes US\$ 6,93 bilhões foram movimentados em serviços relacionados. Em 2008, houve um aumento de 35%, o que resultou em US\$ 15 bilhões. Este resultado manteve o mercado brasileiro na 12ª posição

no cenário mundial, respondendo por 1,68% (*software*) e 1,72% (serviço) do cenário global.

Ainda, de acordo com a Associação Brasileira das Empresas De *Software* (2008), este mercado é atendido por 7.937 empresas dedicadas ao desenvolvimento, produção e distribuição de *software* e de prestação de serviços. Outro dado importante é que das empresas que atuam no desenvolvimento e produção de *software*, 94% são classificadas como micro e pequenas empresas. O mesmo estudo aponta para um extraordinário crescimento médio anual estimado superior a 10% até 2010, a despeito da crise econômica mundial de 2008/2009.

Segundo pesquisas nacionais sobre esse tema (FERNANDES; CÔRTEZ; OSHI, 2000; SEBRAE; INSTITUTO..., 2001; MACULAN, 2003; TOLEDO et al., 2008), as empresas de base tecnológica brasileiras de menor porte possuem, principalmente, as seguintes características: operam em pequena escala; são comprometidas com o projeto; realizam desenvolvimento e produção de novos produtos de alto conteúdo tecnológico que, na maioria dos casos, não são produtos finais, mas em geral, bens de capital, componentes e sistemas industriais; e, servem a mercados restritos e específicos (nichos de mercado), normalmente atuando com substituição de importações.

O Brasil tende a tornar-se um importante exportador de *software*. Segundo dados da IT WEB (2009), em 2008 o País exportou US\$ 340 milhões em *software*. A mesma publicação aponta que o mercado de TI movimentou US\$ 61 bilhões na América Latina em 2008 e que o Brasil respondeu por 48% desse montante. Embora este número seja pequeno comparado ao mercado mundial de TI, que movimentou US\$ 1,08 trilhão em 2005.

Em vista de tudo isso, os métodos ágeis surgem como uma boa proposta para gerenciar os projetos de desenvolvimento de *software* das pequenas empresas de base tecnológica.

### 3 Métodos ágeis

Ao longo dos anos, vários métodos de desenvolvimento de produtos foram apresentados. Entre eles, existem os chamados métodos ágeis (AMBLER, 2002), também chamados de métodos leves (FOWLER, 2000). Estes métodos são mais adaptativos e flexíveis em relação aos tradicionais. Além disso, eles são indicados para cenários em que existe constante mudança de requisitos e os resultados devem ser entregues em pequenos espaços de tempo.

Geralmente, esses métodos dividem o desenvolvimento em diversas iterações de ciclos mais curtos (no caso do *Scrum* e para o presente trabalho, estes ciclos são chamados de *Sprints*) e realizam entregas ao final de cada uma delas, de forma que “[...] o cliente (interno ou externo) receba uma versão que

agregue valor ao seu negócio.” (DANTAS, 2003, p.37). Assim, as mudanças de requisitos podem ser acompanhadas pelos desenvolvedores no início de cada ciclo. E existe uma retroalimentação por parte do cliente para a equipe de desenvolvimento, o que reduz o risco do projeto.

Esses métodos foram fortemente influenciados pelas melhores práticas da indústria japonesa, particularmente pelos princípios da manufatura enxuta implementados pelas companhias Honda e Toyota e pelas estratégias de gestão do conhecimento de Takeuchi e Nonaka (2004) e Senge (1990).

Os métodos tradicionais de desenvolvimento têm o foco na geração de documentação sobre o projeto e no cumprimento rígido de processos. Já os métodos ágeis concentram as atenções na constante entrega do produto (MUNDIM et al., 2002) e nas interações entre os indivíduos. Nestes métodos, é minimizada a fase de planejamento inicial, de modo que os desenvolvedores se concentram em entregar o produto ao fim de cada iteração, ao invés de traçar diretrizes e planejamentos para o projeto como um todo.

Nos anos recentes, os métodos ágeis de desenvolvimento de *software* tem ganhado grande popularidade. Entretanto, existem poucos estudos empíricos sobre o assunto. Uma recente revisão bibliográfica sistemática sobre o tema (DYBÅ; DINGSØYR, 2008) encontrou 1.996 artigos na área. Destes, apenas 36 eram estudos empíricos, com aceitáveis rigor metodológico, credibilidade e relevância, o que representa apenas 1,8% dos trabalhos.

Existem diversos métodos ágeis, além do *Scrum*: a Programação extrema, o *Feature Driven Development*, *Dynamic Systems Development Method*, *Adaptive Software Development*, *Crystal*, *Pragmatic Programming* e *Test Driven Development*. Para o desenvolvimento deste trabalho, o método escolhido foi o *Scrum*, que será apresentado em detalhes no tópico a seguir.

## 4 *Scrum*

### 4.1 História

O método *Scrum* segue os princípios do Manifesto Ágil (MANIFESTO ÁGIL, 2001) e tem como pai três de seus signatários: Mike Beedle, Ken Schwaber e Jeff Sutherland. Segundo Schwaber e Beedle (2002), ele tem como objetivo definir um processo de desenvolvimento de projetos focado nas pessoas da equipe.

O nome *Scrum* surgiu da comparação entre desenvolvedores e jogadores de *Rugby*. *Scrum* é a denominação da rápida reunião que ocorre quando os jogadores de *Rugby* vão iniciar um lance. A primeira utilização deste termo surgiu em um estudo de Takeuchi e Nonaka (1986), no qual os autores

notaram que pequenos projetos com equipes pequenas e multifuncionais obtinham os melhores resultados. Esta analogia foi usada porque no *Rugby* cada time age em conjunto, como uma unidade integrada. Nele, cada membro desempenha um papel específico e todos se ajudam em busca de um objetivo comum. E assim devem ser os times de desenvolvimento que adotam o método *Scrum*. Ele baseia-se em algumas características (SCHWABER, 1995): flexibilidade dos resultados, flexibilidade dos prazos, times pequenos, revisões frequentes e colaboração.

## 4.2 Benefícios

A literatura pesquisada mostra que a utilização deste método pode gerar benefícios como aumento da satisfação de clientes, por meio da diminuição das reclamações (MANN; MAURER, 2005; SALO; ABRAHAMSSON, 2008); melhoria na comunicação e aumento da colaboração entre envolvidos nos projetos (BERCZUK, 2007); aumento do retorno do investimento em projetos de novos produtos (SULAIMAN; BARTON; BLACKBURN, 2006; SUTHERLAND, 2005); aumento da motivação da equipe de desenvolvimento de produtos (KNIBERG; FARHANG, 2008; PAASIVAARA; DURASIEWICZ; LASSENIUS, 2008); melhoria da qualidade do produto produzido (SUTHERLAND et al., 2008; BARTON; CAMPBELL, 2007); diminuição dos custos de produção (SUTHERLAND et al., 2007; BRUEGGE; SCHILLER, 2008); aumento de produtividade da equipe de desenvolvimento (SUTHERLAND et al., 2008; MARÇAL et al., 2007); diminuição no tempo gasto para terminar projetos de desenvolvimento de novos produtos (SUTHERLAND et al., 2008; SANDERS, 2007); e diminuição do risco em projetos de desenvolvimento de novos produtos (EDWARDS, 2008).

## 4.3 Práticas e artefatos do *Scrum*

Este método não requer ou fornece qualquer técnica específica para a fase de desenvolvimento, apenas estabelece conjuntos de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto.

O ponto inicial do *Scrum* é o *Backlog* do Produto, sendo considerada a prática responsável pelo armazenamento e gerenciamento dos requisitos coletados, conforme aponta Schwaber e Beedle (2002). Nesta prática, por meio de reuniões com todos os envolvidos, investidores, clientes e parceiros no projeto, são apontadas todas as necessidades do negócio e as funcionalidades a serem desenvolvidas. Assim, o *Backlog* do Produto é uma lista de funcionalidades, ordenadas por prioridade, que provavelmente serão desenvolvidas durante o projeto.

A reunião diária de *Scrum* (*Daily Scrum*) é um rápido encontro que ocorre entre os membros do

time para definir quais serão as tarefas do dia e saber os resultados das tarefas do dia anterior. Esta reunião é também chamada de *Stand Up Meeting* (reunião em pé), já que é de praxe que todos os membros a realizem de pé, de forma a conseguir maior agilidade. Três perguntas são respondidas por cada membro sobre suas responsabilidades (RISING; JANOFF, 2000): O que foi feito ontem? O que será feito hoje? Há algum obstáculo à realização das atividades?

Na reunião diária de *Scrum*, os membros do time não respondem a essas perguntas como forma de prestar contas à gerência, mas sim como formalização do comprometimento com o resto da equipe. Assim, todos os membros do time conhecem as metas individuais de cada integrante, conhecem seus impedimentos (riscos) e podem cobrar compromissos assumidos.

O *Sprint* é considerado a principal prática do *Scrum*. É o período de tempo no qual são implementados os itens de trabalho definidos no *Backlog* do Produto pela equipe *Scrum*. Conforme Abrahamsson et al. (2002), ele normalmente dura de uma a quatro semanas, mas não há uma regra para isto; as equipes que decidem a duração a ser adotada para o projeto. No caso do desenvolvimento de *software*, o *Sprint* inclui as fases tradicionais do desenvolvimento de *software*: requisitos, análise, projeto e entrega.

