

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**Bernardo Vasconcelos de Carvalho**

**APLICAÇÃO DO MÉTODO ÁGIL *SCRUM* NO  
DESENVOLVIMENTO DE PRODUTOS DE  
SOFTWARE EM UMA PEQUENA EMPRESA DE  
BASE TECNOLÓGICA**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Produção como requisito parcial à obtenção do título de *Mestre em Ciências em Engenharia de Produção*

**Orientador:** Prof. Carlos Henrique Pereira Mello, Dr.

**Itajubá**

**2009**

CARVALHO, Bernardo Vasconcelos de  
Aplicação do método ágil *Scrum* na gestão de desenvolvimento de  
produtos de software por uma pequena empresa de base tecnológica  
/ Bernardo Vasconcelos de Carvalho – Itajubá: UNIFEI, 2009.

100 p.

Dissertação (mestrado) – Universidade Federal de Itajubá, 2009.  
Orientador: Carlos Henrique Pereira Mello

1. Gestão de Projetos 2. Scrum 3. Desenvolvimento de Produtos  
I. Carvalho, Bernardo Vasconcelos de. II. Universidade Federal de  
Itajubá III. Aplicação do método ágil Scrum na gestão de  
desenvolvimento de produtos de software por uma pequena empresa de  
base tecnológica

**UNIVERSIDADE FEDERAL DE ITAJUBÁ**

**Bernardo Vasconcelos de Carvalho**

**APLICAÇÃO DO MÉTODO ÁGIL *SCRUM* NA  
GESTÃO DE DESENVOLVIMENTO DE  
PRODUTOS DE SOFTWARE POR UMA  
PEQUENA EMPRESA DE BASE TECNOLÓGICA**

Dissertação submetida para aprovação por banca examinadora em 17 de setembro de 2009,  
conferindo ao autor o título de *Mestre em Ciências em Engenharia de Produção*

**Banca Examinadora:**

Prof. Dr. Carlos Henrique Pereira Mello (Orientador)

Prof. Dr. Carlos Eduardo Sanches da Silva

Prof. Dr. Paulo Henrique de Souza Bermejo

**Itajubá**

**2009**

# **DEDICATÓRIA**

Aos meus pais, Ronaldo e Raquel e às minhas irmãs, Tatiana e Isabela, sem os quais nenhuma das minhas conquistas seria possível.

À minha mulher, Priscila, que caminha ao meu lado e se sacrificou por esta conquista.



# **AGRADECIMENTOS**

Ao meu incansável orientador Professor Dr. Carlos Henrique Pereira Mello, que foi verdadeiramente um orientador.

Aos amigos professores Rita de Cássia Trindade Stano, Luiz Fernando Valadão Flores, Edmilson Marmo Moreira, João Bosco Schumann Cunha, Carlos Eduardo Sanches da Silva, João Batista Turrioni, Renato Silva Lima e Renato Aquino de Faria Nunes, que sempre me ajudaram prontamente quando precisei, desde meu primeiro dia como aluno da Unifei.

Aos amigos da B2ML Sistemas, Web-de-Resultado Lojas Virtuais, Mútuos Inteligência em Compras, INCIT (Incubadora de Empresas de Base Tecnológica de Itajubá) e CEGEIT (Centro Gerador de Empresas de Itajubá).

*“Bom mesmo é ir a luta com determinação,  
abraçar a vida e viver com paixão,  
perder com classe e vencer com ousadia.  
Pois o triunfo pertence a quem se atreve  
e a vida é muito para ser insignificante.”*

**Fernando Pessoa** (1888-1935), poeta português.

## 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. O objeto de pesquisa foi uma pequena empresa de Itajubá/MG, cujos principais produtos são sistemas de software. Estudos mostram que a indústria de produção de software é extremamente ineficiente e ineficaz. E as micro e pequenas empresas de base tecnológica (MPEBT) tem um desafio ainda maior devido aos seus recursos restritos. Além disso, os métodos tradicionais de desenvolvimento de produtos de softwares são pesados e custosos. Na década de 1980 surgiram as técnicas de desenvolvimento ágil de produtos, entre eles o *Scrum*. Este trabalho se principiou com uma revisão, análise, classificação e codificação da literatura sobre o método *Scrum* em toda a base de dados dos periódicos da CAPES. Tendo em vista a importância estratégica das MPEBT no desenvolvimento regional, seria muito importante que o *Scrum* fosse compatível com seus processos, para que as mesmas pudessem se tornar mais competitivas e usufruir de seus benefícios. Desta forma, o objetivo desta dissertação 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 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 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.

Palavras-chave: *Scrum*; Desenvolvimento Ágil de Produtos; Desenvolvimento de Produtos de Software; Gestão de Projetos de Software.

## **ABSTRACT**

This paper presents the result of an action research in a small technology-based company, which applied the Scrum agile method on a software product project. The research object was a small company, located at Itajubá - MG, whose main products are software systems. Studies has shown that the software industry production is extremely inefficient and ineffective. Small companies have even a greater challenge, considering their limited resources. Furthermore, traditional methods of software products development are heavy and expensive. In the 80's, new agile development techniques became available, including Scrum. Considering the strategic importance of small technology-based companies in regional development, it would be very important that Scrum is compatible with their processes, so that they could become more competitive and enjoy its benefits. Thus, the objective of the dissertation is monitor and assist the deployment of agile method Scrum in new software products development in a small technology-based company, understanding and measuring the impact of this deployment in the enterprise. Moreover, it also presented a review, analysis, classification and codification of the literature on the Scrum method across journals. It is concluded that the method has increased the motivation of the team, reduced the cost, time and risk of the project and increased the productivity of the team. With these achievements, the organization became more competitive because of the successful product development management processes, that is crucial to the success of a technology-based company.

Keywords: Scrum, Agile Product Development, Software Product Development, Software Project Management.

## **LISTA DE ABREVIATURAS E SIGLAS**

APL	Arranjo Produtivo Local.
B2ML	Empresa objeto de estudo deste trabalho.
CMMI	Capability Maturity Model Integration.
EBT	Empresa de Base Tecnológica.
INCIT	Incubadora de Empresas de Base Tecnológica de Itajubá.
MPE	Micro e pequena empresa.
MPEBT	Micro e pequenas empresas de base tecnológica.
MPS.br	Melhoria de Processos do Software Brasileiro.
P&D	Pesquisa e Desenvolvimento.
PDP	Processo de Desenvolvimento de Produtos.
UNIFEI	Universidade Federal de Itajubá.

## LISTA DE FIGURAS

Figura 1 - Visão geral da dinâmica de processo <i>Scrum</i> .....	26
Figura 2 - Itens da implantação .....	28
Figura 3 - Número de publicações por ano de publicação .....	34
Figura 4 -Distribuição das publicações por data de publicação .....	34
Figura 5 - Distribuição das publicações por base de dados.....	35
Figura 6 - Distribuição das publicações por tipo de estudo.....	36
Figura 7 - Distribuição das publicações por tipo de abordagem .....	36
Figura 8 - Distribuição das publicações por período de análise .....	37
Figura 9 - Distribuição das publicações por filiação dos autores.....	37
Figura 10 - Número de citações dos benefícios.....	38
Figura 11 – Sistemática da pesquisa-ação .....	42
Figura 12 - Teoria e prática na pesquisa-ação .....	42
Figura 13 - Cronograma da pesquisa-ação definido para a presente pesquisa .....	44
Figura 14 - Datas importantes do cronograma .....	44
Figura 15 - Antigo grupo de processos de planejamento padrão da B2ML (parte 1) .....	50
Figura 16 - Antigo grupo de processos de desenvolvimento padrão da B2ML (parte 2) .....	51
Figura 17 - Visualização das Tarefas do Projeto no b2mlProject .....	53
Figura 18 - Gráfico de Gantt exibido pelo b2mlProject .....	53
Figura 19- Diagrama de caso de uso de Cotação.....	58
Figura 20 - Diagrama de atividades (processo) do cadastro de produtos.....	60
Figura 21 - Itens implantados na primeira iteração .....	61
Figura 22 - <i>Sprints</i> do projeto no <i>Trac</i> .....	64
Figura 23 – Tela de edição do <i>Sprint</i> .....	67
Figura 24 - Aproximação da tela de edição do <i>Sprint</i> .....	67
Figura 25 - Tela que mostra o <i>Sprint</i> .....	68
Figura 26 - Aproximação da tela que mostra o <i>Sprint</i> .....	68
Figura 27 - Tela de detalhamento da funcionalidade .....	69
Figura 28 - Aproximação da tela de detalhamento da funcionalidade .....	69
Figura 29 - Itens implantados na segunda iteração.....	71
Figura 30 - Itens implantados na iteração final .....	75
Figura 31 - Gráfico <i>Burndown</i> da terceira iteração .....	77
Figura 32 - Produtividade por projeto .....	79



## LISTA DE TABELAS

Quadro 1 - Time: ações a serem tomadas	29
Quadro 2 - <i>Scrum Master</i> : ações a serem tomadas	29
Quadro 3 - Dono do Produto: ações a serem tomadas	29
Quadro 4 - <i>Backlog</i> do Produto: ações a serem tomadas	29
Quadro 5 - Estimativas: ações a serem tomadas	30
Quadro 6 - Reunião de Planejamento do <i>Sprint</i> : Ações a serem tomadas	30
Quadro 7 - <i>Sprint</i> : ações a serem tomadas	30
Quadro 8 - Reunião Diária: ações a serem tomadas	30
Quadro 9 - Reunião Retrospectiva: ações a serem tomadas	31
Quadro 10 - <i>Backlog</i> de Impedimentos: ações a serem tomadas	31
Quadro 11 - Velocidade: ações a serem tomadas	31
Quadro 12 - Gráfico <i>Burndown</i> : ações a serem tomadas	31
Quadro 13 - <i>Backlog</i> do <i>Sprint</i> : ações a serem tomadas	31
Tabela 14 - Benefícios e seus respectivos códigos	39
Tabela 15 - Time: ações tomadas na primeira iteração	61
Tabela 16 - <i>Scrum Master</i> : ações tomadas na primeira iteração	61
Tabela 17 - Dono do Produto: ações tomadas na primeira iteração	62
Tabela 18 - <i>Backlog</i> do Produto: ações tomadas na primeira iteração	62
Tabela 19 - Estimativas: ações tomadas na primeira iteração	63
Tabela 20 - Reunião de Planejamento do <i>Sprint</i> : ações tomadas na primeira iteração	63
Tabela 21- Duração dos <i>Sprints</i> do projeto.	64
Tabela 22 - <i>Sprint</i> : ações tomadas na primeira iteração	65
Tabela 23 - Reunião Diária: ações tomadas na primeira iteração	65
Tabela 24 - Reunião Retrospectiva: ações tomadas na primeira iteração	65
Tabela 25 – <i>Backlog</i> de Impedimentos: ações tomadas na primeira iteração	66
Tabela 26 - Velocidade: ações tomadas na primeira iteração	66
Tabela 27 - Gráfico <i>Burndown</i> : ações tomadas na primeira iteração	66
Tabela 28 - <i>Backlog</i> do <i>Sprint</i> : ações tomadas na primeira iteração	70
Tabela 29 - Time: ações tomadas na segunda iteração	71
Tabela 30 - Estimativas: ações tomadas na segunda iteração	72
Tabela 31 - Velocidade: ações tomadas na segunda iteração	72
Tabela 32 - Gráfico <i>Burndown</i> : ações tomadas na segunda iteração	72



Tabela 33 - <i>Sprint</i> : ações tomadas na segunda iteração	73
Tabela 34 - Reunião Diária: ações tomadas na segunda iteração	73
Tabela 35 - Reunião Retrospectiva: ações tomadas na segunda iteração	74
Tabela 36 – <i>Backlog</i> de impedimentos: ações tomadas na segunda iteração	74
Tabela 37 - <i>Backlog</i> do <i>Sprint</i> : ações tomadas na segunda iteração	74
Tabela 38 – <i>Backlog</i> de Impedimentos: ações tomadas na iteração final.	75
Tabela 39 - Estimativas: ações tomadas na iteração final	76
Tabela 40 - Velocidade: ações tomadas na iteração final	76
Tabela 41 - Gráfico <i>Burndown</i> : ações tomadas na iteração final.	76
Tabela 42 - Afirmações sobre os benefícios do <i>Scrum</i>	78
Tabela 43 – Média e desvio padrão das respostas	78
Tabela 44 - Análise das práticas gerenciais do <i>Scrum</i>	81
Tabela 45 – Quadro comparativo antes e após a implantação do <i>Scrum</i> .	83
Tabela 46 - Apresentação dos artigos codificados	96
Tabela 47 - Apresentação dos periódicos, anos de publicação, base de dados e métodos dos artigos codificados	97
Tabela 48 - Apresentação da abordagem, filiação do pesquisador e período de análise dos artigos codificados e os benefícios do <i>Scrum</i> neles citados	99

# SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>15</b>
1.1. Caracterização do problema .....	15
1.2. Objetivos.....	17
1.3. Justificativa.....	18
1.4. Estrutura do trabalho .....	21
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>22</b>
2.1 Métodos ágeis .....	22
2.2 O <i>Scrum</i> .....	24
2.3 Análise da literatura científica sobre Scrum.....	32
2.4 Desenvolvimento de software .....	39
<b>3. MÉTODO DA PESQUISA .....</b>	<b>41</b>
3.1. Classificação da pesquisa .....	41
3.2. Apresentação da unidade de análise .....	45
<b>4. DESCRIÇÃO DA PESQUISA-AÇÃO .....</b>	<b>48</b>
4.1. Fase exploratória: identificação da situação-problema .....	48
4.2. Planejamento da implantação .....	55
4.3. Fase de ação: primeira iteração .....	57
4.4. Fase de ação: segunda iteração.....	70
4.5. Fase de ação: iteração final.....	74
4.6. Fase de avaliação: análise dos resultados .....	77
<b>5. CONCLUSÃO.....</b>	<b>84</b>
5.1 Considerações finais .....	84
5.2 Limitações e recomendações para pesquisas futuras.....	86
<b>REFERÊNCIAS .....</b>	<b>87</b>
<b>ANEXO A – APRESENTAÇÃO DOS ARTIGOS CODIFICADOS .....</b>	<b>96</b>

# 1. Introdução

Neste capítulo será apresentada a caracterização do problema a ser resolvido com este trabalho. Também serão apresentados os objetivos, as justificativas e a estrutura do trabalho.

## 1.1. Caracterização do problema

No atual ambiente de desenvolvimento de software, os requisitos estão sujeitos a freqüentes alterações durante o ciclo de desenvolvimento do produto para atender as 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. Construir software nunca é uma atividade simples, já que ele é normalmente uma replicação virtual de processos, entidades ou sistemas complexos.

A entrada de novos produtos de software com preços mais acessíveis e o acirramento da competitividade levam empresas e consumidores a modificarem seus hábitos e valores. Em adição, a produção de software no Brasil ainda esbarra em barreiras como a pirataria e altas cargas tributárias. Segundo a Associação Brasileira das Empresas 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 alarmantes do Standish Group, que pesquisou mais de 30 mil projetos norte-americanos de desenvolvimento de produtos de software desde 1994, revelam que, embora tenha havido melhorias com o passar do tempo, ainda há um grande problema neste setor (JOHNSON, 1995). E existe uma boa razão para a preocupação: o gasto americano em projetos de aplicações de software chegou a US\$ 275 bilhões somente no ano de 1994, com mais de 200 mil projetos em andamento.

O estudo comprova 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 média era de somente 16%. Taxa de sucesso se refere aos projetos que resultam em produtos de software que são realmente utilizados pelo usuário final.

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).

O período em que a indústria de software melhorou foi exatamente o mesmo em que começaram a surgir técnicas de desenvolvimento ágil de produtos de software: meados da década de 90. 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 & Nonaka (2004) e Senge (1990).

Nesse contexto, destaca-se o *Scrum*, uma abordagem enxuta 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 & 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 *et. al.*, 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 do enorme desperdício de recursos de tempo e dinheiro nos projetos de software, existe a necessidade cada vez maior pela 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 sofre nenhum efeito de um sistema de informação. Por isso, é de suma importância que eles sejam eficazes. Quando isto não ocorre, todos são prejudicados com piores serviços, aumento de preços, diminuição de competição etc.