O *Backlog* do *Sprint* é um subconjunto do *Backlog* do Produto. Ele é uma lista de atividades a serem desenvolvidas durante o *Sprint*. Sua definição acontece durante a Reunião de Planejamento do *Sprint*. Já a Reunião de Revisão do *Sprint* (*Sprint Review Meeting*) é a reunião que acontece após cada *Sprint*. Nela, a equipe discute sobre seus erros, acertos e lições aprendidas.

No início do projeto, cliente e desenvolvedores definem o *Backlog* do Produto (sua lista de requisitos). Também são estimados os custos do projeto e definidas as datas para entrega de resultados a partir da priorização mais favorável ao cliente. Uma análise inicial de riscos é preparada. As ferramentas de trabalho e os integrantes das equipes são escolhidos. Um dos desenvolvedores é eleito "*Scrum Master*", cujo papel se assemelha a um gerente de projetos (embora existam diferenças cruciais entre um *Scrum Master* e um Gerente de Projetos).

O *Scrum Master* trabalha para que o processo *Scrum* aconteça e para que não existam impedimentos para os membros da equipe realizarem seu trabalho. Remover os obstáculos apontados na reunião de *Scrum* diária é seu dever, de modo que os desenvolvedores se concentrem apenas nas questões técnicas. Esses obstáculos são colocados em uma lista chamada de *Backlog* de Impedimentos, que fica à vista de todos.

Outro papel importante no método é o do Dono do Produto (*Product Owner*). Este membro do time geralmente representa o cliente (interno ou externo).

Ele define quais são os requisitos e qual é o grau de importância e prioridade de cada um deles. Este membro necessita conhecer muito bem as regras de negócios do cliente, de forma que ele possa tirar qualquer dúvida que o time possa ter em relação às funcionalidades do produto.

No início de cada *Sprint*, quando as equipes fazem a lista das atividades que precisam ser realizadas naquele *Sprint* (*Backlog* do *Sprint*), as responsabilidades são distribuídas. Os desenvolvedores discutem os padrões que serão adotados e as atividades de análise, codificação e testes se iniciam. Ao final de cada *Sprint*, uma versão do produto (no caso do produto de *software*, um executável do *software*) é apresentada ao cliente para obter a retroalimentação. Os defeitos encontrados são adicionados ao *Backlog* do produto. Ao longo de todo o projeto, são aplicados mecanismos de gerência *Scrum*, como o acompanhamento de alguns controles. A quantidade de funcionalidades não entregues, a necessidade de mudanças para corrigir defeitos ou para atualização tecnológica, os problemas técnicos encontrados e os riscos e as estratégias para evitá-los são exemplos de controles observados durante o desenvolvimento.

O último artefato do *Scrum* é o gráfico *burndown*. Trata-se de uma representação gráfica do trabalho restante em comparação com o trabalho já realizado. Geralmente, coloca-se a quantidade de trabalho no eixo vertical e o tempo no eixo horizontal. Ele é muito útil para prever quando todo o trabalho será completado e para alertar o time em caso de atraso (que ficará bastante aparente). Geralmente é traçada uma linha com a representação da execução do trabalho. Esta linha representa o esforço já realizado na execução das tarefas. Espera-se que a execução das atividades (tarefas) leve a linha de início em Y ao encontro de X. Este encontro representa o término das execuções das tarefas.

#### 4.4 Críticas

Apesar de bem aceito na indústria de *software*, o *Scrum* também sofre grandes críticas e é questionado quanto ao seu domínio de aplicação. Segundo alguns acadêmicos mais tradicionais (GREGÓRIO et al., 2007), ele tem como principais pontos fracos a falta de escalabilidade para equipes grandes e geograficamente dispersas e a necessidade da mudança de cultura da instituição.

Entretanto, a primeira crítica foi negada empiricamente por Paasivaara, Durasiewicz e Lassenius (2008), que mostrou que o *Scrum* foi usado com sucesso em diversos projetos de grandes dimensões e cujos times estavam distribuídos em diversas plantas empresariais. A segunda crítica procede, já que uma das grandes barreiras para a

implantação do *Scrum* é a necessidade de se mudar a cultura da gestão de projetos.

#### 4.5 Publicações científicas

É notável o grande aumento das publicações sobre *Scrum* ao longo dos anos. A literatura sobre *Scrum* é escassa, mas está em franca expansão (CARVALHO; MELLO, 2009). Ainda segundo Carvalho e Mello (2009), os benefícios do *Scrum* mais citados na literatura são os seguintes (em ordem de número de citações): melhoria na comunicação e aumento da colaboração entre envolvidos; melhoria da qualidade do produto produzido; aumento de produtividade da equipe; aumento da satisfação de clientes (diminuição de reclamações); aumento do retorno do investimento do projeto; aumento da motivação da equipe de desenvolvimento; diminuição dos custos de produção (mão de obra); diminuição no tempo gasto para terminar o projeto (prazo); diminuição do risco do projeto (menor possibilidade de insucesso).

#### 4.6 O uso do *Scrum* no presente trabalho

O presente trabalho adaptou para a realidade da empresa estudada o método de Kniberg (2007). Em sua obra, ele explica exatamente quais itens de verificação devem ser observados para garantir a correta implantação do *Scrum* em uma instituição. O presente trabalho considera que um item está implementado após a equipe realizar cada uma das ações mostradas no Quadro 1.

### 5 Método da pesquisa

Pode-se classificar o presente trabalho como de natureza aplicada, devido ao interesse prático na aplicação de seus resultados na resolução de problemas reais. Quanto aos seus objetivos, a presente pesquisa é descritiva, pois visou descrever as características do objeto de estudo e estabelecer relações entre as variáveis. Sua abordagem é, em sua maior parte, qualitativa, pois a maioria dos resultados da pesquisa-ação não pode ser mensurada numericamente e há uma relação dinâmica, entre o objeto de estudo e o pesquisador, que não pode ser traduzida em números.

Neste estudo, utilizou-se a estratégia de pesquisa-ação. Trata-se de uma sistemática de pesquisa social com base empírica concebida e realizada em associação com uma ação ou com a resolução de um problema coletivo no qual os pesquisadores e os participantes representativos da situação ou do problema estão envolvidos de modo participativo (THIOLLENT, 2005). A fase exploratória da pesquisa durou de janeiro a abril de 2008. A fase de planejamento, de abril a junho de 2008. A primeira iteração foi de junho a novembro

**Quadro 1.** Time: ações a serem tomadas.

1. Time (membros do time)	1.1. Trabalham lado a lado. Este aspecto é importante para melhorar a comunicação e aumentar a sinergia no time.
	1.2. Colaboram entre si para implementar os requisitos do <i>backlog</i> do <i>Sprint</i> .
	1.3. Têm capacidade de desempenhar várias funções. Deve-se evitar uma superespecialização do time, de modo que um componente não saiba realizar a tarefa de outros.
	1.4. Admitem quando têm problemas e pedem ajuda ao <i>Scrum Master</i> quando isto acontece.
	1.5. Ajudam-se mutuamente. Eles precisam entender que as metas do projeto são todas coletivas: não há vitória ou derrota individual.
	1.6. Aceitam responsabilidades e se comprometem. O time deve se autogerenciar, o que torna esta característica fundamental.
	1.7. Não necessitam fazer hora extra sistematicamente, permitindo que tenham uma vida social agradável e menos estressante.
2. <i>Scrum Master</i>	2.1. Todos os times têm um.
	2.2. Trabalha ao lado de seu time e está presente quando solicitado.
	2.3. Tem como prioridade máxima resolver os impedimentos da equipe (ele trabalha para resolver os itens do <i>backlog</i> dos impedimentos).
3. Dono do produto ( <i>Product owner</i> )	3.1. Todos os times têm um. Está sempre disponível para o time tirar dúvidas quanto às regras de negócio.
	3.2. Entende o produto e as necessidades do cliente (ou do público alvo do produto) a ponto de tirar dúvidas da equipe sobre seu funcionamento esperado em cada situação.
	3.3. Entende o produto e as necessidades do cliente (ou do público alvo do produto) a ponto de saber quais são as funcionalidades prioritárias e, portanto, que devem ser implementadas primeiro.
	3.4. Tem o poder de fazer o time trabalhar primeiramente para implementar as funcionalidades prioritárias.
4. <i>Backlog</i> do produto ( <i>Product backlog</i> )	4.1. O dono do produto tem total controle sobre o <i>Backlog</i> do Produto, podendo alterar prioridades e colocar novos itens.
	4.2. Ele está facilmente acessível e visível para todo o time em qualquer momento.
	4.3. Ele é atualizado após cada <i>Sprint</i> , durante a reunião de revisão do <i>Sprint</i> .
	4.4. Contém somente funcionalidades do produto e nada mais. O <i>Backlog</i> do Produto não contém nenhuma tarefa para o time.
	4.5. Cada funcionalidade do <i>Backlog</i> do Produto contém uma definição própria, de modo que não haverá dúvidas sobre considerá-la ou não implementada.
5. Estimativas	5.1. O dono do produto recebe da equipe estimativas de quantidade de trabalho de cada funcionalidade.
	5.2. O time tem liberdade de estimar e não sofre pressões externas para alterar estimativas.
	5.3. Todos os membros do time participam das estimativas.
6. Reunião de planejamento do <i>Sprint</i>	6.1. Todos os membros do time participam dela.
	6.2. Ela resulta em um plano do <i>Sprint</i> , com um <i>Backlog</i> do <i>Sprint</i> . Este <i>Backlog</i> do <i>Sprint</i> está corretamente priorizado de acordo com o dono do produto.
	6.3. Todos os membros do time devem concordar que o planejamento ficou realista. Caso contrário, a reunião deve continuar até se chegar a um consenso.
	6.4. Todos os membros do time se comprometem com o planejado.
7. <i>Sprint</i>	7.1. O time entrega um produto (ou protótipo) funcionando no final de cada <i>Sprint</i> .
	7.2. O time segue rigorosamente as prioridades do <i>Backlog</i> do <i>Sprint</i> .
	7.3. O time age corretivamente quando está atrasado.
	7.4. O time alerta o <i>Scrum Master</i> e o dono do produto quando há problemas.
	7.5. O time sabe quando encontrar informações detalhadas que sejam úteis para implementar cada funcionalidade.
	7.6. Os problemas são discutidos e resolvidos no momento em que ocorrem.
	7.7. A duração de cada <i>Sprint</i> é sempre a mesma durante um projeto.
	7.8. O intervalo entre dois <i>Sprints</i> é de, no máximo, um dia.
	7.9. Todos os envolvidos (incluindo clientes e outros times de outros projetos da empresa) sabem sobre o <i>Sprint</i> , seus prazos e quais são seus produtos finais.
	7.10. As funcionalidades que começam a ser implementadas num <i>Sprint</i> terminam no mesmo <i>Sprint</i> .