Esse conjunto de fatores que marcam a economia realçam também o interesse sobre o papel que as micro e pequenas empresas de base tecnológica podem ter no desenvolvimento de regiões e países. Tendo em vista a importância estratégica dessas empresas na geração de emprego, renda e desenvolvimento regional, seria muito importante que o *Scrum* fosse compatível para as mesmas. Isso porque a literatura mostra que a utilização deste método pode gerar benefícios como aumento da satisfação de clientes (diminuição de reclamações) (MANN & MAURER, 2005; SALO & ABRAHAMSSON, 2008), melhoria na comunicação e aumento da colaboração entre envolvidos nos projetos (MAURER et. al., 2007; BERZUK, 2007), aumento do retorno do investimento em projetos de novos produtos (SULAIMAN et. al., 2006; SUTHERLAND, 2005), aumento da motivação da equipe de desenvolvimento de produtos (KNIBERG & FARHANG, 2008; PAASIVAARA et. al., 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).

.

## **1.2. Objetivos**

Os objetivos desse trabalho são:

- (i) 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;

- (ii) 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.

### **1.3. Justificativa**

Segundo Toledo *et al.* (2008), as empresas de base tecnológica de pequeno e médio porte têm despertado o interesse de governos, da comunidade acadêmica e de agentes econômicos. Isto acontece em virtude do efeito positivo que essas empresas exercem no desenvolvimento regional. Isso por que elas tendem a se agrupar e a atuar na forma de arranjos produtivos locais. Em adição, as empresas de base tecnológica são associadas ao lançamento de produtos inovadores e desempenham um importante papel na substituição de importações (no caso de países em desenvolvimento), e na criação de redes locais de empresas de maior conteúdo tecnológico (FERNANDES *et al.*, 2000).

Segundo pesquisa do Serviço Brasileiro de Apoio à Micro e Pequena Empresa - SEBRAE/MG (2003), as MPEs brasileiras são responsáveis por mais de 61% dos empregos formais gerados no país. Para Lastres *et al.* (2005), a capacidade inovativa de uma região decorre das relações entre diversos atores econômicos, políticos e sociais. Neste sentido, o desenvolvimento dessa pesquisa a partir da interação da academia com uma empresa de base tecnológica residente em uma incubadora de empresas cumpre um papel importante no crescimento econômico do país. Por um lado, a universidade desempenha uma função importante na transferência e aplicação do conhecimento científico ao processo tecnológico. Do outro lado, uma pequena empresa investe em pesquisa aplicada para seu desenvolvimento, competitividade e inovação buscando atingir seu objetivo social e econômico de geração de emprego e renda.

Sob o ponto de vista da melhoria da gestão dos projetos de desenvolvimento de produtos, não são novidades trabalhos e entidades com este foco (COOPER & KLEINSCHMIDT, 1987; CLARK & WHEELWRIGHT, 1993; GRIFFIN, 1997; ROZENFELD *et al.*, 2006; PDMA, 2008; PMI, 2008). Isto possibilitou o desenvolvimento de diversas estratégias, metodologias e ferramentas que estão sendo continuamente empregadas nas empresas, visando melhorias em indicadores como custo, volume de vendas, *time-to-market* e qualidade.

Todavia, a maioria das pesquisas realizadas foca o desenvolvimento de produtos nas grandes empresas que, segundo Toledo *et al.* (2008), “normalmente realizam o ciclo completo de inovação tecnológica e adotam um modelo de referência para gestão integrada deste processo”. Porém, sabe-se que a realidade da maioria das pequenas empresas de base tecnológica é bastante diferente, devido à escassez de recursos e inexperiência da equipe de gestão com o desenvolvimento de projetos. E ao mesmo tempo, tais empresas destacam-se pela incorporação de inovações tecnológicas nos novos produtos, o que torna a gestão da tecnologia e do processo de desenvolvimento de produto críticos para a competitividade dessas empresas.

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 *et al.*, 1998). De acordo com Carvalho *et al.* (2000), 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 (no tópico 2.3 serão mostradas tais evidências). Isto faz com que considere-se relevante a contribuição científica deste trabalho.

Segundo pesquisas nacionais sobre esse tema (FERNANDES *et al.*, 2000; SEBRAE/IPT, 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.

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 12<sup>a</sup> posição no mercado mundial

de software e serviços relacionados, tendo movimentado aproximadamente 11,12 bilhões de dólares, equivalente 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 ABES (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.

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 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.

Para este trabalho, considera-se uma empresa de base tecnológica uma empresa que se adéqua à definição do decreto nº 43.442 de 17 de julho de 2003 do Governo de Minas Gerais, que legisla sobre o Programa de Apoio Financeiro ao Desenvolvimento de Médias, Pequenas e Microempresas de Base Tecnológica do Estado de Minas Gerais.

“Define-se como empresa de base tecnológica aquela que se utiliza de aplicação sistemática de conhecimentos científico e tecnológico em biotecnologia, ciência da computação, mecânica de precisão, microeletrônica, novos materiais, química fina e outras ciências e áreas afins, usados isoladamente ou em combinações entre si, para o desenvolvimento de novos produtos, processos ou serviços” (MINAS GERAIS, 2003).

Da mesma forma, neste trabalho considera-se uma micro ou pequena



empresa a entidade que se à definição vigente do governo federal, que estabelece que é microempresa a pessoa jurídica que auferir em cada ano-calendário receita bruta igual ou inferior a R\$ 240.000,00 (duzentos e quarenta mil reais) e que é empresa de pequeno porte a pessoa jurídica que auferir, em cada ano-calendário, receita bruta superior a R\$ 240.000,00 (duzentos e quarenta mil reais) e igual ou inferior a R\$ 2.400.000,00 (dois milhões e quatrocentos mil reais) (BRASIL, 2006).

Por fim, pode-se dizer que a presente pesquisa contribui para o crescimento da pesquisa em engenharia de produção da UNIFEI e do Brasil na área da gestão de desenvolvimento de produtos e gestão de projetos. E, ao mesmo tempo, esse trabalho contribui com o desenvolvimento econômico e tecnológico do país.

#### **1.4. Estrutura do trabalho**

Este trabalho está estruturado em cinco capítulos. O primeiro capítulo é essa introdução que visa contextualizar a pesquisa e apresentar seus objetivos e justificativas. O próximo capítulo apresenta as construções teóricas acerca das temáticas que subsidiam os capítulos posteriores. O Capítulo 3 apresenta a classificação e método científico adotado para esta pesquisa. O Capítulo 4 descreve as cinco etapas da pesquisa realizada no objeto de estudo e analisa os resultados alcançados. Por fim, o Capítulo 5 apresenta as conclusões do trabalho, aponta algumas limitações da análise desenvolvida e dá recomendações para continuidade do estudo.

## 2. Fundamentação teórica

Neste capítulo serão apresentadas as fundamentações teóricas deste trabalho, que incluem revisões bibliográficas sobre os métodos ágeis, o *Scrum* e o Desenvolvimento de software. Também será apresentada uma análise da literatura científica sobre *Scrum*.

### 2.1 Métodos ágeis

O desenvolvimento de produtos é uma atividade bastante complexa, principalmente para as pequenas empresas que possuem grandes limitações de recursos. Segundo Mundin *et al.* (2002), o desenvolvimento de produtos se relaciona com praticamente todas as demais funções de uma empresa. Isso porque, para desenvolver produtos são necessárias informações e habilidades de membros de todas as áreas funcionais, caracterizando-se como uma atividade, em princípio, multidisciplinar. Além disso, trata-se de uma atividade com uma característica *ad-hoc*, em que cada projeto de desenvolvimento pode apresentar características específicas e um histórico particular.

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

Geralmente, estes métodos dividem o desenvolvimento em diversas iterações de ciclos mais curtos (*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). 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.

Estes 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 & Nonaka (2004) e Senge (1990).

Os métodos tradicionais de desenvolvimento tem o foco na geração de documentação sobre o projeto e no cumprimento rígido de processos. Já os métodos ágeis concentram as atenções no produto final (MUNDIN *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 neste tópico. Uma recente revisão sistemática bibliográfica 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.

Na linha dos métodos ágeis, em 2001 um grupo de diversos profissionais ligados à área de software inconformados com os métodos pesados de desenvolvimento de software muito utilizadas na época, lançaram o Manifesto Ágil, que é uma declaração de princípios que fundamentam o desenvolvimento ágil de software.

***“Manifesto para o desenvolvimento ágil de software***

*Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar:*

- 1. Indivíduos e interação entre eles mais que processos e ferramentas*
- 2. Software em funcionamento mais que documentação abrangente*
- 3. Colaboração com o cliente mais que negociação de contratos*
- 4. Responder a mudanças mais que seguir um plano*

*(MANIFESTO ÁGIL, 2001)*

Eles consideravam que havia valor em processos, documentações, contratos e planos. Mas achavam que eram mais valiosos os indivíduos, o software em si, a colaboração e a capacidade de mudança da equipe. Inicialmente, o texto contou com dezessete signatários: Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn,

Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas.

Existem diversos métodos ágeis, como o *Scrum*: a Programação extrema, o *Feature Driven Development*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Pragmatic Programming* e *Test Driven Development*. Para o desenvolvimento deste trabalho, o método escolhido foi o *Scrum* (esta escolha é explicada no tópico 4.1.), que será apresentado em detalhes no próximo tópico.

## **2.2 O Scrum**

O método *Scrum* segue os princípios do Manifesto Ágil e tem como pai três de seus signatários: Mike Beedle, Ken Schwaber e Jeff Sutherland. Segundo Schwaber & 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* irão iniciar um lance. A primeira utilização deste termo surgiu em um estudo de Takeuchi & 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 seis características (SCHWABER, 1995):

- Flexibilidade dos resultados;
- Flexibilidade dos prazos;
- Times pequenos;
- Revisões frequentes;
- Colaboração;
- Orientação a objetos.

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 pela coleta dos requisitos, conforme aponta Schwaber & Beedle (2002). Nesta prática, através 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 devem ser 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 de suas 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 devem decidir 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 que devem ser 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.

Uma visão geral da dinâmica de funcionamento do processo *Scrum* pode ser observada na figura 1. 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).

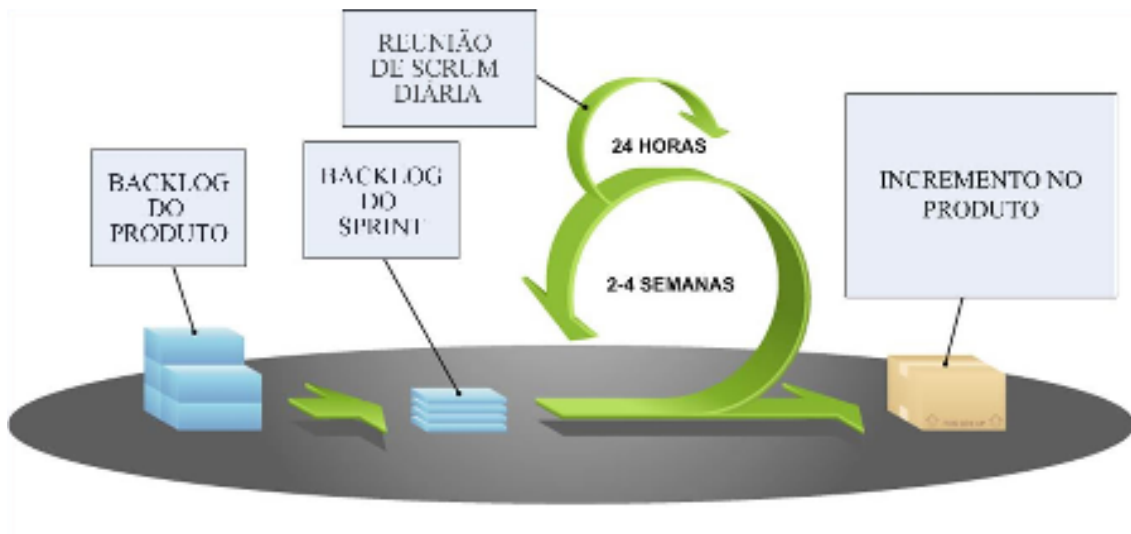


Figura 1 - Visão geral da dinâmica de processo *Scrum*

Fonte: Cohn (2008)

O *Scrum Master* deve trabalhar 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 (*Daily Scrum*) 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 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.

De acordo com a Figura 1, 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.

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 *et al.* (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.

O presente trabalho adaptou para a realidade da empresa o método de Kniberg (2007). Em sua obra, ele explica exatamente quais os itens de verificação devem ser observados para garantir a correta implantação do *Scrum* em uma instituição. A figura 2 apresenta tais itens necessários para a correta implantação do *Scrum*. O presente trabalho considera que um item está implementado após a equipe realizar cada uma das ações mostradas nos quadros de 1 a 13.

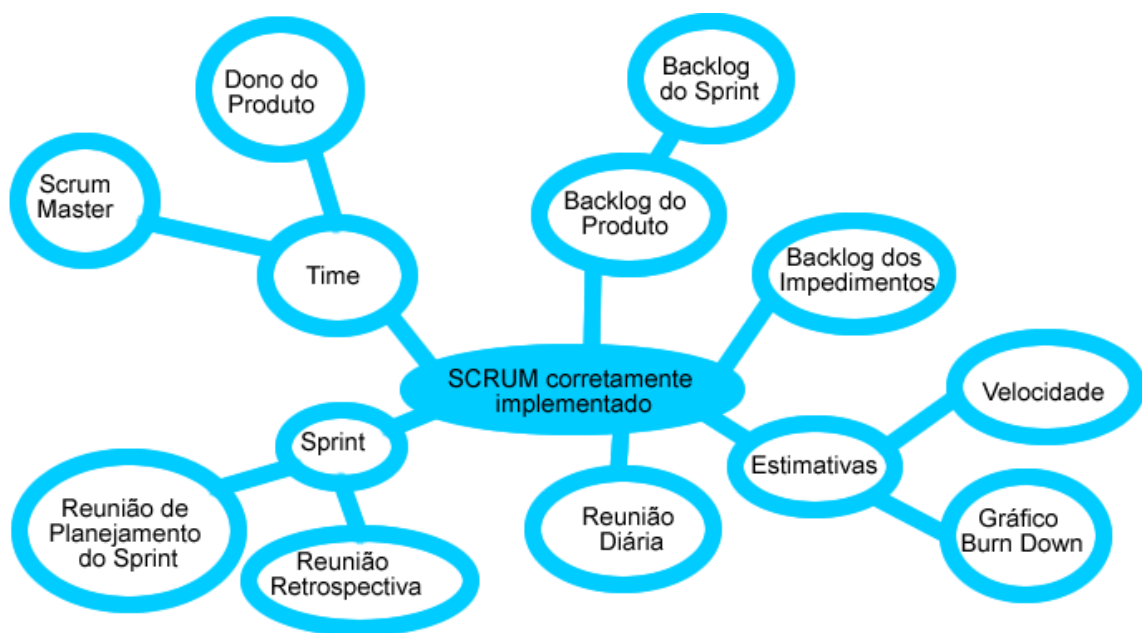


Figura 2 - Itens da implantação  
Fonte: Adaptado de Kniberg (2007)

Nos quadros a seguir serão apresentadas as ações a serem tomadas para se considerar cada item da implantação implementado.



Quadro 1 - Time: ações a serem tomadas

Time (membros do time)	Trabalham lado a lado. Este aspecto é importante para melhorar a comunicação e aumentar a sinergia no time.
	Colaboram entre si para implementar os requisitos do <i>backlog</i> do <i>Sprint</i> .
	Tem capacidade de desempenhar várias funções. Deve-se evitar uma super-especialização do time, de modo que um componente não saiba realizar a tarefa de outros.
	Admitem quando tem problemas e pedem ajuda ao <i>Scrum Master</i> quando isto acontece.
	Ajudam-se mutuamente. Eles precisam entender que as metas do projeto são todas coletivas: não há vitória ou derrota individual.
	Aceitam responsabilidades e se comprometem. O time deve se auto-gerenciar, o que torna esta característica fundamental.
	Não necessitam fazer hora extra sistematicamente, permitindo que tenham uma vida social agradável e menos estressante.

Quadro 2 - *Scrum Master*: ações a serem tomadas

<i>Scrum Master</i>	Todos os times tem um.
	Trabalha ao lado de seu time e está presente quando solicitado.
	Tem como prioridade máxima resolver os impedimentos da equipe (ele trabalha para resolver os itens do <i>backlog</i> dos impedimentos).

Quadro 3 - Dono do Produto: ações a serem tomadas

Dono do Produto ( <i>Product Owner</i> )	Todos os times tem um. Está sempre disponível para o time tirar dúvidas quanto às regras de negócio.
	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.
	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.
	Tem o poder de fazer o time trabalhar primeiramente para implementar as funcionalidades prioritárias.

Quadro 4 - *Backlog* do Produto: ações a serem tomadas

<i>Backlog</i> do Produto  ( <i>Product Backlog</i> )	O dono do produto tem total controle sobre o <i>Backlog</i> do Produto, podendo alterar prioridades e colocar novos itens.
	Ele está facilmente acessível e visível para todo o time em qualquer momento.
	Ele é atualizado após cada <i>Sprint</i> , durante a reunião de revisão do <i>Sprint</i> .
	Contém somente funcionalidades do produto e nada mais. O <i>Backlog</i> do Produto não contém nenhuma tarefa para o time.
	Cada funcionalidade do <i>Backlog</i> do Produto contém uma definição própria, de modo que não haverá dúvidas sobre considerá-la ou não implementada.

Quadro 5 - Estimativas: ações a serem tomadas

Estimativas	O dono do produto recebe da equipe estimativas de quantidade de trabalho de cada funcionalidade.
	O time tem liberdade de estimar e não sofre pressões externas para alterar estimativas.
	Todos os membros do time participam das estimativas.

Quadro 6 - Reunião de Planejamento do *Sprint*: Ações a serem tomadas

Reunião de Planejamento do <i>Sprint</i>	Todos os membros do time participam dela.
	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.
	Todos os membros do time devem concordar que o planejamento ficou realista. Caso contrário, a reunião deve continuar até se chegar a um consenso.
	Todos os membros do time se comprometem com o planejado.

Quadro 7 - *Sprint*: ações a serem tomadas

<i>Sprint</i>	O time entrega um produto (ou protótipo) funcionando no final de cada <i>Sprint</i> .
	O time segue rigorosamente as prioridades do <i>Backlog</i> do <i>Sprint</i> .
	O time age corretivamente quando está atrasado.
	O time alerta o <i>Scrum Master</i> e o dono do produto quando há problemas.
	O time sabe quando encontrar informações detalhadas que sejam úteis para implementar cada funcionalidade.
	Os problemas são discutidos e resolvidos no momento em que ocorrem.
	A duração de cada <i>Sprint</i> é sempre a mesma durante um projeto.
	O intervalo entre dois <i>Sprints</i> é de, no máximo, um dia.
	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.
Os funcionalidades que começam a ser implementadas num <i>Sprint</i> terminam no mesmo <i>Sprint</i> .	

Quadro 8 - Reunião Diária: ações a serem tomadas

Reunião Diária ( <i>Daily Scrum</i> )	Acontecem no mesmo lugar e horário todos os dias.
	Começam e terminam pontualmente.
	Todos os membros do time estão presentes.
	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?
	Nela não acontecem interrupções.
	O dono do produto a visita regularmente.
	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.

	Os membros da equipe cobram entre si a realização das tarefas.
--	--

Quadro 9 - Reunião Retrospectiva: ações a serem tomadas

Reunião Retrospectiva	Acontece no final de cada <i>Sprint</i> .
	Todos os membros da equipe e o dono do produto participam.
	Resulta em sugestões concretas de melhoria.
	Algumas sugestões apontadas pela reunião retrospectiva são implementadas de fato.

Quadro 10 - *Backlog* de Impedimentos: ações a serem tomadas

<i>Backlog</i> de Impedimentos  ( <i>Impediment Backlog</i> )	Todos os times tem um.
	Está visível para todos. Além disso, qualquer membro do time pode inserir novos impedimentos a qualquer momento com facilidade.
	Está sempre atualizado.
	Tem itens priorizados. O <i>Scrum Master</i> tem o objetivo de resolvê-los, primeiramente pelos mais importantes.
	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.

Quadro 11 - Velocidade: ações a serem tomadas

Velocidade	A velocidade de desenvolvimento é mensurada.
	A velocidade é utilizada para ações corretivas.
	A registro da velocidade é utilizada para melhorar a estimativa da equipe em planejamentos posteriores.

Quadro 12 - Gráfico *Burndown*: ações a serem tomadas

Gráfico <i>Burndown</i>	O time tem um gráfico <i>Burndown</i> .
	O gráfico <i>Burndown</i> é visível para toda a equipe.
	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.
	O time age corretivamente caso o gráfico <i>Burndown</i> mostre que o desenvolvimento está fora do planejado.

Quadro 13 - *Backlog* do *Sprint*: ações a serem tomadas

<i>Backlog</i> do <i>Sprint</i>	Todos os times tem um.
	É visível por toda a equipe.
	É atualizado diariamente (e facilmente) por toda a equipe.
	A estimativa de trabalho para as tarefas é atualizado diariamente.
	Apesar do <i>Backlog</i> do <i>Sprint</i> conter funcionalidades e tarefas, elas são facilmente distinguíveis.
	Nele está bem claro quais tarefas foram originadas por cada funcionalidade.

O próximo tópico irá discorrer sobre a revisão da literatura realizada sobre o método.

## **2.3 Análise da literatura científica sobre Scrum**

Foi realizada uma ampla revisão de literatura sobre *Scrum*, juntamente com uma análise, classificação e codificação dos artigos encontrados. O objetivo foi verificar quais as características dos estudos realizados, visando identificar possíveis tendências e carências na literatura científica sobre o tema (Carvalho & Mello, 2009).

É importante destacar que para identificar, localizar e adquirir as publicações de interesse sobre *Scrum*, foram consultadas todas as bases de dados e periódicos disponíveis no portal da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES): AAAS, ACM, ACS, AIP, Annual Reviews, Begen House, Bentham Science, BioOne, Balackwell, Cambridge University Pres, Cold Sprint Harbor Laboratory, Duke University Press, EBSCO, Emerald, Gale, Guilford Press, HighWire Press, IEEE, Informas, IOP, JSTOR, Karger, Maney Publishing, Nature, OECD, OVID, Oxford University Press, Red CLACSO, ProQuest, Sage, SciELO, Science Direct, Slack Inc., Springer, Thieme, Wilson e World Scientific.

Segundo Carnevalli e Cauchick Miguel (2007), é vital em uma revisão de literatura desta natureza utilizar as bases de dados disponíveis nos periódicos da CAPES devido a sua grande abrangência e facilidade de acesso para a maioria dos pesquisadores no Brasil.

A coleta de dados foi realizada durante os dias 15, 16 e 17 de outubro de 2008. Na consulta dos periódicos, buscou-se, como palavra-chave, o termo *Scrum*. Realizou-se a busca pelo título do trabalho e no campo *abstract*. Os artigos não foram filtrados pela sua data de publicação. Foram coletados os artigos produzidos em qualquer data.

Inicialmente, foram identificados 48 trabalhos. Entretanto, foram desconsiderados 8 artigos. Isto porque sete deles tratavam de temas esportivos relacionados ao *Rugby* e um artigo da área médica tratava da substância denominada “Scrum OestradioI-17-β”. Sendo assim, o universo de investigação deste trabalho contou com 40 artigos.

Para a análise dos dados, foram considerados os artigos em periódicos, em congressos e simpósios internacionais. Foram desconsideradas dissertações e teses, pelo fato da base de dados dos periódicos da CAPES divulgar um número limitado de teses e dissertações.

Para realizar os fichamentos dos artigos, utilizou-se uma adaptação do método utilizado por Carnevalli e Cauchick Miguel (2008). Os artigos foram catalogados e classificados em dois grupos principais: “pesquisa conceitual” e “pesquisa empírica”.

Os trabalhos classificados como “pesquisa conceitual” receberam ainda uma subclassificação da seguinte maneira: “teórico-conceitual”, “revisão de literatura”, “simulação” e “modelagem teórica”. Os trabalhos classificados como “pesquisa empírica” receberam uma subclassificação da seguinte maneira: “*survey*”, “estudo de caso”, “pesquisa-ação” e “pesquisa experimental”.

Também foram classificados os artigos conforme seu ano de publicação, sua base de dados de origem e o período de análise (que podem ser classificados em “Atual”, “Longitudinal” e “Retrospectivo”). A filiação dos autores foi classificada em “Universidade”, “Centro de Pesquisa” e “Empresa”.

Também se buscou conhecer quais os benefícios do *Scrum* são citados na literatura. Identificou-se um conjunto de nove benefícios apresentados e estes benefícios foram mapeados nesses artigos. No Anexo A é apresentado um detalhamento de todos os 40 artigos identificados.

Embora o método Scrum seja muito popular na Internet e nas empresas, não é uma tarefa simples encontrar artigos científicos sobre o tema. Entretanto, esta pesquisa mostra que este panorama tende a mudar. É notável o grande aumento das publicações sobre Scrum ao longo dos anos (figura 3). Para se ter uma idéia, se esta pesquisa tivesse sido realizada no começo de 2007, seriam encontrados apenas 11 artigos na literatura.

Este aumento pode também ser notado na figura 4, que mostra que 73% da literatura sobre Scrum foi publicada nos últimos dois anos (2007 e 2008).

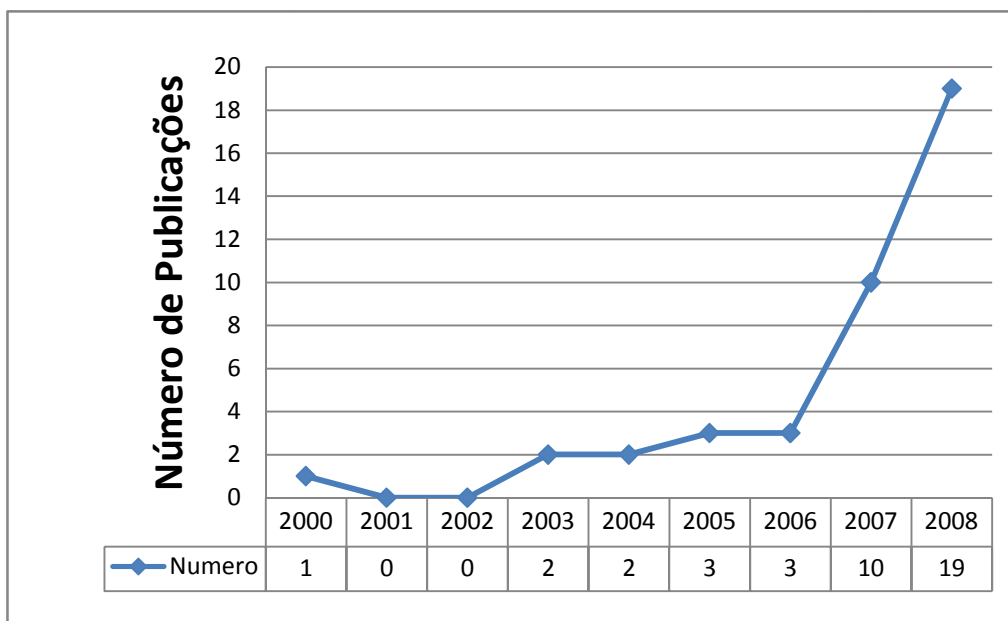


Figura 3 - Número de publicações por ano de publicação

O método *Scrum* foi apresentado à comunidade por Schwaber em 1995 no artigo “*Scrum Development Process*”. Entretanto, este não era um artigo com rigor científico e nem foi publicado por periódicos. Por isso, pode-se dizer que o artigo científico mais antigo sobre o tema foi de Rising e Janoff (2000). Trata-se de um artigo histórico, que na prática, apresentou o *Scrum* ao meio acadêmico. Somente três anos depois, novos artigos sobre *Scrum* passaram a ser publicados.

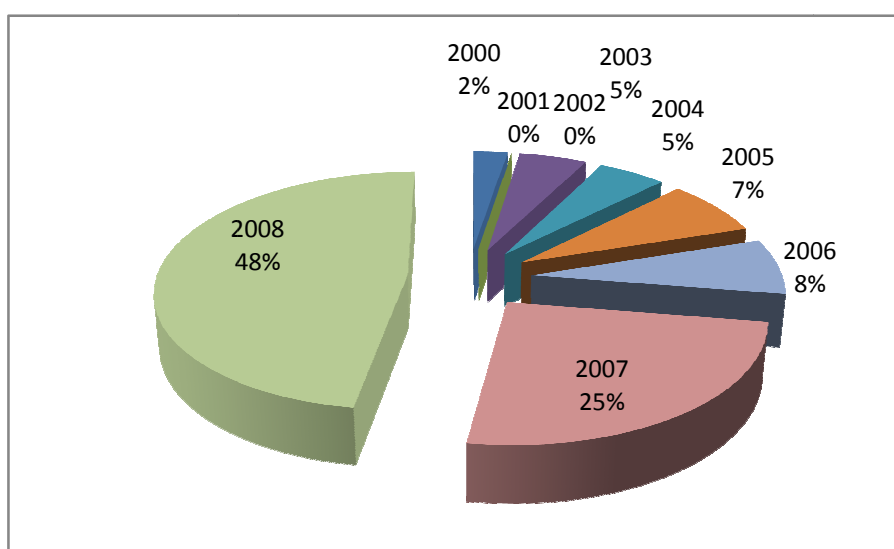


Figura 4 - Distribuição das publicações por data de publicação

Outro aspecto percebido foi a enorme concentração das publicações sobre *Scrum* na base de dados IEEE e ACM. Juntas, estas bases de dados concentram 94% das publicações. A figura 5 mostra ainda que as bases AIP e Science Direct tem uma pequena participação. O tema *Scrum* não faz parte da maioria das bases de dados: AAAS, ACS, Annual Reviews, Begen House, Bentham Science, BioOne, Balackwell, Cambridge University Pres, Cold *Sprint* Harbor Laboratory, Duke University Press, EBSCO, Emerald, Gale, Guilford Press, HighWire Press, Informs, IOP, JSTOR, Karger, Maney Publishing, Nature, OECD, OVID, Oxford University Press, Red CLACSO, ProQuest, Sage, SciELO, Slack Inc., Springer, Thieme, Wilson e World Scientific.

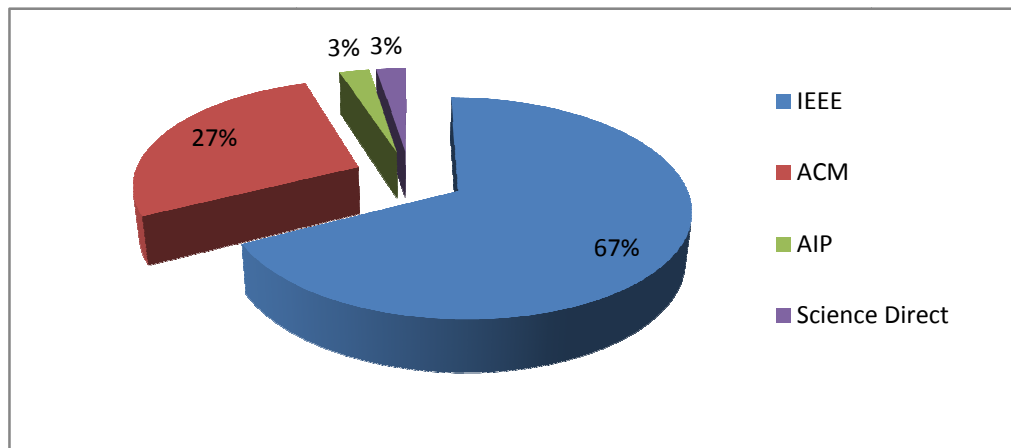


Figura 5 - Distribuição das publicações por base de dados

A figura 6 mostra o resultado da classificação dos artigos quanto ao tipo de estudo. Percebemos que os tipos mais comuns são o estudo de caso e o trabalho puramente teórico-conceitual. O pequeno número de revisões de literatura talvez seja causado pela escassez de artigos sobre o tema. O pequeno número de *surveys* e pesquisas-ações talvez reflitam a atual imaturidade das pesquisas sobre o tema, que ainda é muito novo. Além disso, é digno de nota que não foram encontrados trabalhos de simulação ou de modelagem teórica.

Ainda seguindo o conceito de classificação pelo tipo de trabalho, a figura 7 mostra a enorme concentração de pesquisas qualitativas. Foram encontrados apenas três artigos com abordagem quantitativa, como em Salo e Abrahamsson (2008) e Sulaiman

et al. (2006).

Como era de se esperar, tendo em vista a novidade do tema, não foi encontrado nenhum artigo de análise retrospectiva. Como vê-se na figura 8, a quase totalidade dos artigos faz análises atuais, com a exceção do artigo de Mann e Maurer (2005), que faz uma análise longitudinal (um estudo de caso de dois anos) para medir o impacto do *Scrum* na satisfação dos clientes de uma empresa.