Quadro 1. Continuação...

8. Reunião diária (Daily scrum)	8.1. Acontecem no mesmo lugar e horário todos os dias.
	8.2. Começam e terminam pontualmente.
	8.3. Todos os membros do time estão presentes.
	8.4. Nela, todos os membros do time respondem às três perguntas: O que fiz ontem? O que farei hoje? O que está me impedindo de fazer o que é preciso?
	8.5. Nela não acontecem interrupções.
	8.6. O dono do produto a visita regularmente.
	8.7. Os membros da equipe buscam as tarefas. Cada um decide que tarefa irá fazer. Não é o <i>Scrum Master</i> quem delega as atribuições.
	8.8. Os membros da equipe cobram entre si a realização das tarefas.
9. Reunião retrospectiva	9.1. Acontece no final de cada <i>Sprint</i> .
	9.2. Todos os membros da equipe e o dono do produto participam.
	9.3. Resulta em sugestões concretas de melhoria.
	9.4. Algumas sugestões apontadas pela reunião retrospectiva são implementadas de fato.
10. <i>Backlog</i> de impedimentos ( <i>Impediment backlog</i> )	10.1. Todos os times têm um.
	10.2. Está visível para todos. Além disso, qualquer membro do time pode inserir novos impedimentos a qualquer momento com facilidade.
	10.3. Está sempre atualizado.
	10.4. Tem itens priorizados. O <i>Scrum Master</i> tem o objetivo de resolvê-los, primeiramente pelos mais importantes.
	10.5. Seus itens que não podem ser resolvidos pelo <i>Scrum Master</i> ou pela equipe são destinados para o dono do produto ou para a alta gerência da empresa. O time cobra diariamente pela resolução do problema.
11. Velocidade	11.1. A velocidade de desenvolvimento é mensurada.
	11.2. A velocidade é utilizada para ações corretivas.
	11.3. O registro da velocidade é utilizado para melhorar a estimativa da equipe em planejamentos posteriores.
12. Gráfico <i>burndown</i>	12.1. O time tem um gráfico <i>Burndown</i> .
	12.2. O gráfico <i>Burndown</i> é visível para toda a equipe.
	12.3. O gráfico <i>Burndown</i> é atualizado, no mínimo, diariamente pelo <i>Scrum Master</i> . Preferencialmente, ele é atualizado logo após a implementação de uma funcionalidade.
	12.4. O time age corretivamente caso o gráfico <i>Burndown</i> mostre que o desenvolvimento está fora do planejado.
13. <i>Backlog</i> do <i>sprint</i>	13.1. Todos os times têm um.
	13.2. É visível por toda a equipe.
	13.3. É atualizado diariamente (e facilmente) por toda a equipe.
	13.4. A estimativa de trabalho para as tarefas é atualizada diariamente.
	13.5. Apesar do <i>Backlog</i> do <i>Sprint</i> conter funcionalidades e tarefas, elas são facilmente distinguíveis.
	13.6. Nele estão bem claras quais tarefas foram originadas por cada funcionalidade.

do mesmo ano. A segunda, de dezembro a fevereiro de 2009. A iteração final, de março a maio de 2009. E a análise dos resultados foi de março a junho de 2009. Cada iteração mencionada corresponde a um ciclo do método da pesquisa-ação.

A unidade de análise selecionada para este trabalho foi a B2ML Sistemas, uma empresa de tecnologia da informação que oferece soluções e serviços de tecnologia sob demanda para clientes de diversos setores de atividade. Ela é uma empresa focada na pesquisa e desenvolvimento. Cerca de

60% do seu faturamento é destinado para P&D. Essas constantes atividades fizeram com que a empresa lançasse diversos produtos e serviços em tecnologia da informação. Todos os seus produtos foram desenvolvidos internamente e apresentam alto grau de inovação.

A B2ML pode ser considerada uma pequena empresa, já que seu faturamento anual é maior que R\$ 240 mil e menor que R\$ 2,4 milhões, de acordo com a lei complementar brasileira nº 123, de 14 de dezembro de 2006 (BRASIL, 2006), que

instituiu o Estatuto Nacional da Microempresa e da Empresa de Pequeno Porte.

A escolha da empresa como unidade de análise do presente trabalho foi motivada pelo seu destacado perfil, podendo-se dizer que o tipo de amostragem foi o casual simples. Outro motivo é a facilidade de acesso a dados e à ação que o pesquisador tem sobre a organização, já que ele ocupa cargo de direção executiva na referida empresa. Sendo assim, as ações de implantação podem ser efetivamente experimentadas e validadas em um típico ambiente empresarial.

## 6 Descrição da pesquisa-ação

### 6.1 Fase exploratória: identificação da situação-problema

Nesta fase, foi divulgada a proposta desta pesquisa à empresa e foi obtido o comprometimento dos participantes e interessados.

Foram realizadas entrevistas semiestruturadas com pessoas-chave que participam do processo de desenvolvimento de produtos da empresa. O objetivo destas entrevistas foi de complementar a análise de dados secundários disponibilizados pela empresa com a opinião e o depoimento dos indivíduos envolvidos sobre as práticas de desenvolvimento.

É interessante explorar o contexto histórico da empresa. Algumas semanas após sua fundação, a empresa havia feito, em parceria com um consultor independente, um processo de desenvolvimento de *software* baseado no desenvolvimento em cascata. Esta tentativa se mostrou um completo fracasso, já que os processos desenhados nunca saíram do papel. Esses processos se mostraram muito pesados e tornaram-se inviáveis as tentativas de sua implantação. Em entrevista, um dos colaboradores da empresa na época (que hoje trabalha em uma multinacional em São José dos Campos/SP) disse que:

Aqueles processos eram burocráticos demais. Não havia a menor necessidade de termos aquilo tudo naquele momento. Como o consultor era muito focado em qualidade de *software*, aquilo poderia até servir para empresas com grandes equipes e certificadas em CMMI e MPS.br. Mas se nós implementássemos aquele processo, iríamos ficar fazendo muita atividade burocrática e nenhum produto. (Antigo colaborador da B2ML).

Pelas outras entrevistas realizadas, o colaborador expressa o sentimento da maioria em relação aos primeiros processos definidos. Esta situação levou a empresa a ir para um outro extremo: a total falta de processos. O diretor de tecnologia comenta a situação:

Sabíamos que estávamos errados, mas não sabíamos que direção tomar e como melhorar. Os projetos estavam cada vez mais atrasados e os prazos

mais apertados. Por isso, ninguém tinha tempo de estudar maneiras de fazer direito. Todos estavam programando com a maior velocidade possível. (Diretor de Tecnologia da B2ML).

Apesar de ineficiente, caótico e com diversos problemas, pode-se dizer que naquele momento o processo de desenvolvimento de novos produtos da empresa analisada era eficaz. Isso porque a empresa constantemente colocava no mercado produtos de diversas linhas com resultados razoáveis.

Esta situação se estendeu até os primeiros meses de 2008, quando a INCIT (incubadora de empresas na qual a empresa residia) fez uma parceria com o Núcleo de Otimização da Manufatura e Tecnologia da Inovação (NOMATI) da UNIFEI. Este órgão criou na INCIT um Núcleo de Desenvolvimento de Produtos (NDP). O NDP visa mitigar os problemas correlatos ao gerenciamento de processos e o desenvolvimento de produtos encontrados nas empresas incubadas na INCIT.