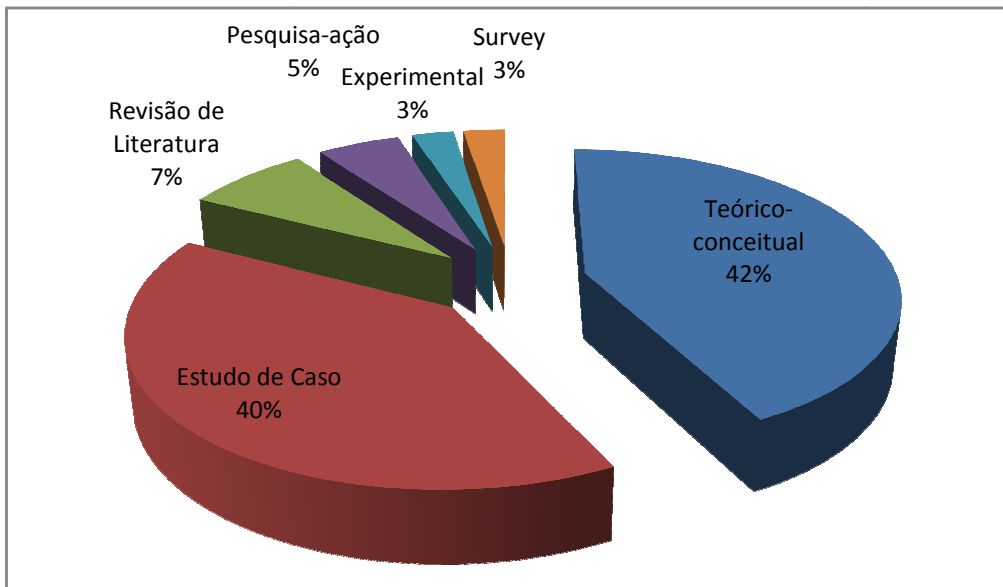


Figura 6 - Distribuição das publicações por tipo de estudo

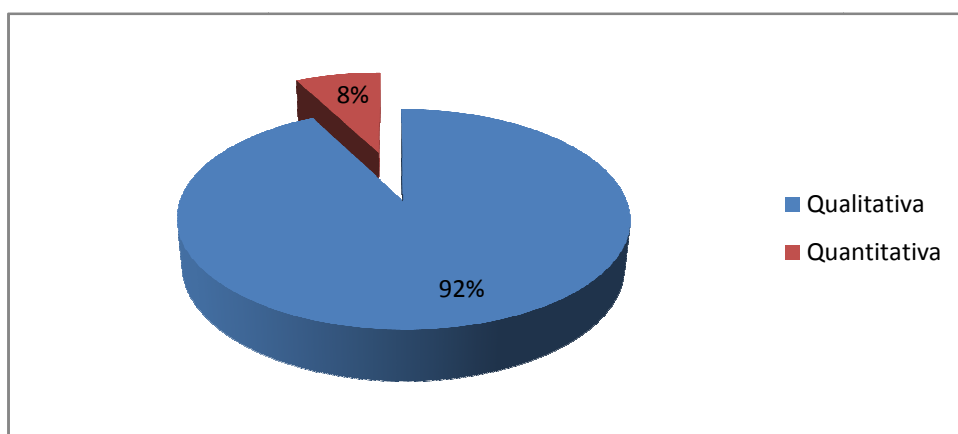


Figura 7 - Distribuição das publicações por tipo de abordagem



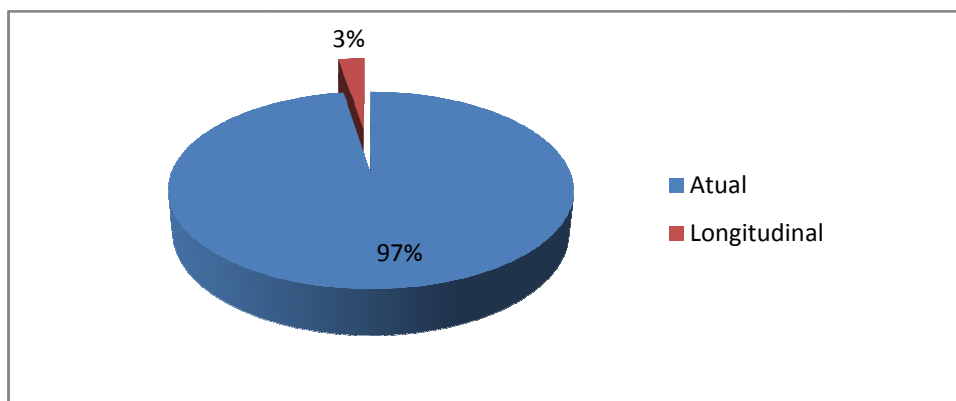


Figura 8 - Distribuição das publicações por período de análise

Uma informação interessante pode ser vista na figura 9, que mostra a filiação dos autores. Diferentemente de vários outros temas, a maioria dos pesquisadores sobre *Scrum* vem da indústria (principalmente de software) e não da academia.

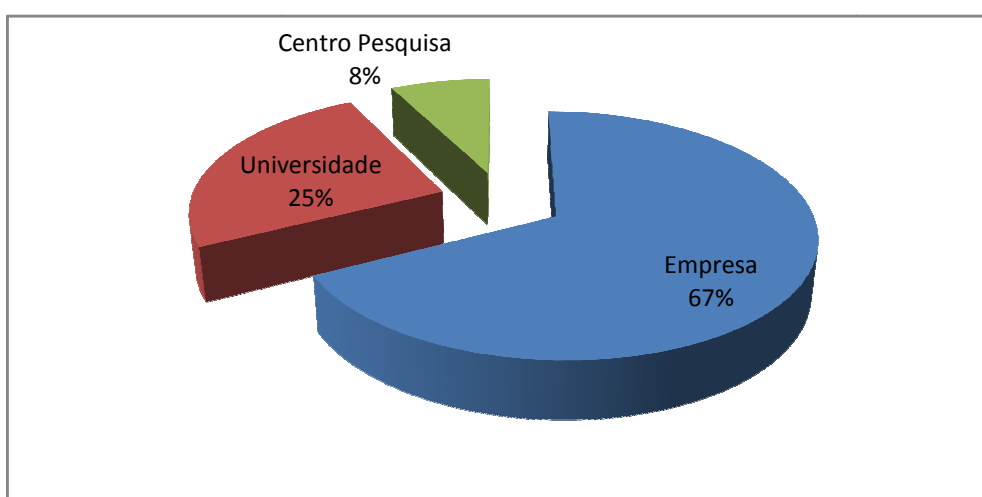


Figura 9 - Distribuição das publicações por filiação dos autores

Por fim, realizou-se um mapeamento dos benefícios do uso do *Scrum* mais citados nos artigos (figura 10 e tabela 14). O levantamento mostra que o benefício mais citado é a “melhoria na comunicação e aumento da colaboração entre envolvidos”. Este fato não é surpreendente, já que o *Scrum* é orientado às pessoas e um de seus pilares é realmente a alta colaboração entre membros da equipe. O que causa certa surpresa é o fato do segundo benefício mais citado ser a “melhoria da qualidade do produto produzido”. Inicialmente, o *Scrum* não foi criado com o foco na qualidade. Mas,

possivelmente, suas características acabaram impactando fortemente no aumento da qualidade.

De acordo com os dados levantados, podemos concluir que a literatura sobre *Scrum* é escassa, mas está em franca expansão. Se as tendências continuarem, em poucos anos serão comuns publicações sobre este tema. Notou-se também que existe uma grande concentração das publicações em poucos periódicos e bases de dados.

Ficou claro também que a literatura tem grande carência de trabalhos longitudinais e quantitativos sobre este tema. Sendo assim, trabalhos com estas características certamente tendem a ser muito bem aceitos na comunidade científica.

O fato da melhoria da qualidade ser um benefício tão citado nos artigos leva a ser levantada a hipótese de que o *Scrum* tem forte impacto na melhoria da qualidade do produto. Estudos mais apurados são necessários para testar esta hipótese, o que demandaria um interessante trabalho de pesquisa.

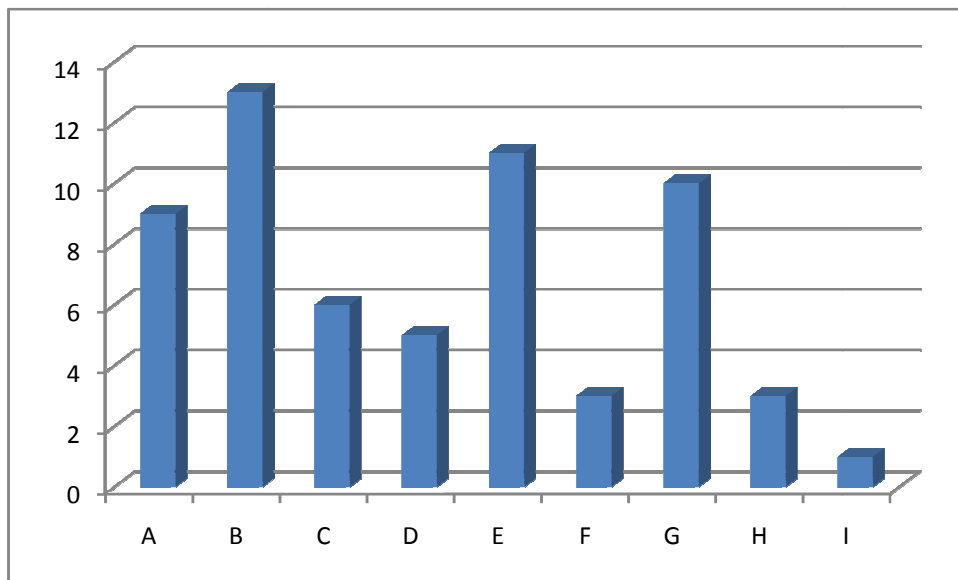


Figura 10 - Número de citações dos benefícios

Percebe-se ainda que o *Scrum* ainda é um tema predominantemente empresarial e pouco acadêmico. Isto leva a crer que existe uma grande lacuna científica a ser preenchida por pesquisadores da área com a produção deste conhecimento.

Tabela 14 - Benefícios e seus respectivos códigos

<b>Código</b>	<b>Benefício</b>	<b>Número de Citações</b>
A	Aumento da satisfação de clientes (diminuição de reclamações).	9
B	Melhoria na comunicação e aumento da colaboração entre envolvidos.	13
C	Aumento do retorno do investimento do projeto.	6
D	Aumento da motivação da equipe de desenvolvimento.	5
E	Melhoria da qualidade do produto produzido.	11
F	Diminuição dos custos de produção (mão-de-obra).	3
G	Aumento de produtividade da equipe.	10
H	Diminuição no tempo gasto para terminar o projeto (prazo).	3
I	Diminuição do risco do projeto (menor possibilidade de insucesso).	1

No próximo tópico, teremos uma rápida revisão bibliográfica sobre Desenvolvimento de Software.

## 2.4 Desenvolvimento de software

Para o completo entendimento deste trabalho, também é importante conhecer algumas peculiaridades do desenvolvimento de software que, segundo Pressman (2005), é o ato de elaborar e implementar um sistema computacional, isto é, transformar a necessidade de um cliente ou de um mercado em um produto de software.

O desenvolvimento de software utiliza os conceitos e técnicas de uma área do conhecimento chamada “engenharia de software”. A engenharia de software é uma área do conhecimento da computação voltada para a especificação, desenvolvimento e manutenção de sistemas de software aplicando tecnologias e práticas de gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade (PRESSMAN, 2005).

A engenharia de software chama de engenharia de requisitos “o ato de conhecer, levantar e especificar as funcionalidades necessárias de um sistema” (PRESSMAN, 2005). Esses requisitos geralmente são conhecidos através de entrevistas

com clientes e comparação com outros softwares do mercado.

Após o levantamento e análise de requisitos, um sistema de software é modelado. Segundo Pressman (2005), a modelagem de software é a atividade de construir modelos que expliquem as características ou o comportamento de um software.

Geralmente, essas duas atividades (requisitos e modelagem) são realizadas utilizando-se a Linguagem de Modelagem Unificada (*Unified Modelling Language - UML*), que é uma linguagem de diagramação ou notação para especificar, visualizar e documentar modelos de sistemas de software orientados a objeto. A UML não é um método de desenvolvimento, mas ela auxilia a visualizar o desenho do sistema e é um padrão da indústria para descrever graficamente softwares (CARDOSO, 2003).

A atividade de modelagem pode ser auxiliada por diagramas de atividades de cada funcionalidade. Este diagrama da UML descreve a seqüência de atividades de uma parte do sistema e tem o objetivo de mostrar o fluxo de atividades em um único processo.

Os requisitos funcionais podem ser descritos em diagramas de casos de uso (um dos diagramas da UML), identificando as principais funções do sistema e suas interações, ou seja, quem irá interagir com o mesmo, para facilitar a visão dos programadores. Um caso de uso representa uma unidade discreta da interação entre um usuário (humano ou máquina) e o sistema (HENSGEN *et al.*, 2003).

Estas são as ferramentas básicas de um projeto de desenvolvimento de produtos de software. Com elas, espera-se ser possível gerenciar um projeto deste tipo de forma mais eficiente e eficaz.

### **3. Método da pesquisa**

Neste capítulo será apresentada o método da presente pesquisa, assim como sua classificação metodológica. Também será apresentada a unidade de análise da pesquisa.

#### **3.1. Classificação da pesquisa**

Pode-se classificar o trabalho em questão 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 podem ser mensurados 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 (THIOLENT, 2005).

No presente trabalho os pesquisadores estavam diretamente envolvidos na aplicação do *Scrum* para o desenvolvimento de softwares e tiveram participações e influências sobre os processos. Este envolvimento permitiu a geração de dados nos processos organizacionais relacionados com o projeto, o que segundo Coughlan e Coughlan (2002), caracteriza uma pesquisa-ação (figura 11).

A pesquisa-ação é crítica e deve pressupor a exposição entre valores pessoais e práticos. Além disso, a pesquisa-ação não pretende apenas compreender ou descrever o mundo da prática, mas transformá-lo (FRANCO, 2005). O equilíbrio entre a teoria e a prática forma um círculo de aprendizagem constante, conforme mostrado na figura 12.

Os dados não foram gerados somente por meio da participação e observação de problemas resolvidos e de decisões tomadas, mas também, de intervenções que

foram realizadas para a continuidade do projeto. Algumas dessas observações e intervenções foram feitas de maneira informal. Outras foram realizadas de maneira formal, através de reuniões, documentação e entrevistas.

Segundo Coughlan e Coughlan (2002), a pesquisa-ação é apropriada quando a questão de pesquisa relaciona-se em descrever o desdobramento de uma série de ações ao longo do tempo em um dado grupo, comunidade ou organização; para explicar como e porque a ação de um membro de um grupo pode mudar ou melhorar o trabalho de alguns aspectos do sistema; e para entender o processo de mudança ou de melhoria para aprender com ele. Desta forma, percebe-se que a pesquisa-ação é o melhor método para cumprir os objetivos do presente trabalho.

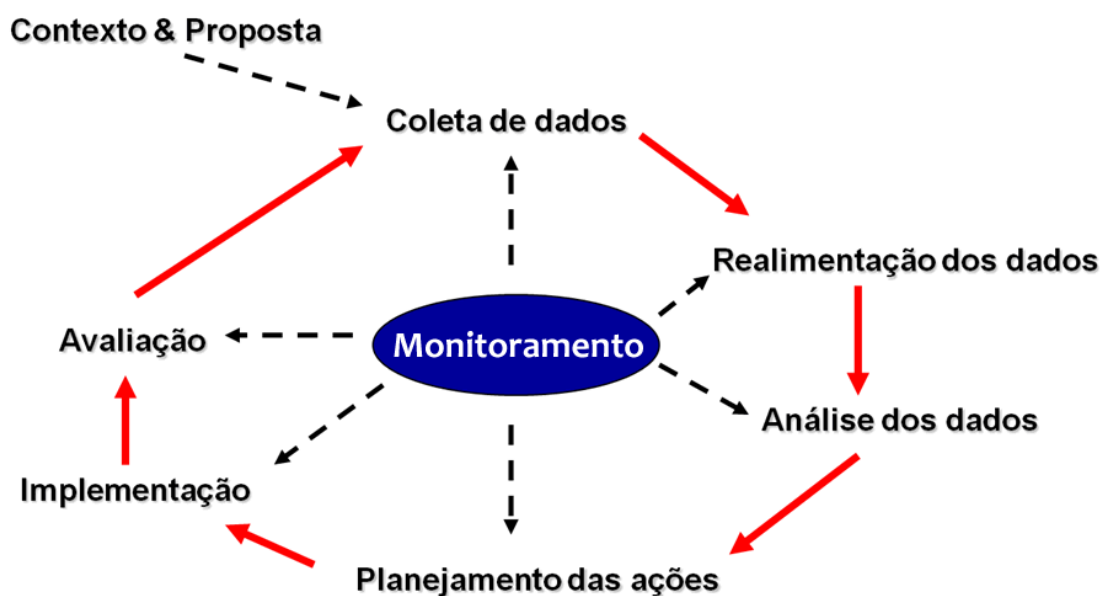


Figura 11 – Sistemática da pesquisa-ação

Fonte: Coughlan & Coughlan (2002)

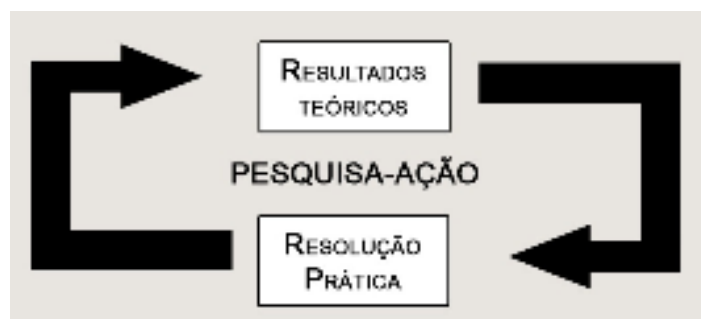


Figura 12 - Teoria e prática na pesquisa-ação

Fonte: Esteves (2006)

As figuras 13 e 14 mostram o cronograma simplificado de como a pesquisa foi realizada. As atividades desenvolvidas em cada fase são detalhadas no capítulo 4.