Assim, o pesquisador, juntamente com a equipe do NDP (composta por oito pesquisadores), realizou um diagnóstico da empresa. Para o levantamento de dados, foram utilizados como instrumento principal: a aplicação de entrevista fundamentada em um roteiro estruturado, com questões semiabertas e abertas, elaboradas a partir de um modelo de referência para causas genéricas de problemas no processo de desenvolvimento de produtos; a utilização de anotações escritas e gravação de entrevistas; e consulta a documentos (relatórios diversos).

O diagnóstico fundamentou-se principalmente em fontes primárias e secundárias. Com a finalidade de minimizar estes desvios inerentes à coleta de dados, os registros das entrevistas foram encaminhados para posterior validação pelos entrevistados. O relatório foi desdobrado em duas partes complementares: caracterização da empresa e produto; e caracterização do processo de desenvolvimento de produto.

Percebeu-se que o processo de desenvolvimento de produtos da empresa era informal, sendo que cada produto era tratado como um projeto diferente e possuía um ciclo de desenvolvimento próprio. Portanto, não existia um padrão de desenvolvimento de produto específico, contribuindo para que ocorressem problemas para a correta definição de prazo e custo.

Desta maneira, apontou-se que a principal oportunidade de aperfeiçoamento em seu processo de desenvolvimento de produtos se encontrava no aspecto de controle. Os pesquisadores fizeram uma comparação entre os diversos métodos ágeis existentes. Neste caso, concluiu-se que o método de aplicação mais adequado seria o *Scrum*, que atende especificamente às necessidades de empresas da área de *software* e possui diversas vantagens, já abordadas anteriormente neste trabalho.



## 6.2 Planejamento da implantação do Scrum

No momento em que havia um claro diagnóstico sobre a realidade da organização, os pesquisadores iniciaram a prática, que ocorreu por meio de seminários para guiar a ação. Os seminários, operacionalizados pelos promotores da pesquisa, tinham o objetivo de definir os temas e problemas prioritários a serem investigados.

Esta fase foi composta por um grande conjunto de entrevistas individuais e coletivas e questionários aplicados a pessoas-chave da organização, que expuseram suas reclamações, constatações e sugestões. Todas estas informações coletadas entre os entrevistados serviram como base para o posterior debate em seminário.

Em abril de 2008, os pesquisadores do NDP se reuniram com o objetivo de detectar as principais dificuldades em relação à implementação do *Scrum* e divulgá-las aos integrantes da equipe de trabalho. Além disso, decidiu-se planejar e divulgar a realização de um estudo dirigido em *Scrum* para a equipe de trabalho. A equipe também decidiu implantar o método de maneira gradual em um projeto piloto. Escolheu-se como piloto um projeto realizado em parceria com a Universidade Federal de Itajubá (UNIFEI) e com o fomento da Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG).

Este projeto possuía como principal produto final um Sistema de *Software* para Gestão de Compras (comércio eletrônico *business-to-business*) que permite a integração entre empresas pertencentes a um arranjo produtivo local (APL) e seus fornecedores em um sistema *web* com foco em compras em grupo. A complexidade do desenvolvimento deste *software* deveu-se aos requisitos, que nem mesmo as empresas os tinham claramente definidos, já que não possuíam um conhecimento prévio do impacto da tecnologia de comércio eletrônico no setor de compras. Isto implicou em constantes alterações dos requisitos e, por consequência imediata, de todo o projeto. Este fato tornou este projeto um importante objeto de estudo para análise da aplicação do *Scrum* em ambientes de desenvolvimentos de pequenas empresas.

Os membros da equipe decidiram que o desenvolvimento do projeto seria feito utilizando-se o IDE “Eclipse” versão 3.2.1, a linguagem Java em plataforma WEB, fazendo uso de HTML. Foi dado um treinamento para todos os integrantes do projeto piloto. Devido às características inovadoras e singulares do método, os pesquisadores tomaram a decisão de realizar um treinamento de *Scrum* para melhor compreensão das atividades. O treinamento foi realizado na própria empresa e durou oito horas. Conforme já explicado anteriormente, a implantação do *Scrum* seguiu os passos do método proposto por Kniberg (2007).

## 6.3 Fase de ação: primeira iteração

A primeira iteração se iniciou com a definição de alguns prazos principais do *Scrum Master* e do dono do produto no projeto. A equipe do projeto era composta por seis pessoas, sendo: um doutor em logística; um engenheiro de produção; três engenheiros da computação; e um bacharel em ciência da computação.

O *Scrum Master* escolhido foi um dos engenheiros da computação. Esta escolha foi feita porque este era o membro da equipe que tinha mais conhecimento do método *Scrum* e tinha a maior experiência no gerenciamento de projetos de desenvolvimento de *software*.

O Dono do Produto escolhido foi o engenheiro de produção. Ele era o membro da equipe que tinha mais conhecimento prévio nas regras de negócio do cliente em potencial do sistema. Preferencialmente, este papel deveria ser exercido pelo próprio cliente. Entretanto, como no caso o sistema ainda não tinha um cliente (apenas clientes potenciais), a equipe optou por eleger um Dono do Produto entre seus próprios membros. Este ator criou o *Backlog* do Produto do projeto, que listava todos os requisitos desejáveis do sistema em ordem decrescente de prioridade.

Os pesquisadores, juntamente com o time, decidiram que na primeira iteração seria interessante implantar totalmente as seguintes práticas ágeis do *Scrum*: *Scrum Master*, Dono do Produto, *Sprint*, *Backlog* do Produto e *Backlog* do *Sprint*.

O item “time” foi parcialmente implementado. Eles trabalharam bem, colaborando entre si, e tinham grande capacidade de desempenhar várias funções simultâneas. Mas eles não trabalharam lado a lado (alguns membros do time trabalhavam na empresa em horários diferentes), faziam horas extras sistematicamente e tinham dificuldade em se autogerenciar (os membros sentiram uma grande dificuldade por não ter um gerente de projetos).

O item “*Scrum Master*” foi totalmente implementado. O membro do time designado para fazer este papel tinha um bom treinamento e interesse na área e desempenhou corretamente seu papel. O item “Dono do Produto” foi totalmente implementado. O membro do time designado para fazer este papel teve algumas dificuldades no início, mas conhecia muito bem o domínio do problema e, por isso, não teve dificuldades em ser um bom Dono do Produto.

O item “*Backlog* do Produto” foi totalmente implementado. Ele era controlado pelo dono do produto e estava facilmente acessível para todos os membros do time. O *Backlog* do Produto foi automatizado e implementado pelo *software* “*Trac Open Source Project*” (TRAC..., 2009). Este sistema foi projetado especialmente para a utilização em projetos de desenvolvimento de *software*. Ele se

integra com o controle de versões do *software*, o que facilita muito o trabalho dos programadores.

O item “Estimativas” não foi implementado. O time não tinha conhecimento técnico suficiente em estimativas de esforço de *software* para realizá-lo. Algumas ações foram tomadas no sentido de resolver esta deficiência da equipe, como a compra de livros e realização de pesquisas na área.

O item “Reunião de Planejamento do *Sprint*” foi totalmente implementado. Nesta reunião, a equipe definiu qual seria o *Backlog* daquele primeiro *Sprint* e também definiu sua data de término. Na *Sprint Review Meeting* do primeiro *Sprint*, a equipe apontou vários erros cometidos. O maior deles foi ter criado uma *Backlog* do *Sprint* muito grande, que acabou resultando em um atraso no *Sprint*.

O item “*Sprint*” foi parcialmente implementado, pois algumas de suas ações foram implementadas e outras não. Devido à realização do *Sprint*, foi observada uma melhoria no ânimo da equipe, justificada, segundo os membros, pelos resultados tangíveis emergidos em cada *Sprint*, enquanto que em projetos tradicionais são obtidos somente no fim. Ao longo da primeira iteração do projeto, foram realizados seis *Sprints*. Um dos grandes erros deste item foi o fato da duração do *Sprint* ter sido variada, o que é expressamente desaconselhado pelo modelo de implantação. Algumas funcionalidades foram iniciadas no primeiro *Sprint* e não acabaram nele, o que não é recomendado. Outro erro cometido foi o fato dos integrantes do time não seguirem as prioridades do *Backlog* do *Sprint*.

O item “Reunião Diária” foi parcialmente implementado. As reuniões diárias existiram, mas não eram feitas em horários fixos. Além disso, ela não foi realizada em alguns dias, o que tem de ser evitado. Outra falha foi que o time ainda tinha o hábito de esperar pelo gerente do projeto, ou seja, ainda não tinha a capacidade de se autogerenciar.

O item “Reunião Retrospectiva” não foi implementado. Só foi realizada uma reunião retrospectiva durante a primeira iteração. Devido ao atraso, o time acabou deixando de lado esta importante prática. As consequências disto foram a repetição dos mesmos erros em *Sprints* consecutivos e a diminuição da possibilidade de melhorias.

O item “*Impediment Backlog*” não foi implementado. O time não utilizou nenhuma técnica formal para catalogar os riscos e ameaças. O item “Velocidade” não foi implementado porque não se implementou o item “Estimativas”, que, na prática, é um pré-requisito. O item “Gráfico *Burndown*” não foi implementado também porque o item “Estimativas” é seu pré-requisito.

O item “*Backlog* do *Sprint*” foi parcialmente implementado. Ele era visível para toda a equipe, atualizado diariamente e bastante claro. Mas não

havia qualquer estimativa de esforço em cada tarefa. Conforme já foi explicado, o “*Backlog* do *Sprint*” foi automatizado pelo sistema *Trac*.

Pelos resultados encontrados, o pesquisador e a própria equipe entenderam que seria interessante continuar o processo de implantação do *Scrum* no projeto. Os acontecimentos estavam sugerindo que o método aumentou a motivação de seus membros e iria aderir muito bem à pequena equipe. A equipe também considerou que as reuniões diárias (*Daily Scrum*) aumentaram consideravelmente o controle do projeto e diminuíram seus riscos. Além disso, foi constatado que um bom planejamento do *Sprint* é crucial para seu sucesso.

#### 6.4 Fase de ação: segunda iteração

Para a segunda iteração, decidiu-se pela implantação total dos seguintes itens do *Scrum*: *Scrum Master*; Dono do Produto; Time; Reunião de Planejamento do *Sprint*; Reunião Retrospectiva; *Sprint*; Reunião Diária; *Backlog* do Produto; e *Backlog* do *Sprint*. Desta forma, para a total implementação do *Scrum*, somente os itens relacionados às Estimativas e ao *Backlog* de Impedimentos ficariam pendentes. Os itens *Scrum Master*, Dono do Produto, *Backlog* do Produto e Reunião de Planejamento do *Sprint* continuaram sendo implementados normalmente.

Na segunda iteração, o item “Time” foi totalmente implementado com sucesso. Percebeu-se que os membros do time conseguiram romper a barreira inicial da falta da figura do gerente de projetos e passaram a trabalhar bem com autossuficiência no gerenciamento, o que também diminuiu bastante a necessidade de realização de horas extras. Os membros também passaram a trabalhar no mesmo horário e, por isto, lado a lado.

Na segunda iteração, houve um esforço no sentido de se realizar as estimativas. Os membros do time que foram treinados na metodologia de pontos de função passaram a tentar estimar cada funcionalidade.

A equipe determinou que o tipo de contagem de ponto de função seria o de “Projeto de Desenvolvimento”. Em cada medição, era identificado o escopo de contagem e a fronteira da aplicação. Depois, dentro deste escopo, eram contadas as funções de dados para determinar a contribuição delas para a contagem de pontos de função não ajustada e as funções transacionais para determinar a contribuição delas para a contagem de pontos de função não ajustada. O valor do ajuste era calculado com base nas características do projeto e com base nos dados históricos da empresa. Assim, a equipe tinha a contagem de pontos de função ajustada para cada requisito.

Depois da respectiva implementação, era feita a comparação entre o estimado e o realizado, o que

serviu de grande treinamento para a equipe. Como essas estimativas eram somente internas, o dono do produto ainda não recebia a estimativa.

Apesar do esforço para estimar, ainda não foi possível medir a velocidade do desenvolvimento de maneira confiável. O gráfico *Burndown* também não foi feito, pois não se tinha confiança nas estimativas realizadas.

No item “*Sprint*”, foram corrigidos os problemas da primeira iteração. Os membros do time passaram a seguir rigorosamente as prioridades das funcionalidades e a terminar cada implementação de funcionalidade no mesmo *Sprint*. Outra grande novidade foi a duração do *Sprint*, que passou a ser uniforme, ao contrário da primeira iteração. Foram realizados seis *Sprints* de duas semanas cada um nesta iteração.

As reuniões diárias passaram a ser mais produtivas quando o time começou a se autogerenciar corretamente. Com isso eles passaram a buscar as tarefas e a cobrar entre si sua realização. Além disso, com o fato de que todos do time passaram a trabalhar no mesmo horário, ficou viável realizar a reunião sempre no mesmo horário e local.

As Reuniões de Retrospectiva passaram a ser realizadas rigorosamente ao final de cada *Sprint*. Nelas havia uma rodada de registro das lições aprendidas e sugestões de melhoria. A maioria das sugestões era de simples implementação e, por isso, elas geralmente eram implementadas no *Sprint* subsequente.

Na segunda iteração, passou-se a fazer uso do *Backlog* de impedimentos. Segundo o *Scrum Master*, embora à primeira vista este artefato parecesse pouco importante, sua utilização acabava por mitigar os riscos na prática, já que eles ficaram muito mais claros e eram lembrados diariamente. A única falha deste item foi não priorizar cada risco. O *Backlog* do *Sprint* continuou a ser implementado, mas, desta vez, com as estimativas juntamente com cada funcionalidade.

### 6.5 Fase de ação: iteração final

As práticas ágeis *Scrum Master*, Dono do Produto, Time, Reunião de Planejamento do *Sprint*, Reunião Retrospectiva, *Sprint*, Reunião Diária, *Backlog* do Produto e *Backlog* do *Sprint* continuaram sendo implementadas normalmente. E as práticas ágeis relacionadas às Estimativas e ao *Backlog* de Impedimentos foram implementadas pela primeira vez. Nesta iteração foram realizados seis *Sprints* de duas semanas cada um.

O *Backlog* de Impedimentos passou a ser totalmente implementado. Desta vez, conseguiu-se priorizar a lista de impedimentos de modo que o *Scrum Master* sabia qual deveria ser sua prioridade de ação. Na última iteração, todos sabiam que o maior desafio seriam as práticas ágeis de estimativas e velocidade. Por isso,

todos os esforços focaram resolver o problema da falta de mensuração de trabalho do projeto.

As estimativas que já haviam sido realizadas na segunda iteração se consolidaram e a equipe passou a confiar mais nelas, de modo que passou a ser publicada, juntamente com cada funcionalidade, sua estimativa de esforço em pontos de função. Escolheu-se utilizar a ferramenta de análise de pontos de função porque atualmente este é o instrumento mais utilizado por profissionais da área de *software* e em empresas de todos os seguimentos, como afirmam Vazquez, Simões e Albert (2007).

Com a estimativa baseada em pontos de função dando bons resultados, viabilizou-se a mensuração da velocidade do projeto. Com isso, a equipe ganhou uma grande confiança no cumprimento dos prazos e pôde agir corretivamente quando se viu desviando do planejado.

O gráfico *burndown* foi totalmente implementado. Ele era atualizado pelo *Scrum Master* cada vez que uma funcionalidade era implementada. Ele foi elaborado em uma planilha eletrônica, de modo a ficar o mais simples possível para o entendimento de todos.

Nesta fase, a falta de uma documentação adequada para os projetos B, C e D (vide Figura 1) prejudicou a realização de uma análise mais aprofundada sobre como os participantes de cada projeto contribuíram para a produtividade do projeto e como o escopo de cada projeto contribuiu com a produtividade. Esta é uma limitação do presente trabalho.

### 6.6 Fase de avaliação: análise dos resultados

Esta etapa final do processo de pesquisa-ação teve dois objetivos principais: verificar e mensurar os resultados das ações no contexto organizacional da pesquisa e suas consequências; e extrair ensinamentos (lições aprendidas) que serão úteis para continuar a experiência e aplicá-la em estudos futuros.

Em termos técnicos de desenvolvimento de *software*, a implementação do sistema foi realizada

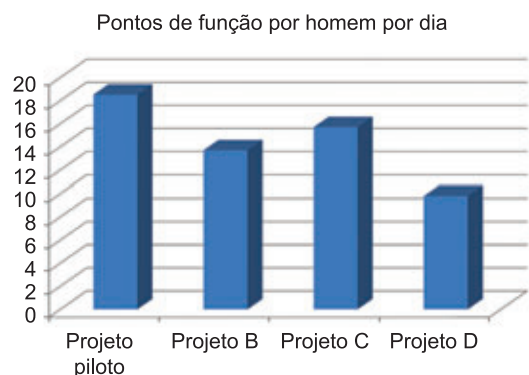


Figura 1. Produtividade por projeto.

no IDE “Eclipse”, versão 3.2.1. A codificação completa do projeto gerou 23 pacotes de classes, 89 classes, 893 métodos em classes, 8.930 linhas de código Java e 14.850 linhas de código HTML, que se configurou em um grande sistema para os padrões da empresa estudada.

A primeira análise feita na fase de avaliação foi do impacto do *Scrum* no desenvolvimento do produto. Para isso, foram elaboradas afirmações, baseadas nos nove indicadores de benefícios do *Scrum* que já haviam sido previamente levantados na literatura (CARVALHO; MELLO, 2009), e enviadas aos participantes da equipe de implantação. Essas pessoas deveriam ler cada afirmação e responder qual seu grau de concordância com ela, utilizando-se a escala de Likert, que variou de 1 (discordo plenamente) a 5 (concordo plenamente).