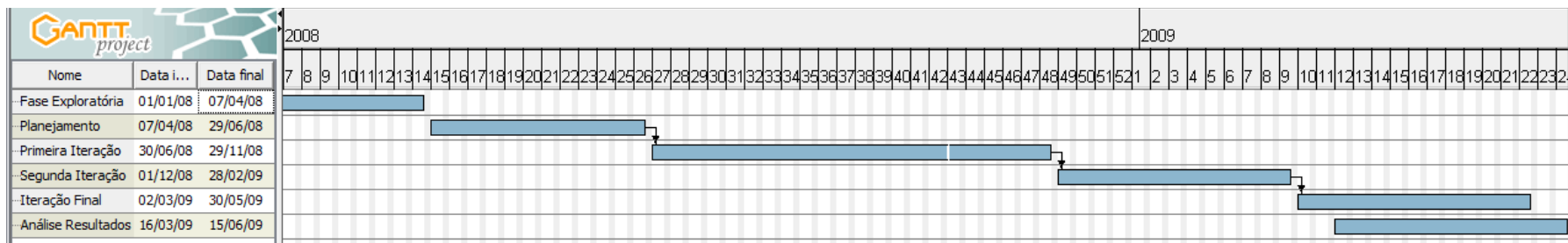


Figura 13 - Cronograma da pesquisa-ação definido para a presente pesquisa

Nome	Data i...	Data final
Fase Exploratória	01/01/08	07/04/08
Planejamento	07/04/08	29/06/08
Primeira Iteração	30/06/08	29/11/08
Segunda Iteração	01/12/08	28/02/09
Iteração Final	02/03/09	30/05/09
Análise Resultados	16/03/09	15/06/09

Figura 14 - Datas importantes do cronograma



## 3.2. Apresentação da unidade de análise

A unidade de análise selecionada para este trabalho é 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. A empresa desenvolve, implanta e mantém soluções completas para *Desktop*, Internet e *Intranet*, desde *websites* até sistemas de informação completos. Seus produtos são desenvolvidos para funcionar em diversas plataformas, como Windows, Linux, Unix, BSDs, MacOS etc.

Fundada em 2005 por então estudantes de computação da Universidade Federal de Itajubá (UNIFEI), a empresa reside atualmente na Incubadora da Empresas de Base Tecnológica de Itajubá (INCIT), no interior de Minas Gerais, e tem uma forte parceria com a UNIFEI. Outra parceira recorrente em projetos é a Fundação de Amparo a Pesquisa de Minas Gerais (FAPEMIG). Ela conta com laboratório próprio de desenvolvimento, plenamente equipado, onde realiza P&D voltado aos mais diversos tipos de sistemas de informação. Além disso, a parceria com a UNIFEI e com a Faculdade de Ciências Aplicadas do Sul de Minas (FACESM) possibilita a utilização de laboratórios especializados e grandes bibliotecas técnicas, além do auxílio e consultoria de professores doutores em diversos projetos já desenvolvidos.

A empresa conta com uma equipe multidisciplinar. Este quadro é composto de pessoas graduadas e pós-graduadas em áreas da Ciência e Engenharia da Computação, Engenharia de Produção e Administração.

A B2ML Sistemas é uma empresa focada na pesquisa e desenvolvimento. Cerca de 60% do seu faturamento são destinados para P&D. Essas constantes atividades fizeram com que essa empresa, mesmo tão nova, tenha lançado diversos produtos e serviços em tecnologia da informação. Todos os seus produtos foram desenvolvidos internamente e apresentam alto grau de inovação.

A qualidade de seu software produzido é reconhecida nacionalmente por entidades como a conceituada revista INFO-Exame, da Editora Abril, que elegeu um de seus aplicativos como o melhor do Brasil em sua área. Também a revista “Pequenas Empresas, Grandes Negócios”, da editora Globo, teceu vários elogios a um sistema produzido pela B2ML Sistemas.

Em 2006, a B2ML foi vencedora do "Prêmio Empresa do Ano", promovido

pelas incubadoras itajubenses: Incubadora de Empresas de Base Tecnológica de Itajubá (INCIT) e Centro Gerador de Empresas de Itajubá (CEGEIT). Em 2007, a B2ML foi a vencedora do programa “Empreender é Show” realizado pela Associação Nacional de Promoção de Empreendimentos Inovadores (ANPROTEC), além de ganhar pela segunda vez o "Prêmio Empresa do Ano".

Seus produtos atuais são:

- b2mlportal: Sistema de Gerenciamento de Conteúdos (SGC) para Internet;
- b2mlvarejo: ferramenta de gestão empresarial destinada ao mercado varejista;
- Empreenda!: ferramenta que permite ao empreendedor a avaliação econômica e financeira de um novo empreendimento;
- e-mercado: software para realização de compras conjuntas para Arranjos Produtivos Locais e associações de empresas;
- Web de Resultado: software associado ao serviço de gestão, logística e marketing em comércio eletrônico e lojas virtuais.

Além disso, a empresa ainda oferece os seguintes serviços:

- Fábrica de softwares: realiza desenvolvimento de software sob medida para seus clientes;
- *Outsourcing*: oferece serviços na administração, suporte e manutenção de sistemas corporativos.

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 institui o Estatuto Nacional da Microempresa e da Empresa de Pequeno Porte. Conhecida como Estatuto da MPE (Micro e Pequena Empresa), esta lei também define que as micro-empresas tem faturamento anual menor ou igual a R\$ 240 mil e as médias empresas tem faturamento anual acima de R\$ 2,4 milhões.

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. Acredita-se que o presente estudo possa contribuir para um melhor entendimento da realidade nacional tipicamente formada por micro e pequenas empresas.

Outro motivo é a facilidade de acesso à dados e ação que o pesquisador tem sobre a organização já que o mesmo 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. E, a partir dos resultados encontrados, espera-se que seja possível replicar os esforços em outras organizações de porte e características semelhantes.

## **4. Descrição da pesquisa-ação**

Neste capítulo será feita a descrição da pesquisa-ação propriamente dita. Serão apresentadas suas fases detalhadamente e será realizada uma análise dos resultados.

### **4.1. Fase exploratória: identificação da situação-problema**

A fase inicial da pesquisa diz respeito ao diagnóstico da situação e das necessidades da unidade de análise (através de entrevistas abertas, para detectar os principais problemas existentes). Também são apresentados a seguir os atores e a formação de equipes envolvendo pesquisadores e membros da empresa analisada. Nesta fase, foi divulgada a proposta desta pesquisa à empresa e foi obtido o comprometimento dos participantes e interessados.

Foram realizadas entrevistas semi-estruturadas 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 velho paradigma do desenvolvimento em cascata. Esta tentativa se mostrou um completo fracasso, já que os processos desenhados nunca saíram do papel. Nas figuras 15 e 16 vê-se os antigos grupos de processos padrões de planejamento e desenvolvimento.

Esses processos se mostraram muito pesados e tornaram-se inviáveis as tentativas de sua implantação. Na época, a equipe era pequena (cinco desenvolvedores) e não havia a menor possibilidade daquele time seguir todo o protocolo e gerar todos os artefatos e documentações necessárias. 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”.

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”.

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.

Apesar de haver um ambiente aberto e de fácil comunicação em todos os níveis da organização, os prazos apertados e o volume de trabalho não permitiam que as equipes trocassem experiências entre si. Assim, a boa comunicação com o objetivo de transferência de conhecimento funciona efetivamente apenas dentro de cada célula de trabalho. Outra característica observada foram os altos índices de turnover registrados, de cerca de 25% ao ano.

No que tange ao uso de tecnologias, a empresa possui diversos sistemas de informação instalados. Existe grande facilidade na instalação de novos sistemas devido, principalmente, às características da empresa. Porém, esse domínio da tecnologia da informação que poderia ser positivo, acabava gerando uma instabilidade no uso dos sistemas que muitas vezes eram usados por um curto período de tempo e acabavam sendo esquecidos e descartados.

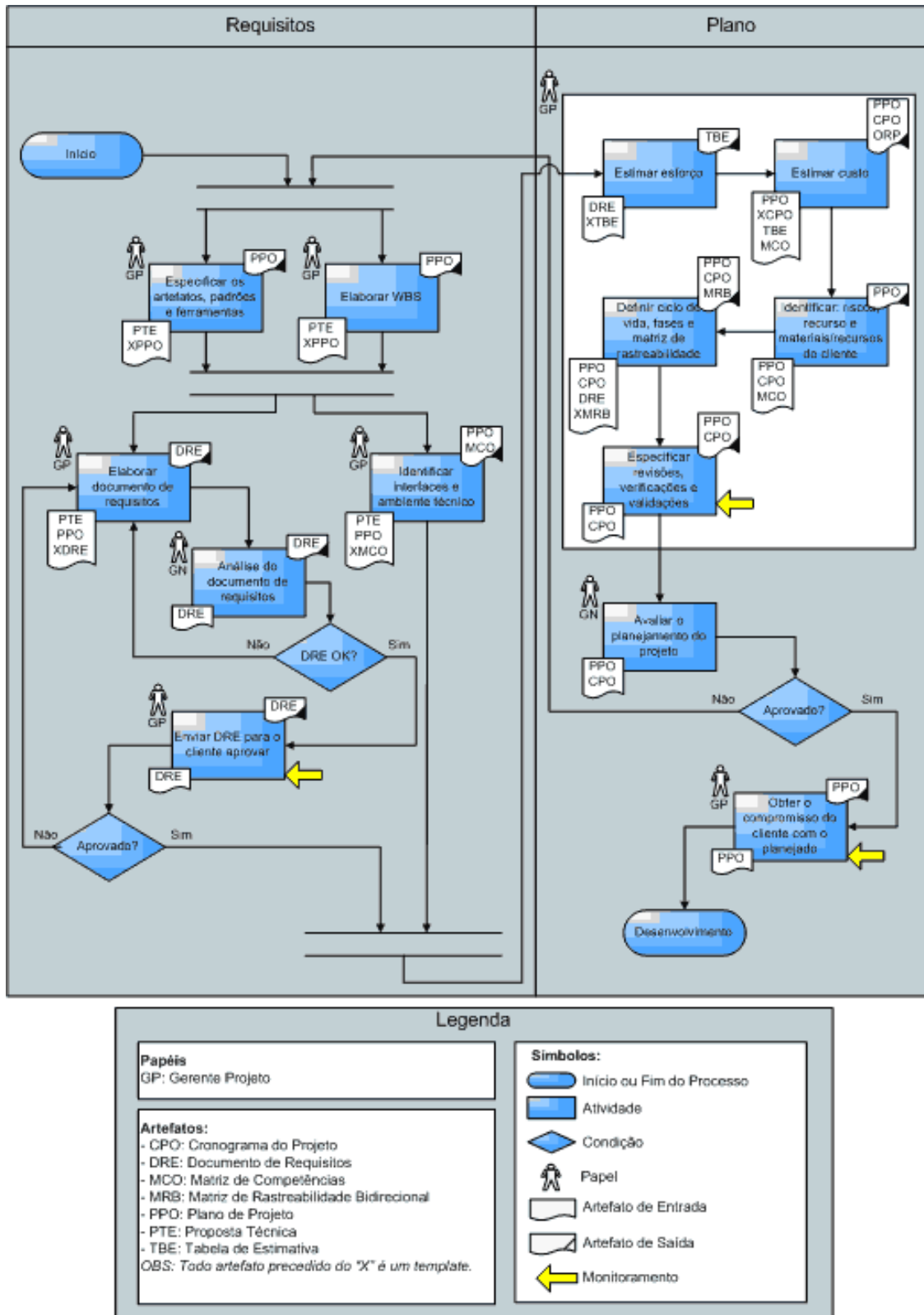


Figura 15 - Antigo grupo de processos de planejamento padrão da B2ML (parte 1)

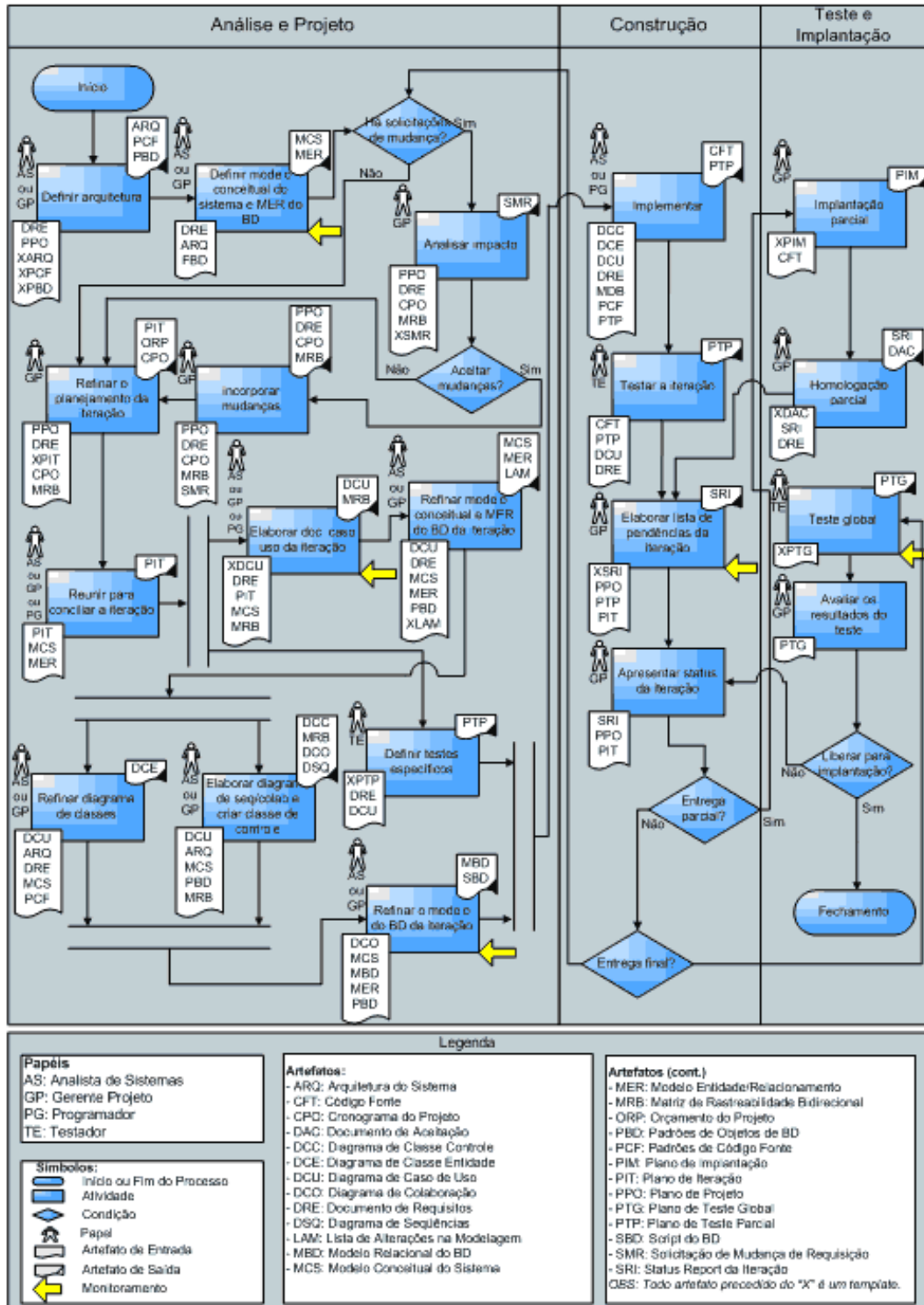


Figura 16 - Antigo grupo de processos de desenvolvimento padrão da B2ML (parte 2)

Essa prática de frequente experimentação tem trazido benefícios para a informatização da organização. Porém, é certo também que essa prática prejudicou a

utilização por toda a empresa de alguns desses sistemas que foram instalados e passaram a ser usados por alguns colaboradores, mas não por todos e posteriormente teve seu uso descontinuado.