Dos nove indicadores, um é sobre reclamações de clientes, um sobre retorno do investimento e outro sobre melhoria da qualidade do produto. Sob o aspecto da qualidade do sistema desenvolvido, não se pode fazer análises, pois ele não foi avaliado pelos clientes. Isso também impede análises sobre o índice de reclamações. Além disso, sem o produto ir para o mercado, não se pode fazer inferências sobre o retorno do investimento do projeto.

Outros dois indicadores são sobre a produtividade e os custos de produção (mão de obra). Estes indicadores serão analisados posteriormente. Desta forma, foram desenvolvidas quatro afirmações, conforme mostra o Quadro 2. Os resultados são apresentados na Tabela 1.

Seis pessoas responderam ao questionário. Três delas participaram de todas as iterações, duas participaram das duas primeiras iterações e uma delas participou somente da terceira.

A afirmação que mais apresentou concordância por parte da equipe foi a melhoria da comunicação e colaboração entre o time. Acredita-se que este resultado se deva ao fato de que o método tem foco exatamente neste aspecto. Também as outras afirmações (2, 3 e 4)

**Tabela 1.** Média e desvio padrão das respostas.

Item	Média	Desvio padrão
1	4,67	0,52
2	4,00	0,63
3	4,17	0,75
4	4,17	0,41

**Quadro 2.** Afirmações sobre os benefícios do *Scrum*.

Item	Afirmações
1	O método <i>Scrum</i> melhorou a comunicação e a colaboração entre os envolvidos.
2	O método <i>Scrum</i> aumentou nossa motivação.
3	O método <i>Scrum</i> facilitou para que o projeto terminasse mais rápido.
4	O método <i>Scrum</i> diminuiu os riscos do projeto e as possibilidades de insucesso.

apresentaram um bom grau de concordância, o que leva à conclusão de que o time percebeu a maioria dos benefícios do *Scrum*.

Outra análise quantitativa realizada foi sobre o aspecto da produtividade das equipes. A Figura 1 mostra os dados de produtividade colhidos do projeto piloto em comparação com outros três projetos da empresa que estavam em andamento no mesmo período analisado. Estes dados mostram que o time do projeto analisado foi mais produtivo que os colaboradores que atuaram em outros projetos.

Mas, alguns fatores, além do método de desenvolvimento, causam grandes impactos na produtividade do time. Deve-se considerar que pode ter havido um certo “efeito Hawthorne”, ou seja, os colaboradores podem ter sido mais produtivos porque sabiam que estavam sendo analisados. Também se deve levar em consideração que os times dos projetos eram diferentes e isto também altera bastante a produtividade de um projeto.

Além disso, é importante analisar-se o grau de dificuldade do projeto. A Figura 2 mostra a classificação dos projetos, seguindo o modelo usado em Schwaber e Beedle (2002). Os projetos A (projeto piloto, com *Scrum*) B e C tinham a mesma tecnologia, mas tinham graus diferentes de precisão nos requisitos. Já o projeto D era bastante diferente dos outros em termos tecnológicos e de requisitos. Por todos esses fatores, não é prudente afirmar categoricamente que o *Scrum* melhorou a produtividade do time. Outros estudos são necessários para se chegar com precisão a esta conclusão. O que se pode dizer é que os dados levantados são indícios importantes de que a produtividade melhorou com o novo método.

Esses indícios se tornam mais fortes com as entrevistas feitas com o *Scrum Master* do projeto. Ele afirma que percebeu o aumento da produtividade do time, principalmente porque eles passaram a se sentir mais comprometidos com o sucesso do projeto. Como já foi dito, esta pessoa possuía boa experiência com projetos de *software* e, desta forma, sua opinião foi relevante.

Uma segunda análise mais aberta foi executada. O pesquisador realizou um seminário com toda a equipe do projeto para discutir e avaliar as práticas gerenciais e os artefatos do *Scrum*. Estas análises estão resumidas no Quadro 3.

Para a equipe, tomando como comparação o histórico de projetos, o *Scrum* foi um método

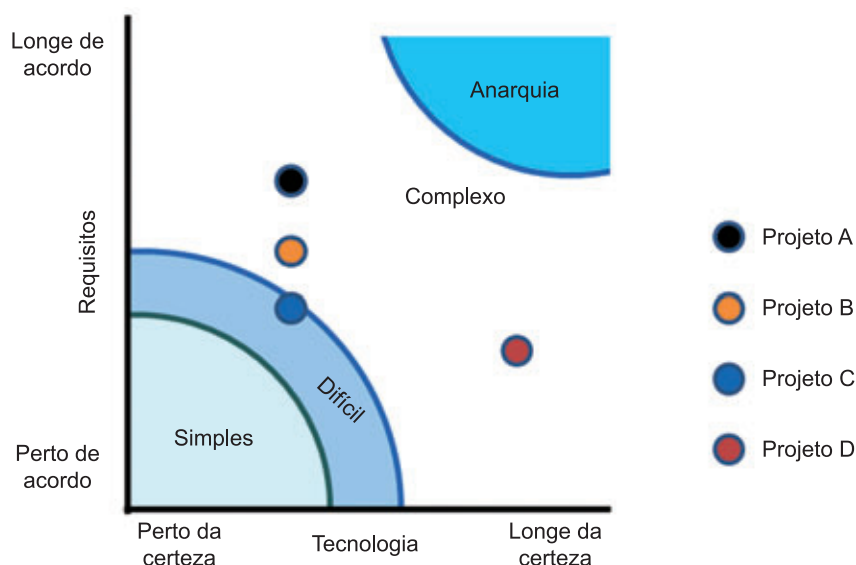


Figura 2. Classificação dos projetos. Fonte: Adaptado de Schwaber e Beedle (2002).

Quadro 3. Análise das práticas gerenciais do Scrum.

Item	Vantagens/Facilidades	Desvantagens/Dificuldades
Backlog do produto	Sua adoção melhorou o planejamento do time.	O time sentiu dificuldade para elaborar sua primeira versão que geralmente é muito extensa.
Reunião diária	Sua utilização aumentou muito o controle do projeto. Os riscos foram minimizados.	Houve algumas falhas de periodicidade devido a desencontros de horários dos membros da equipe.
Sprint	A divisão do projeto em Sprints aumentou a motivação do time.	Devido à inexperiência da equipe, alguns Sprints tiveram duração muito longa.
Planejamento do sprint	Sua realização melhorou o planejamento do Sprint.	As primeiras reuniões foram pouco produtivas devido à inexperiência da equipe.
Backlog do sprint	Sua boa elaboração é crucial para o sucesso ou fracasso do Sprint.	Seu superdimensionamento causou um atraso nos primeiros Sprints.
Revisão do sprint	Importante reunião para o recolhimento das lições aprendidas.	Tende a não ser executado pela equipe quando o time está ansioso para o planejamento do próximo Sprint.
Backlog de impedimentos	É uma boa ferramenta para cobrar a resolução de problemas conhecidos, diminuindo os riscos do projeto.	Tende a ser negligenciado pelo próprio Scrum Master, para evitar maiores responsabilidades.
Velocidade e estimativas	Muito importante para o controle do projeto e para o planejamento de custos da empresa.	Foi o item mais complexo de ser implementado e que exigiu mais treinamento e novos conhecimentos.
Gráfico burndown	É uma ferramenta visual interessante, que deixa claro o que os números já mostravam.	Tende a ser abandonado, pois, à primeira vista, parece informação redundante.
Dono do produto	Facilita para o time, que passa a ter um representante do cliente dentro da própria empresa.	Não foi o caso do projeto estudado, mas provavelmente é difícil encontrar uma pessoa que entenda realmente o negócio do cliente a ponto de fazer bem este papel.
Scrum master	É imprescindível para tirar dúvidas quanto ao processo de desenvolvimento e liderar as cerimônias do Scrum.	Muitas vezes o time o vê como um gerente do projeto, deixando para ele atribuições que deveriam ser do próprio time.

condizente à realidade da empresa, pois propôs um processo focado em resultados, na comunicação da equipe e interação com os clientes, sem desrespeitar as restrições enfrentadas por uma pequena empresa.

Uma terceira análise foi realizada em um segundo seminário no sentido de buscar lições aprendidas. Assim, como no estudo de Rayhan e Haque (2008), foi sentida uma grande necessidade do comprometimento do time na implantação do *Scrum*. Ficou claro que não é possível uma implantação deste tipo sem o forte apoio da equipe de desenvolvimento.

Além disso, a equipe considera que as seguintes lições foram aprendidas:

- É importante encontrar boas ferramentas computacionais para conseguir implementar o processo. A automação é importante para evitar que o método *Scrum* demande muito tempo do time, o que seria um contrassenso. A equipe deste projeto utilizou o *software* “*Trac Open Source Project*” (TRAC..., 2009) para automatizar parte das práticas ágeis;
- A implantação do *Scrum* de forma incremental e iterativa parece ser menos arriscada e traumática que uma implantação única e total;
- É preciso ter foco na construção de uma cultura de autogerenciamento. É necessário muito treinamento para convencer os integrantes do time de que eles são seus próprios gerentes. Esta é uma quebra de paradigma grande e custosa. Não é exagero reforçar esta mensagem em cada reunião do time;
- Existe a necessidade de quebrar logo no início barreiras hierárquicas e de relacionamento no time. Todos necessitam sentir-se membros do time, com grandes responsabilidades e no mesmo nível. Inclusive, considerou-se uma boa prática encorajar as pessoas a chamarem membros do time pelo primeiro nome, evitando tratamentos especiais e pomposos;
- É imprescindível que exista transparência entre o *Scrum Master*, o Dono do Produto e o resto do time. Deve-se ter comunicação sem limites entre eles e uma relação de sinceridade; e
- Durante o trabalho, foi proposto entretenimento após o horário de trabalho diversas vezes. O time considerou estas atividades extremamente positivas para o relacionamento. Se divertir junto deu mais sentimento de cumplicidade ao time, reforçando a proposta de alcançar metas coletivas e não individuais.