Atualmente, alguns sistemas ainda são de muita valia para a empresa. Entre os sistemas com o uso mais bem-sucedido destacam-se os softwares de comunicação instantânea e o sistema de *e-mails*, além do software de gerenciamento de versões de softwares utilizado no desenvolvimento dos produtos. Muitos outros softwares como *intranets*, repositório de arquivos e sistemas de gestão têm questionável eficiência e utilização dentro organização.

Durante o ano de 2007, a empresa buscou sem muito sucesso automatizar algumas atividades do desenvolvimento de produtos. Uma das tentativas foi de utilizar o software de gestão de projetos b2mlProject, desenvolvido pela própria empresa. O b2mlproject é um sistema de gerência de projetos em código aberto, baseado no projeto *dotProject* (uma aplicação *web* de código aberto para a gestão de projetos, desenvolvido por voluntários), de fácil utilização, com um conjunto de funcionalidades e características que o tornam indicado para implementação em ambientes corporativos, pois atende a diversas necessidades de gerentes e Escritórios de Projetos (DOTPROJECT, 2009). Este sistema é uma aplicação *web* e seu acesso é feito através de um navegador, independe de sistema operacional e sem necessidade de instalação na máquina do usuário. O b2mlproject unifica informações de empresas, seus projetos e suas tarefas, permitindo que vários usuários colaborem com vários projetos sem complicação. A figura 17 mostra a visualização das tarefas de um projeto e a figura 18 mostra o gráfico de Gantt gerado por um projeto.



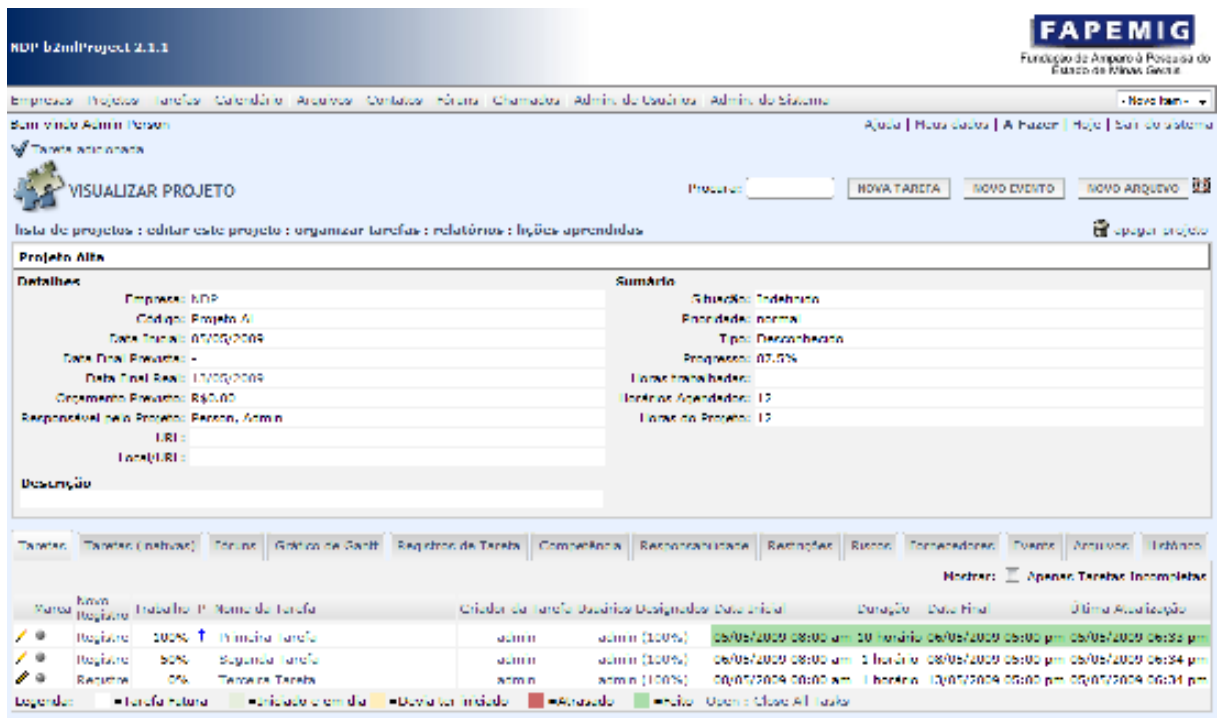


Figura 17 - Visualização das Tarefas do Projeto no b2mlProject

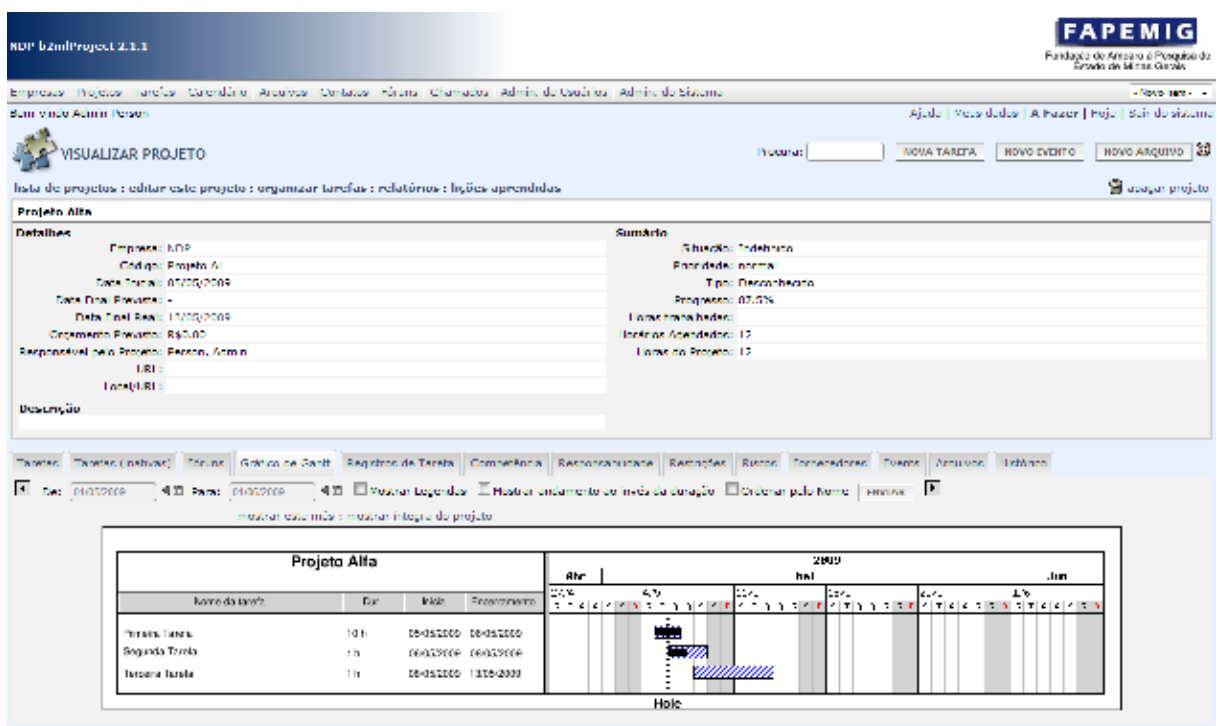


Figura 18 - Gráfico de Gantt exibido pelo b2mlProject

O sistema permite informar usuários de suas associações a tarefas (via *e-mail*) e dar lembretes sobre prazos próximos ao fim. Mostra calendários com visões diferentes (mensal, semanal e diária) e o gráfico de Gantt de cada projeto. Ele permite ainda realizar fóruns relacionados a projetos, fazer o gerenciamento do risco, das responsabilidades, dos fornecedores e criar um repositório de arquivos relacionados a projetos e tarefas.

Entretanto, o sistema b2mlProject não aderiu bem às necessidades da empresa. A sua falta de integração com o ambiente produtivo fez com que ele não fosse utilizado pelos times. E a equipe voltou a gerenciar os projetos sem qualquer ferramenta ou método formal.

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 Excelência em Manufatura e Tecnologia da Inovação (NEMATI) da UNIFEI. Este órgão criou na INCIT um núcleo de desenvolvimento de produtos (NDP).

O Núcleo de Desenvolvimento de Produtos visa mitigar os problemas correlatos ao gerenciamento de processos e o desenvolvimento de produtos encontrados nas empresas incubadas na INCIT. Para isso, o NDP capacita e assessora os pesquisadores e colaboradores das empresas incubadas, além de desenvolver e disseminar pesquisas sobre abordagens e técnicas de apoio ao processo de desenvolvimento de produtos nas empresas. Com isso, o núcleo espera aumentar a eficácia e eficiência e reduzir o tempo de desenvolvimento de produtos nas empresas incubadas.

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 semi-abertas 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;
- 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;
- 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. Neste caso, concluiu-se que o métodos de aplicação mais adequada seria o *Scrum*, que atende especificamente às necessidades de empresas da área de softwares e possui diversas vantagens, já abordadas nos capítulos anteriores.

## **4.2. Planejamento da implantação**

No momento em que havia um claro diagnóstico sobre a realidade da organização, os pesquisadores iniciaram a prática, que ocorreu através 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.

No dia 7 de abril de 2008 os pesquisadores do NDP se reuniram com o objetivo de detectar as principais dificuldades em relação à implementação do *Scrum* e divulgar 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 produto final principal 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. Espera-se que seu uso em arranjos empresariais estabeleça diferenciais competitivos para as empresas com a melhoria do processo de compra, a redução dos custos operacionais e das equipes de compra, o aumento dos fornecedores participantes do processo, a diminuição de erros no processo de compra e a troca de informações práticas entre as empresas. Contudo, para que estes benefícios esperados do software fossem viabilizados, houve a necessidade de um contato permanente com os potenciais clientes para que os requisitos estivessem de acordo com os anseios dos clientes e atualizados.

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.

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 no capítulo 2, a implantação do *Scrum* seguiu os passos do método de Kniberg (2007).

### 4.3. 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 do projeto.

A equipe do projeto era composta por seis pessoas, sendo:

- Um doutor em logística;
- Um engenheiro de produção;
- Três engenheiros da computação;
- 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 no 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 é o membro da equipe que tinha mais conhecimento prévio nas regras de negócio do cliente em potencial do sistema. Preferencialmente, este papel deve 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. Meses antes, o Dono do Produto havia realizado um levantamento de requisitos através de duas atividades: *benchmarking* e entrevistas às empresas. Nesta fase, o Dono do Produto trabalhou com os clientes em potencial e os usuários finais potenciais do sistema para descobrir mais informações sobre o domínio da aplicação. Pode-se chamar esta ação de levantamento de requisitos iniciais. Esta envolveu diversas pessoas de diferentes áreas das empresas visitadas. O Dono do Produto realizou diversas reuniões e entrevistas nestas empresas. Foram empregadas diversas técnicas de elucidação de requisitos, com o intuito de aumentar o comprometimento dos desenvolvedores com a solução que se desejava construir. Os arranjos produtivos locais que foram visitados e contribuíram para o projeto foram:

- Santa Rita do Sapucaí/MG (Arranjo produtivo local de eletrônica);
- Ipatinga/MG (Arranjo produtivo local de aço);
- São João Nepomuceno/MG (Arranjo produtivo local de vestuário);
- Ubá/MG (Arranjo produtivo local de móveis).

Percebeu-se que é muito importante que os requisitos sejam extraídos de maneira correta e objetiva. Também percebeu-se que, como o sistema será utilizado por uma grande quantidade de pessoas, é recomendado que as entrevistas fossem feitas com a presença de um conjunto de representantes do cliente em potencial divididos por áreas de atuação e outros especialistas das áreas de interesse.

Para ilustrar este processo, a Figura 19 mostra um dos diagramas gerados pela equipe durante a modelagem. Trata-se de um diagrama de caso de uso que descreve a funcionalidade de “Cotação de Produtos”. Estes diagramas são importantes, pois descrevem as funcionalidades propostas para o sistema.

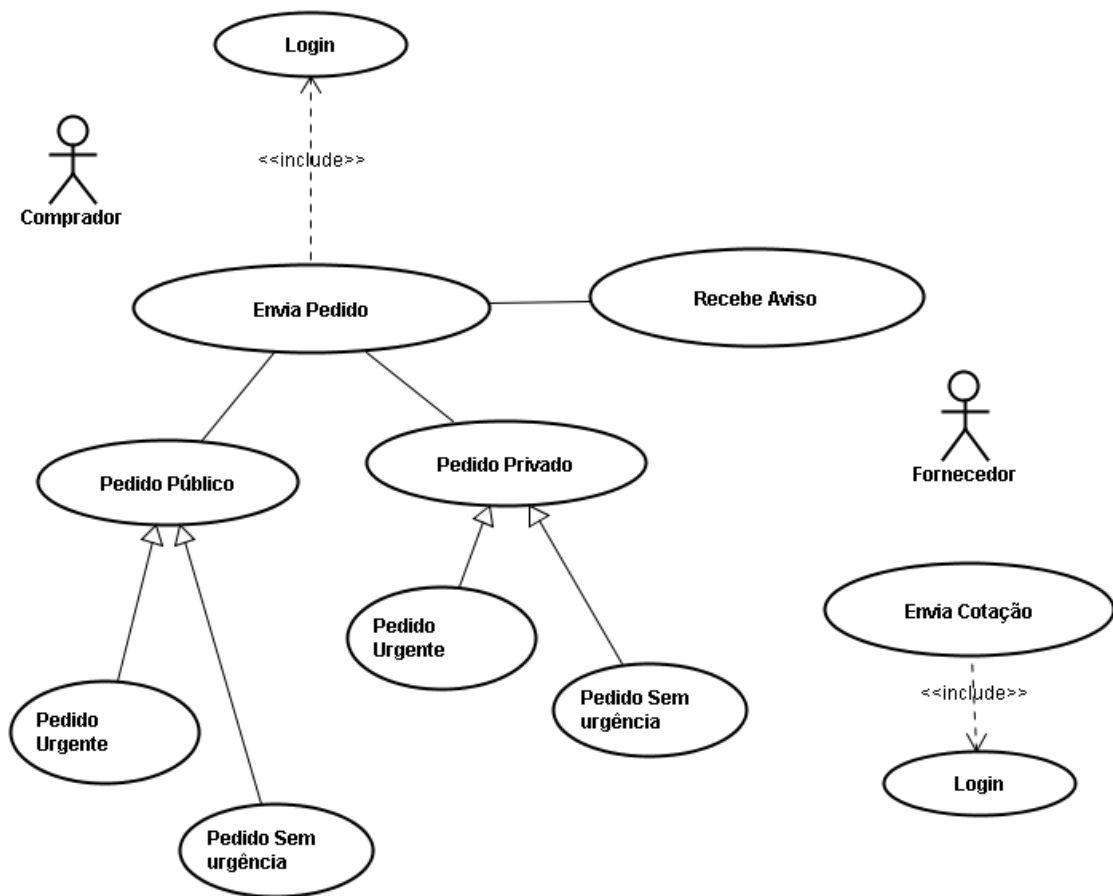


Figura 19- Diagrama de caso de uso de Cotação

Depois de definidos os requisitos do sistema, partiu-se para a modelagem dos requisitos. A modelagem apresentou um papel fundamental na comunicação do que deve ser programado para todos da equipe. Nesta modelagem, foram definidos os

processos do software, como o processo de cotação, compra conjunta, cadastro de produtos etc. Para ilustrar esta fase do projeto, a Figura 20 mostra um diagrama de atividade do cadastro de produtos do sistema. Estes diagramas foram importantes para a equipe porque eles mostram as atividades e as mudanças de uma atividade para outra com os eventos ocorridos em alguma parte do sistema. Desta forma, o desenvolvedor pode ver claramente qual é o processo que o sistema deve seguir em cada atividade.

Após a conclusão da modelagem inicial, o Dono do Produto 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 os seguintes itens do *Scrum*: *Scrum Master*, Dono do Produto, *Sprint*, *Backlog* do Produto e *Backlog* do *Sprint* (como mostra a figura a 21), de acordo com a proposta de Kniberg (2007).