O Quadro 4 mostra a quarta análise realizada, que foi uma comparação entre o desenvolvimento

de novos produtos com *Scrum* e o antigo processo de desenvolvimento de produtos.

Por fim, a equipe considera que a empresa está pronta para adotar o *Scrum* no desenvolvimento de todos os seus produtos. Segundo um colaborador:

Quando olhamos para trás, vemos claramente que progredimos continuamente durante as iterações. Hoje, já fazemos os artefatos e as cerimônias muito melhor que na primeira iteração. Por termos agora esta bagagem, acreditamos completamente que a empresa está preparada para implantar o *Scrum* em todos os outros projetos. (Colaborador da Equipe de Projetos da B2ML).

## 7 Conclusão

Pode-se afirmar que a contribuição principal da presente pesquisa é mostrar de forma científica o impacto da implantação do método ágil *Scrum* nos projetos de desenvolvimento de novos produtos de *software* de uma empresa de base tecnológica. O foco da análise foi o impacto do método sobre o ponto de vista dos benefícios que a literatura lhe atribuíra. A equipe participante da pesquisa sentiu diretamente quatro dos benefícios do *Scrum*: melhoria na comunicação e aumento da colaboração entre envolvidos; aumento da motivação da equipe de desenvolvimento; diminuição no tempo gasto para terminar o projeto (prazo); diminuição do risco do projeto (menor possibilidade de insucesso).

Outros dois benefícios presentes na literatura foram medidos e notados pelos dados quantitativos do projeto piloto e de outros projetos da empresa: diminuição dos custos de produção (mão de obra) e aumento de produtividade da equipe.

Entretanto, não foi possível investigar se houve aumento da qualidade do produto, da satisfação de clientes (pela diminuição das reclamações) e do retorno do investimento do projeto. Seria interessante que, posteriormente, após o produto ser de fato comercializado, se investigue sobre esses três fatores, sendo esta uma proposta para um futuro trabalho. Com isso, se completaria a análise dos benefícios mapeados na pesquisa bibliográfica. Além disso, uma outra proposta de futuro trabalho seria a mensuração quantitativa de certos itens como os custos de produção, o tempo gasto no projeto e o índice de reclamações. Para isso, seria necessário conhecer dados confiáveis de histórico de projetos da empresa e compará-los com dados após a implantação do *Scrum*.

Concluiu-se que o método *Scrum* foi condizente com a realidade da empresa, pois se mostrou um processo focado em resultados, na comunicação da equipe e interação com os clientes sem desrespeitar as restrições enfrentadas por uma pequena empresa. Isso corrobora a pesquisa realizada, mostrando que o *Scrum* pode ser empregado como um método adequado para a gestão de projetos de *software*.

**Quadro 4.** Quadro comparativo antes e após a implantação do *Scrum*.

Item	Antes	Depois
Gerência do projeto	Cada time tinha um gerente. Ele tinha como responsabilidades o controle do projeto, o contato com o cliente e assumia seus riscos.	O time se autogerencia. Todos os membros do time assumem os riscos do projeto e têm comprometimento com seu sucesso.
Contato com o cliente	Somente o gerente do projeto interagira com o cliente.	Criou-se a figura do dono do produto, o qual tem grande interação com o cliente e conhece suas regras de negócio.
Velocidade	Não havia medição formal da velocidade do projeto. O único controle eram os prazos.	A velocidade do projeto é medida e um gráfico de controle visual de toda a equipe é atualizado diariamente.
Planejamento do projeto	O projeto era inteiramente planejado em seu início em detalhes.	O projeto é rapidamente planejado no início e são realizados replanejamentos em cada <i>Sprint</i> .
Funcionalidades	As funcionalidades eram implementadas sem uma ordem específica.	Os itens de maior prioridade são implementados antes, de modo que o retorno do investimento ocorra mais rapidamente e funcionalidades opcionais são implementadas se o tempo permitir.
Lições Aprendidas	Não havia método formal para documentar as lições aprendidas	O time realiza reuniões de revisão para documentar as lições aprendidas.
Riscos	A gestão de riscos não era uma tarefa formal e ficava sempre a cargo do gerente do projeto.	Existe o <i>Backlog</i> de Impedimentos, que é uma boa ferramenta para todo o time conhecer os riscos e cobrar a mitigação de riscos conhecidos.
Entregas	São definidas no início do projeto. Geralmente é feita apenas uma entrega, já com o produto final.	As entregas são diversas durante todo o projeto, sempre com um produto com melhorias incrementais em relação ao anterior.

Durante a realização da pesquisa foi possível a documentação de algumas lições aprendidas com o projeto, que podem ser muito úteis para outras instituições que desejam implantar o método e para orientar outros pesquisadores que desejem replicar o presente trabalho.

Com esses resultados alcançados, percebeu-se que a organização se tornou mais competitiva, pois a bem-sucedida gestão de desenvolvimento de produtos é ponto crucial para o sucesso de uma empresa de base tecnológica.

O uso do método de Kniberg (2007) como base para a implantação, embora não seja um método com grande embasamento científico, foi uma boa opção porque propõe uma lista de verificação bastante objetiva e simples de ser aplicada. Além disso, é um método que nasceu na prática do autor no trabalho com empresas de base tecnológica.

Percebeu-se que os pontos críticos da implantação foram a falta de conhecimento em estimativas de *software* e a dificuldade do time no autogerenciamento. Portanto, pode-se dizer que o método utilizado nesta pesquisa-ação precisa ser aprimorado para ser utilizado em outras instituições. Esse aprimoramento pode ser conseguido com a realização de novas pesquisas sobre o tema.

Um dos fatores de sucesso na implantação foi o comprometimento do time. Os pesquisadores e

os membros do time concordaram que o grande empenho dos envolvidos foi fator fundamental para alcançar o objetivo.

Outro fator relevante a se destacar é a influência e a importância da alta direção em todo o processo de implantação do novo método. Sem este apoio, seria muito mais difícil o sucesso deste projeto, pois ela é responsável por várias ações importantes, principalmente na definição das estratégias e mudança da cultura organizacional. Além disso, é ela quem disponibiliza os recursos humanos e financeiros necessários ao programa.

Sugere-se a replicação desta pesquisa-ação em outras empresas da mesma natureza, que observem problemas na gestão de desenvolvimento de produtos, como mais uma proposta de trabalhos futuros. Com isso, o desdobramento dessa pesquisa em novos trabalhos irá contribuir para a consolidação e validação dos resultados aqui obtidos. Com esta evolução, pode-se, por fim, se criar um método específico de implantação do *Scrum* em pequenas empresas de base tecnológica.

Outra sugestão para trabalhos futuros seria analisar se a implantação de outro método ágil é mais viável e benéfica que o *Scrum*. Neste trabalho, não se pôde analisar vantagens e desvantagens de cada método ágil, embora isto já tenha sido feito de certa forma em alguns trabalhos, como em Abrahamsson et al. (2002).

Também é uma boa sugestão para trabalhos futuros uma análise mais aprofundada da questão da medição de velocidade a estimativas para trabalhos com *Scrum*. Este foi um dos pontos mais críticos do presente trabalho e isto pode ocorrer em outras empresas. É notável que este assunto possui um campo de pesquisa ainda muito grande a ser explorado.

## Agradecimentos

Os autores gostariam de agradecer o apoio dado para o desenvolvimento da presente pesquisa pela FAPEMIG na forma da bolsa de produtividade, do Programa Pesquisador Mineiro, do processo TEC-PPM-00043/08.

## Referências

- ABRAHAMSSON, P. et al. **Agile software development methods** – review and analysis. Espoo: VTT Publications, 2002.
- AMBLER, S. **Agile modeling**. New York: Wiley Computer Publishing, 2002.
- ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE – ABES. Disponível em: <<http://www.abes.org.br/temp12.aspx?id=305&sub=305>>. Acesso em: mar. 2008.
- BARTON, B.; CAMPBELL, E. Implementing a professional services organization using type C Scrum. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES – HICSS, 40., 2007, Maui. **Proceedings...** Maui, 2007. p. 275a.
- BERCZUK, S. Back to basics: the role of agile principles in success with an distributed scrum team. In: AGILE CONFERENCE, 2007, Washington. **Proceedings...** Washington, 2007. p. 382-388.
- BRASIL. Congresso Nacional. Lei complementar nº 123, de 14 de dezembro de 2006. Institui o Estatuto Nacional da Microempresa e da Empresa de Pequeno Porte. **Diário Oficial da República Federativa do Brasil**, Brasília, DF, 14 dez. 2006. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/Leis/LCP/Lcp123.htm](http://www.planalto.gov.br/ccivil_03/Leis/LCP/Lcp123.htm)>. Acesso em: ago. 2008.
- BRUEGGE, B.; SCHILLER, J. Word spotting in scrum meetings. In: INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATION – DEXA, 19. 2008, Turin. **Proceedings...** IEEE Computer Society, 2008. p. 125-129.
- CARVALHO, B. V.; MELLO, C. H. P. Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil Scrum. In: SIMPÓSIO DE ADMINISTRAÇÃO DA PRODUÇÃO, LOGÍSTICA E OPERAÇÕES INTERNACIONAIS – SIMPOI, 12., 2009, São Paulo. **Anais...** São Paulo, 2009.
- CARVALHO, M. M. et al. Fatores críticos de sucesso em empresas de base tecnológica. **Produto & Produção**, v. 4, n. especial, p. 47-59, 2000.
- CRISTAL, M.; WILDT, D.; PRIKLADNICKI, R. Usage of SCRUM: practices within a global company. **Global Software Engineering**, p. 222-226, 2008.
- DANTAS, V. F. **Uma metodologia para o desenvolvimento de aplicações Web num cenário global**. 2003. Dissertação (Mestrado)-Centro de Ciências e Tecnologia, Universidade Federal de Campina Grande, Campina Grande, 2003.
- DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: a systematic review. **Information and Software Technology**, v. 50, n. 9-10, p. 833-859, 2008. <http://dx.doi.org/10.1016/j.infsof.2008.01.006>
- EDWARDS, M. Overhauling a failed project using out of the box scrum. In: AGILE CONFERENCE, 2008, Toronto. **Proceedings...** Toronto, 2008. p. 413-416.
- FERNANDES, A. C.; CÔRTEZ, M. R.; OSHI, J. Innovation characteristics of small and medium sized technology-based firms in São Paulo, Brazil: a preliminary analysis. In: INTERNATIONAL CONFERENCE OF TECHNOLOGY POLICY AND INNOVATION – ICTPI, 4., 200, Curitiba. **Proceedings...** Curitiba, 2000.
- FOWLER, M. **Put your process on a diet**. Software Development, 2000. Disponível em: <<http://www.sdmagazine.com/articles/2000/0012/>>. Acesso em: jul. 2003.
- GONZALEZ, M. O. A.; TOLEDO, J. C. A integração do cliente no processo de desenvolvimento de produto: revisão bibliográfica sistemática e temas para pesquisa. **Produção**, v. 22, n. 1, p. 14-26, 2012.
- GREGÓRIO, M. et al. **Os sete pecados na aplicação de processos de software**. UNIBRATEC – União Brasileira dos institutos de tecnologia, 2007. Disponível em: <[http://www.unibrathec.com.br/revistacientifica/n2\\_artigos/n2\\_gregorio\\_mla.pdf](http://www.unibrathec.com.br/revistacientifica/n2_artigos/n2_gregorio_mla.pdf)>. Acesso em: jan. 2009.
- IT WEB. **Softwares movimentam US\$ 15 bi no Brasil em 2008**. 2009. Disponível em: <<http://www.itweb.com.br/noticias/index.asp?cod=57034>>. Acesso em: maio 2009.
- JOHNSON, J. **The Chaos Report**. West Yarmouth: The Standish Group International, 1995.
- JOHNSON, J. **The Chaos Report**. West Yarmouth: The Standish Group International, 2001.
- JØRGENSEN, M.; MOLØKKEN, K. How large are software cost overruns? Critical Comments on the Standish Group's CHAOS Reports. **Information and Software Technology**, v. 48, n. 4, p. 297-301, 2006.
- KNIBERG, H.; FARHANG, R. Bootstrapping Scrum and XP under crisis. In: AGILE CONFERENCE, 2008, Toronto. **Proceedings...** Toronto, 2008. p. 436-444.
- KNIBERG, H. **Scrum and XP from the trenches – How we do Scrum**. InfoQ, 2007. Disponível em <<http://www.crisp.se/henrik.kniberg/ScrumAndXpFromTheTrenches.pdf>>. Acesso em: mar. 2008.
- LEDWITH, A. Management of new product development in small electronics firms. **Journal of European Industrial Training**, v. 24, n. 2-4, p. 137-148, 2000. <http://dx.doi.org/10.1108/03090590010321106>
- MACULAN, A. M. Ambiente empreendedor e aprendizado das pequenas empresas de base tecnológica. In: LASTRES, H. M. M.; CASSIOLATO, J. E.; MACIEL, M. L. **Pequena empresa: cooperação e desenvolvimento local**. Rio de Janeiro: Relume Dumará, UFRJ, 2003. p. 311-327.
- MANIFESTO ÁGIL. 2001. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em: mar. 2009.



- MANN, C.; MAURER, F. A Case study on the impact of Scrum on overtime and customer satisfaction. In: AGILE DEVELOPMENT CONFERENCE, 2005. **Proceedings...** IEEE Computer Society, 2005. p. 70-79.
- MARÇAL, A. et al. Mapping CMMI project management process areas to SCRUM practices. In: SOFTWARE ENGINEERING WORKSHOP, 2007. **Proceedings...** 2007. p. 13-22.
- MUNDIM, A. P. F. et al. Aplicando o cenário de desenvolvimento de produtos em um caso prático de capacitação profissional. **Gestão & Produção**, v. 9, n. 1, p. 1-16, 2002.
- PAASIVAARA, M.; DURASIEWICZ, S.; LASSENIUS, C. Distributed agile development: using Scrum in a large project. **Global Software Engineering**, p. 87-95, 2008.
- RAYHAN, S.; HAQUE, N. Incremental adoption of Scrum for successful delivery of an IT project in a remote setup. In: AGILE CONFERENCE, 2008, Toronto. **Proceedings...** Toronto, 2008. p. 351-355.
- RISING, L.; JANOFF, N. S. The Scrum software development process for small teams. **IEEE Software**, v. 17, n. 4, p. 26-32, 2000. <http://dx.doi.org/10.1109/52.854065>
- SALO, O.; ABRAHAMSSON, P. Agile methods in European embedded software development organisations. **IET Software**, v. 2, n. 1, p. 58-64, 2008. <http://dx.doi.org/10.1049/iet-sen:20070038>
- SANDERS, D. Using Scrum to manage student projects. **Journal of Computing Sciences in Colleges**, v. 23, n. 1, p. 79-79, 2007.
- SCHWABER, K. **SCRUM Development process**. 1995. Disponível em: <<http://jeffsutherland.com/oops/la/schwapub.pdf>>. Acesso: jul. 2003.
- SCHWABER, K.; BEEDLE, M. **Agile software development with SCRUM**. Prentice Hall, 2002.
- SEBRAE; INSTITUTO DE PESQUISAS TECNOLÓGICAS – IPT. **MPES de base tecnológica: conceituação, formas de financiamento e análise de casos brasileiros**. São Paulo: Sebrae; IPT, 2001. Relatório de Pesquisa.
- SENGE. P. **The fifth discipline: the art and practice of the learning organization**. New York: Currency, 1990.
- SOUDER, W. E.; SHERMAN, D.; DAVIES-COOPER, R. Environmental uncertainty, organizations, and new product development effectiveness: a test of contingency theory. **Journal of Product Innovation Management**, v. 15, n. 6, p. 520-533, 1998. [http://dx.doi.org/10.1016/S0737-6782\(98\)00033-2](http://dx.doi.org/10.1016/S0737-6782(98)00033-2)
- SULAIMAN, T.; BARTON, B.; BLACKBURN, T. AgileEVM - Earned Value Management in Scrum Projects. In: AGILE CONFERENCE - AGILE'06, 2006. **Proceedings...** 2006. p. 7-16.
- SUTHERLAND, J. et al. Fully distributed Scrum - the secret sauce for hyperproductive offshored development teams. In: AGILE CONFERENCE, 2008, Toronto. **Proceedings...** Toronto, 2008. p. 339-344.
- SUTHERLAND, J. et al. Distributed Scrum. Agile project management with outsourced development system sciences. In: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES – HICSS, 2007. **Proceedings...** 2007. p. 274-285.
- TAKEUCHI, H.; NONAKA, I. The new new product development game. **Harvard Business Review**, p. 137-146, 1986.
- TAKEUCHI H.; NONAKA I. **Hitotsubashi on knowledge management**. Singapore: John Wiley & Sons, 2004.
- THIOLLENT, M. **Metodologia da pesquisa-ação**. 14. ed. São Paulo: Cortez, 2005.
- TOLEDO, J. C. et al. Fatores críticos de sucesso no gerenciamento de projetos de desenvolvimento de produto em empresas de base tecnológica de pequeno e médio porte. **Gestão & Produção**, v. 15, n. 1, 2008.
- TRAC OPEN SOURCE PROJECT. Disponível em: <<http://trac.edgewall.org/>>. Acesso em: maio 2009.
- VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de pontos de função: medição, estimativas e gerenciamento de projetos de software**. 7. ed. Rio de Janeiro: Editora Érica, 2007.