As tabelas 15 a 28 mostram quais ações foram implementadas nesta primeira iteração. 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).

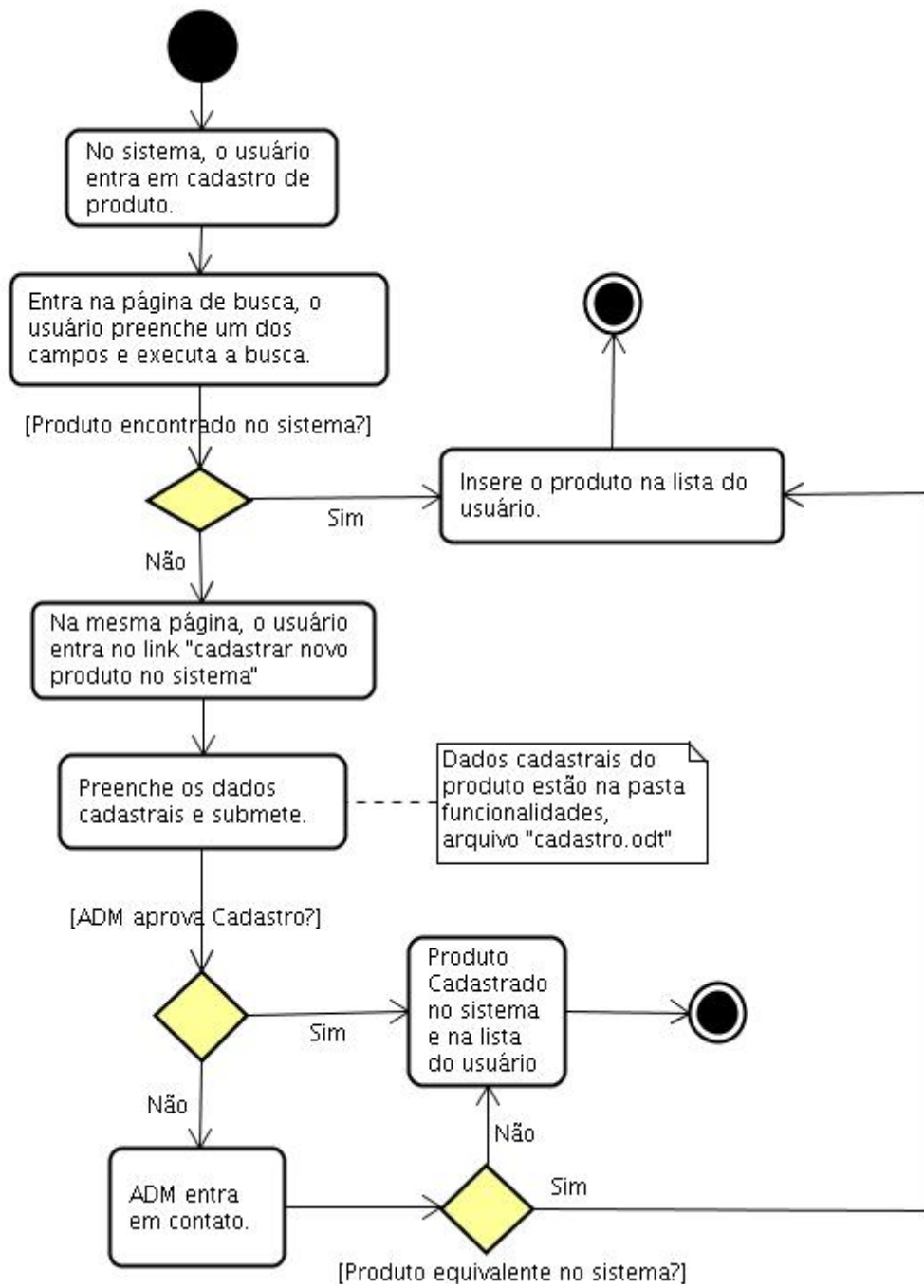


Figura 20 - Diagrama de atividades (processo) do cadastro de produtos



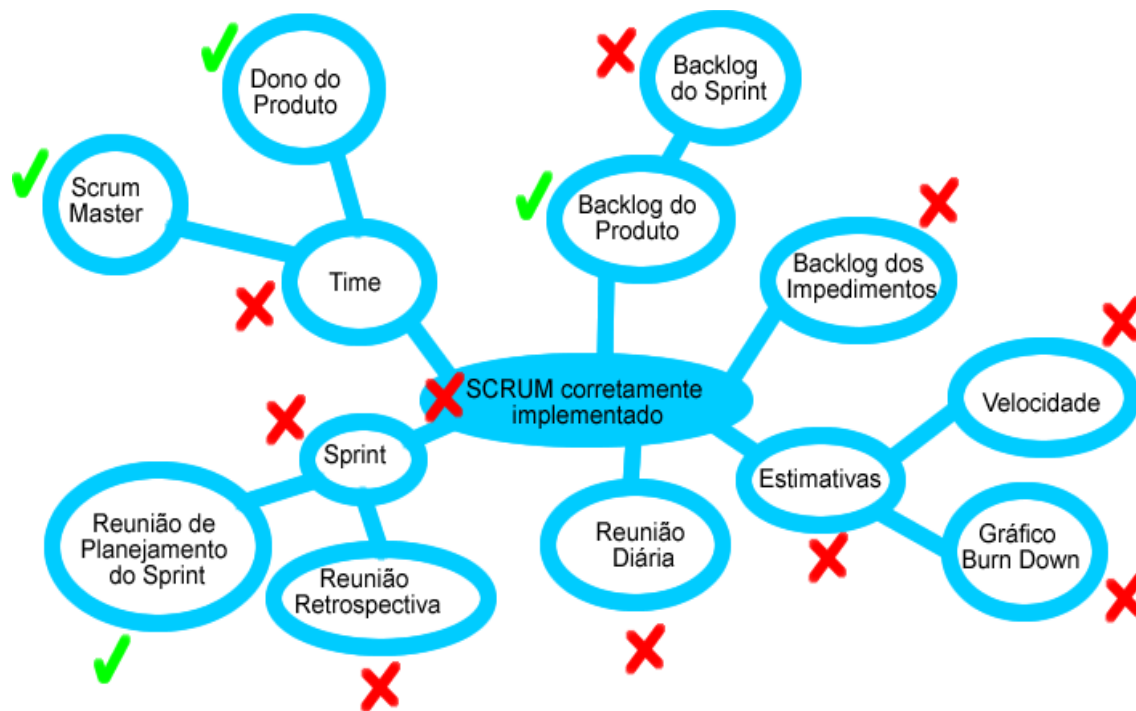


Figura 21 - Itens implantados na primeira iteração

Tabela 15 - Time: ações tomadas na primeira iteração

Time	X	Trabalham lado a lado.
	✓	Colaboram entre si.
	✓	Desempenham várias funções
	✓	Admitem problemas e pedem ajuda ao <i>Scrum Master</i> .
	✓	Ajudam-se mutuamente.
	X	Aceitam responsabilidades e se comprometem.
	X	Não fazem hora extra sistematicamente.

Legenda:

X	Não - implementado
✓	Implementado

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.

Tabela 16 - *Scrum Master*: ações tomadas na primeira iteração

<i>Scrum</i>	✓	O time tem um.
	✓	Trabalha sempre ao lado de seu time.

<i>Master</i>	✓	Resolve os impedimentos.
---------------	---	--------------------------

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.

Tabela 17 - Dono do Produto: ações tomadas na primeira iteração

Dono do Produto	✓	O time tem um e ele está disponível.
	✓	Entende o produto e as necessidades do cliente.
	✓	Sabe quais as funcionalidades prioritárias.
	✓	Faz o time trabalhar nas funcionalidades prioritárias.

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.

Esse software permite que sejam colocadas todas as funcionalidades propostas para o sistema, assim como o nível de prioridade e a estimativa de trabalho. Mais detalhes sobre a maneira com que esta implementação foi realizada serão mostrados no “*Backlog* do *Sprint*”, que também utilizou o mesmo sistema.

Tabela 18 - *Backlog* do Produto: ações tomadas na primeira iteração

<i>Backlog</i> do Produto	✓	Tem total controle do <i>Backlog</i> do Produto.
	✓	Está acessível e visível.
	✓	É atualizado antes de cada <i>Sprint</i> .
	✓	Contém somente funcionalidades.
	✓	Cada funcionalidade contém uma definição de si.

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.

Tabela 19 - Estimativas: ações tomadas na primeira iteração

Estimativas	✘	O dono do produto recebe da equipe estimativas de quantidade de trabalho de cada funcionalidade.
	✘	O time tem liberdade de estimar.
	✘	Todos os times participam das estimativas.

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*.

Tabela 20 - Reunião de Planejamento do *Sprint*: ações tomadas na primeira iteração

Reunião de Planejamento do <i>Sprint</i>	✔	Todos os membros do time participam.
	✔	Ela resulta em plano do <i>Sprint</i> e <i>Backlog</i> do <i>Sprint</i> .
	✔	Há consenso entre os membros do time.
	✔	Time se compromete com o planejado.

O item “*Sprint*” foi parcialmente implementado, pois algumas de suas ações foram implementadas e outras não. Devido à realização do *Sprint*, foi observado uma melhoria no ânimo da equipe justificado, segundo os membros, pelos resultados tangíveis emergidos em cada *Sprint*, enquanto que em projetos tradicionais os mesmos são obtidos somente no fim. É importante salientar que durante cada *Sprint*, eram realizados testes de unidade, que é a fase do processo de teste em que se testaram as menores unidades de software desenvolvidas. Ao longo da primeira iteração do projeto, foram realizados seis *Sprints*. Os *Sprints* estão listados na tabela 21. 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.

A figura 22 mostra de que maneira os *Sprints* eram visualizados pelo sistema *Trac*. Algumas funcionalidades foram iniciadas no primeiro *Sprint* e não acabaram no mesmo, o que não é recomendado. Outro erro cometido foi o fato de que os integrantes do time não seguiam as prioridades do *Backlog* do *Sprint*.

Tabela 21- Duração dos *Sprints* do projeto.

Item	Duração
<i>Sprint 1</i>	6 semanas
<i>Sprint 2</i>	3 semanas
<i>Sprint 3</i>	3 semanas
<i>Sprint 4</i>	2 semanas
<i>Sprint 5</i>	2 semanas
<i>Sprint 6</i>	2 semanas

## Roadmap

### Milestone: 2 - Sprint Backlog - Cadastro



### Milestone: 4 - Sprint Backlog - Compra Conjunta



### Milestone: 5 - Sprint Backlog - Administração de Corações



### Milestone: 6 - Sprint Backlog - Administração de Grupo

8 months late (08/11/08)



### Milestone: Product Backlog

No date set

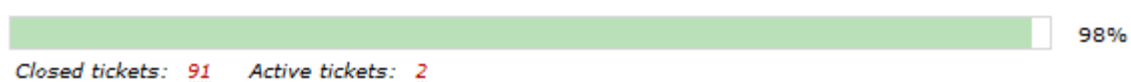


Figura 22 - *Sprints* do projeto no *Trac*

Tabela 22 - *Sprint*: ações tomadas na primeira iteração

<i>Sprint</i>	✓	O time entrega um produto funcionando no final de cada <i>Sprint</i> .
	✗	O time segue rigorosamente as prioridades do <i>Backlog</i> do <i>Sprint</i> .
	✗	O time age corretivamente quando está atrasado.
	✓	O time alerta o <i>Scrum Master</i> quando há problemas.
	✓	Time sabe onde estão informações para implementar funcionalidades.
	✓	Os problemas são resolvidos quando ocorrem.
	✗	A duração do <i>Sprint</i> é o mesmo em todos os <i>Sprints</i> .
	✓	O intervalo entre dois <i>Sprints</i> é de no máximo um dia.
	✓	Todos os envolvidos sabem sobre o <i>Sprint</i> .
	✗	As funcionalidades são terminadas no mesmo <i>Sprint</i> que começa.

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 deve ser evitado. Outra falha foi que o time ainda tinha o hábito de esperar pelo gerente do projeto, ou seja, ainda não tinham a capacidade de se autogerenciar.

Tabela 23 - Reunião Diária: ações tomadas na primeira iteração

Reunião Diária - <i>Daily Scrum</i>	✗	Acontecem no mesmo lugar e horário todos os dias.
	✗	Começam e terminam pontualmente.
	✗	Todos os membros do time estão presentes.
	✓	Todos respondem às três perguntas-chave.
	✓	Não há interrupções.
	✓	O dono do produto a visita regularmente.
	✗	Os membros da equipe buscam e decidem as tarefas.
	✗	Os membros da equipe se cobram as realizações das tarefas.

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 conseqüências disto foram a repetição dos mesmos erros em *Sprints* consecutivas e a diminuição da possibilidade de melhorias.

Tabela 24 - Reunião Retrospectiva: ações tomadas na primeira iteração

Reunião Retrospectiva	✗	Ela existe ao final de cada <i>Sprint</i> .
	✗	Todos participam.

	✘	Resulta em sugestões concretas de melhoria.
	✘	Algumas dessas sugestões são implementadas de fato.

O item “*Impediment Backlog*” não foi implementado. O time não utilizou nenhuma técnica formal para catalogar os riscos e ameaças.

Tabela 25 – *Backlog* de Impedimentos: ações tomadas na primeira iteração

<i>Backlog</i> de Impedimentos	✘	O time tem um.
	✘	Está visível a todos.
	✘	Está sempre atualizado.
	✘	Tem itens priorizados.
	✘	Seus itens são escalados para o dono do produto.

O item “Velocidade” não foi implementado porque não se implementou o item “Estimativas” que na prática é um pré-requisito.

Tabela 26 - Velocidade: ações tomadas na primeira iteração

Velocidade	✘	A velocidade de desenvolvimento é mensurada.
	✘	A velocidade é utilizada para ações corretivas.
	✘	Seu registro é utilizado para melhorar futuras estimativas.

O item “Gráfico *Burndown*” não foi implementado também porque o item “Estimativas” é seu pré-requisito.

Tabela 27 - Gráfico *Burndown*: ações tomadas na primeira iteração

Gráfico <i>Burndown</i>	✘	O time tem um.
	✘	É visível para toda a equipe.
	✘	É atualizado, no mínimo, diariamente pelo <i>Scrum Master</i> .
	✘	O time age corretivamente caso o gráfico <i>Burndown</i> mostre que o desenvolvimento está fora do cronograma.

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*. As figuras 23 e 24 mostram a tela de edição de um *Sprint*. Nela é possível ver a descrição e o andamento do *Sprint* (no exemplo utilizado, o *Sprint* já havia sido encerrado).



Figura 23 – Tela de edição do *Sprint*



Figura 24 - Aproximação da tela de edição do *Sprint*

O *Backlog* do *Sprint*, em si, é mostrado nas figuras 25 e 26. Pode-se ver a identificação (código) da funcionalidade, uma rápida descrição da mesma, seu proprietário (o membro do time que se responsabilizou por ela), seu tipo e sua prioridade. As figuras 27 e 28 mostram o detalhamento da funcionalidade. Desta forma, cada membro do time pode ver a descrição clara da funcionalidade que está implementando.

HOME

logged in as biamardo | [Logout](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

[Wild](#) | [Timeline](#) | [Roadmap](#) | [Browse Source](#) | **[View Tickets](#)** | [New Ticket](#) | [Search](#)

[Available Reports](#) | [Custom Query](#)

**Custom Query** (1 tickets)

Filters

Status:  new  assigned  reopened  closed

Keywords:

And/Or:

Group results by:   ascending  Show full description under each result

Ticket	Summary	Owner	Type	Priority ▲	Component	Resolution
3	Comprador - Adicionar usuário	bruno	nova funcionalidade	altissima	Comprar	fixed
4	Comprador - Editar usuário	orioli	nova funcionalidade	altissima	Comprar	fixed
45	bugfix do nova cotação	wellison	bug	altissima	Administracao	fixed
2	Comprador - Visualizar usuários.	orioli	nova funcionalidade	alta	Comprar	fixed
6	Produtos - Cadastrar novo produto	lazaro	nova funcionalidade	alta	Comprar	fixed
8	Fornecedores - ver lista de meus fornecedores	orioli	nova funcionalidade	alta	Comprar	fixed
9	Produto - Ver a lista de meus produtos	lazaro	nova funcionalidade	alta	Comprar	fixed
5	Produtos - Colocar um produto na lista de meus produtos	wellison	nova funcionalidade	media	Comprar	fixed
7	Fornecedores - adicionar fornecedores na lista de meus fornecedores	orioli	nova funcionalidade	media	Comprar	fixed
20	produto - buscar por produto	lazaro	nova funcionalidade	baixa	Comprar	fixed
19	Vendedor - Funcionalidades de usuário	lazaro	nova funcionalidade	baixissima	Vendas	fixed

Note: See [TracQuery](#) for help on using queries.

Download in other formats:  
[RSS Feed](#) | [Comma delimited Text](#) | [Tab delimited Text](#)

Powered by Trac 0.10.3.1  
By Cogeval Software.

©2014 CIGEMAG  
<http://www.b2m.com.br>

Figura 25 - Tela que mostra o *Sprint*

Ticket	Summary	Owner	Type	Priority ▲
3	Comprador - Adicionar usuário	bruno	nova funcionalidade	altissima
4	Comprador - Editar usuário	orioli	nova funcionalidade	altissima
45	bugfix do nova cotação	wellison	bug	altissima
2	Comprador - Visualizar usuários.	orioli	nova funcionalidade	alta
6	Produtos - Cadastrar novo produto	lazaro	nova funcionalidade	alta
8	Fornecedores - ver lista de meus fornecedores	orioli	nova funcionalidade	alta
9	Produto - Ver a lista de meus produtos	lazaro	nova funcionalidade	alta
5	Produtos - Colocar um produto na lista de meus produtos	wellison	nova funcionalidade	media
7	Fornecedores - adicionar fornecedores na lista de meus fornecedores	orioli	nova funcionalidade	media
20	produto - buscar por produto	lazaro	nova funcionalidade	baixa
19	Vendedor - Funcionalidades de Usuário	lazaro	nova funcionalidade	baixissima

Figura 26 - Aproximação da tela que mostra o *Sprint*



Search

logged in as [bernardo](#) | [Logout](#) | [Settings](#) | [Help/Guide](#) | [About Trac](#)

Wiki | Timeline | Roadmap | Browse Source | **View Tickets** | New Ticket | Search

[Back to Query](#) | [Next Ticket](#) →

### Ticket #29 (closed nova funcionalidade: fixed)

**Lider - Criação da Cotação** Opened 9 months ago  
Last modified 3 seconds ago

Reported by:	bernardo	Assigned to:	orioli
Priority:	altissima	Milestone:	4- Sprint Backlog - Compra Conjunta
Component:	Compras	Keywords:	
Cc:			

Description (Last modified by [bernardo](#)) [Reply](#)

O comprador líder da cotação cria a cotação (segundo basicamente os passos já seguidos na cotação individual). A diferença é que ele cria uma data a mais, que é a data limite para a entrada de novos compradores. Além disso, ele também pode convidar empresas para que entrem na cotação. Não esquecer que ele poderá também convidar os outros compradores para participar da cotação, até o prazo limite (data limite para a entrada de novos compradores). Qualquer dúvida, veja o fluxograma que está no repositório ou pergunte pro Owner.

**Attachments**

[Attach File](#)

Action

leave as closed

reopen ticket

[Preview](#) | [Submit changes](#)

Download in other formats:  
[Comma-delimited Text](#) | [Tab-delimited Text](#) | [RSS Feed](#)


 Powered by [Trac 0.10.3.1](#)  
By [Edgewall Software](#) B2ML SISTEMAS  
<http://www.b2ml.com.br/>

Figura 27 - Tela de detalhamento da funcionalidade

### Ticket #29 (closed nova funcionalidade: fixed)

**Lider - Criação da Cotação** Opened 9 months ago  
Last modified 3 seconds ago

Reported by:	bernardo	Assigned to:	orioli
Priority:	altissima	Milestone:	4- Sprint Backlog - Compra Conjunta
Component:	Compras	Keywords:	
Cc:			

Description (Last modified by [bernardo](#)) [Reply](#)

O comprador líder da cotação cria a cotação (segundo basicamente os passos já seguidos na cotação individual). A diferença é que ele cria uma data a mais, que é a data limite para a entrada de novos compradores. Além disso, ele também pode convidar empresas para que entrem na cotação. Não esquecer que ele poderá também convidar os outros compradores para participar da cotação, até o prazo limite (data limite para a entrada de novos compradores). Qualquer dúvida, veja o fluxograma que está no repositório ou pergunte pro Owner.

Figura 28 - Aproximação da tela de detalhamento da funcionalidade

Tabela 28 - *Backlog* do *Sprint*: ações tomadas na primeira iteração

<i>Backlog</i> do <i>Sprint</i>	✓	O time tem um.
	✓	É visível por toda a equipe.
	✓	É atualizado diariamente (e facilmente) por toda a equipe.
	✗	A estimativa de trabalho para as tarefas é atualizada diariamente.
	✓	As funcionalidades e tarefas são facilmente distinguíveis.
	✓	Estão claras quais tarefas foram originadas por cada funcionalidade.

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 considera 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 o sucesso do mesmo.

#### 4.4. 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* (conforme a figura 29). Desta forma, para a total implementação do *Scrum*, somente os itens relacionados às Estimativas e ao *Backlog* de Impedimentos ficariam pendentes.

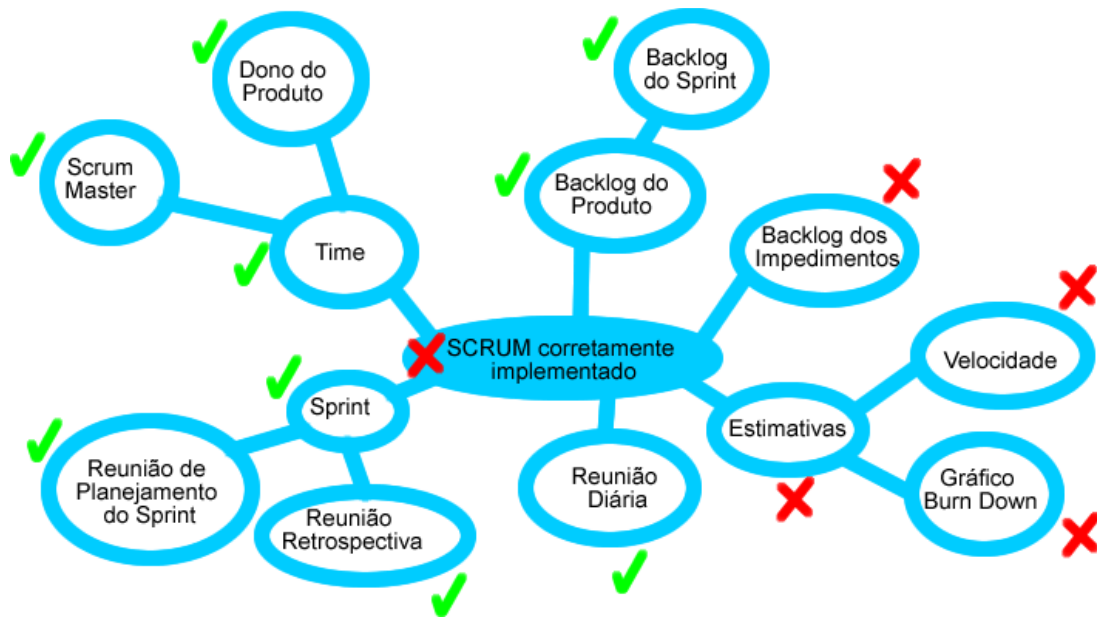


Figura 29 - Itens implantados na segunda iteração

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, como mostra a tabela 29. Percebeu-se que os membros do time conseguiram romper a barreira inicial da falta da figura do gerente de projetos e passou a trabalhar bem com autosuficiência 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.

Tabela 29 - Time: ações tomadas na segunda iteração

Time	✓	Trabalham lado a lado.
	✓	Colaboram entre si.
	✓	Desempenham várias funções.
	✓	Admitem problemas e pedem ajuda ao <i>Scrum Master</i> .
	✓	Ajudam-se mutuamente.
	✓	Aceitam responsabilidades e se comprometem.
	✓	Não fazem hora extra sistematicamente.

Na segunda iteração começou-se um esforço no sentido de se realizar as estimativas, como mostra a tabela 30. Os membros do time que foram treinados na

metodologia de pontos de função passaram a tentar estimar cada funcionalidade. Após a 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.

Tabela 30 - Estimativas: ações tomadas na segunda iteração

Estimativas	✘	O dono do produto recebe da equipe estimativas de quantidade de trabalho de cada funcionalidade.
	✔	O time tem liberdade de estimar.
	✔	Todos os times participam das estimativas.

Apesar do esforço para estimar, ainda não foi possível medir a velocidade do desenvolvimento de maneira confiável.

Tabela 31 - Velocidade: ações tomadas na segunda iteração

Velocidade	✘	A velocidade de desenvolvimento é mensurada.
	✘	A velocidade é utilizada para ações corretivas.
	✘	Seu registro é utilizado para melhorar futuras estimativas.

O gráfico *Burndown*, como mostra a tabela 32, também não foi feito, pois não se tinha confiança nas estimativas realizadas.

Tabela 32 - Gráfico *Burndown*: ações tomadas na segunda iteração

Gráfico <i>Burndown</i>	✘	O time tem um.
	✘	É visível para toda a equipe.
	✘	É atualizado, no mínimo, diariamente pelo <i>Scrum Master</i> .
	✘	O time age corretivamente caso o gráfico <i>Burndown</i> mostre que o desenvolvimento está fora do cronograma.

No item “*Sprint*” foram corrigidos os problemas da primeira iteração, como mostra a tabela 33. 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.

Tabela 33 - *Sprint*: ações tomadas na segunda iteração

<i>Sprint</i>	✓	O time entrega um produto funcionando no final de cada <i>Sprint</i> .
	✓	O time segue rigorosamente as prioridades do <i>Backlog</i> do <i>Sprint</i> .
	✓	O time age corretivamente quando está atrasado.
	✓	O time alerta o <i>Scrum Master</i> quando há problemas.
	✓	Time sabe onde estão informações para implementar funcionalidades.
	✓	Os problemas são resolvidos quando ocorrem.
	✓	A duração do <i>Sprint</i> é o mesmo em todos os <i>Sprints</i> .
	✓	O intervalo entre dois <i>Sprints</i> é de, no máximo, um dia.
	✓	Todos os envolvidos sabem sobre o <i>Sprint</i> .
	✓	As funcionalidades são terminadas no mesmo <i>Sprint</i> iniciado.

As reuniões diárias passaram a ser mais produtivas quando o time começou a se autogerenciar corretamente (vide tabela 34). 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.

Tabela 34 - Reunião Diária: ações tomadas na segunda iteração

Reunião Diária - <i>Daily Scrum</i>	✓	Acontecem no mesmo lugar e horário todos os dias.
	✓	Começam e terminam pontualmente.
	✓	Todos os membros do time estão presentes.
	✓	Todos respondem às três perguntas-chave.
	✓	Não há interrupções.
	✓	O dono do produto a visita regularmente.
	✓	Os membros da equipe buscam e decidem as tarefas.
	✓	Os membros da equipe se cobram as realizações das tarefas.

As Reuniões de Retrospectiva, como mostra a tabela 35, 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.

Tabela 35 - Reunião Retrospectiva: ações tomadas na segunda iteração

Reunião Retrospectiva	✓	Ela existe ao final de cada <i>Sprint</i> .
	✓	Todos participam.
	✓	Resulta em sugestões concretas de melhoria.
	✓	Algumas dessas sugestões são implementadas de fato.

Na segunda iteração passou-se a fazer uso do *Backlog* de impedimentos, dado pela tabela 36. Segundo o *Scrum Master*, embora à primeira vista este artefato parecesse pouco importante, sua utilização acaba por mitigar os riscos na prática, já que os mesmos ficam muito mais claros e são lembrados diariamente. A única falha deste item foi não priorizar cada risco.

Tabela 36 – *Backlog* de impedimentos: ações tomadas na segunda iteração

<i>Backlog</i> de impedimentos	✓	O time tem um.
	✓	Está visível a todos.
	✓	Está sempre atualizado.
	✗	Tem itens priorizados.
	✓	Seus itens escalados para o dono do produto.

O *Backlog* do *Sprint*, mostrado na tabela 37, continuou a ser implementado, mas desta vez com as estimativas juntamente com cada funcionalidade.

Tabela 37 - *Backlog* do *Sprint*: ações tomadas na segunda iteração

<i>Backlog</i> do <i>Sprint</i>	✓	O time tem um.
	✓	É visível por toda a equipe.
	✓	É atualizado diariamente (e facilmente) por toda a equipe.
	✓	A estimativa de trabalho para as tarefas é atualizada diariamente.
	✓	As funcionalidades e tarefas são facilmente distinguíveis.
	✓	Estão claras quais tarefas foram originadas por cada funcionalidade.

## 4.5. Fase de ação: iteração final

Na iteração final, o objetivo era implementar todos os itens do *Scrum*, conforme a figura 30.

Os itens *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 implementados normalmente. E os itens relacionados às Estimativas e ao *Backlog* de Impedimentos foram implementados pela primeira vez. Nesta iteração foram realizados seis Sprints de duas semanas cada um.

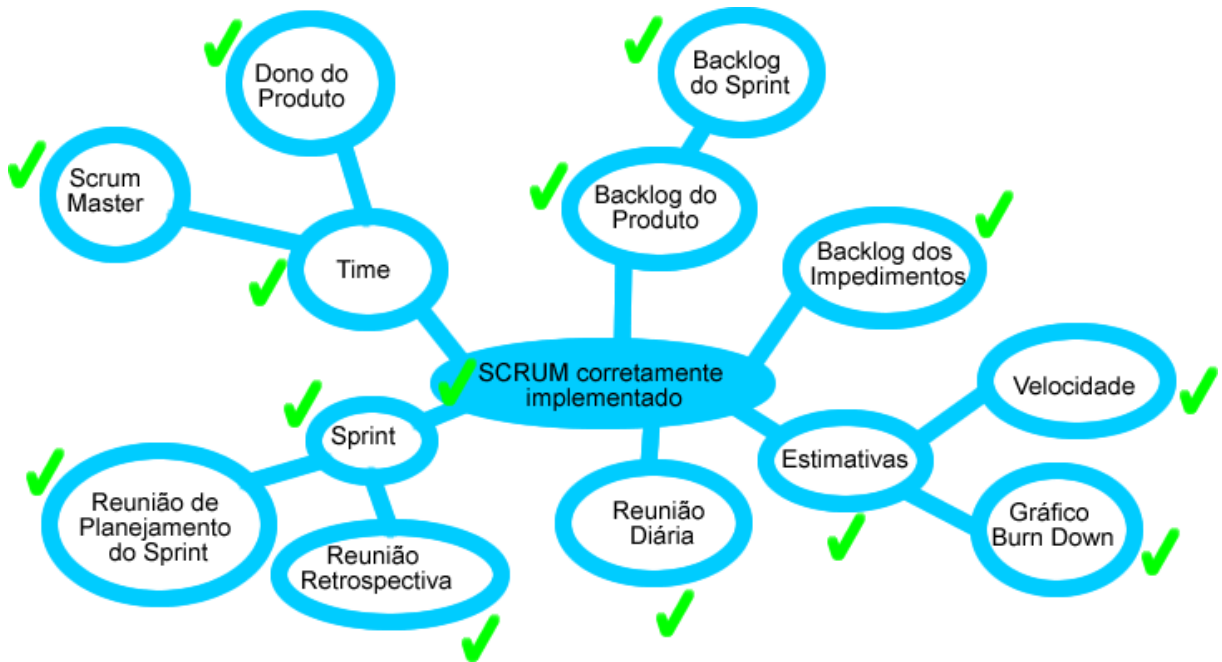


Figura 30 - Itens implantados na iteração final

O *Backlog* de Impedimentos passou a ser totalmente implementado, como mostra a tabela 38. Desta vez, conseguiu-se priorizar a lista de impedimentos de modo que o *Scrum Master* sabia qual deveria ser sua prioridade de ação.

Tabela 38 – *Backlog* de Impedimentos: ações tomadas na iteração final.

<i>Backlog</i> de Impedimentos	✓	O time tem um.
	✓	Está visível a todos.
	✓	Está sempre atualizado.
	✓	Tem itens priorizados.
	✓	Seus itens foram informados para o dono do produto.

Na última iteração todos sabiam que o maior desafio seria os itens de estimativas e velocidade. Por isso, todos os esforços focaram resolver o problema da falta de mensuração de trabalho do projeto.

As estimativas (vide tabela 39) que já haviam sido realizadas na segunda

iteração se consolidaram e a equipe passou a confiar mais nas mesmas, de modo que passou a ser publicado, 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 *et al.* (2007).

Tabela 39 - Estimativas: ações tomadas na iteração final

Estimativas	✓	O dono do produto recebe da equipe estimativas de quantidade de trabalho de cada funcionalidade.
	✓	O time tem liberdade de estimar.
	✓	Todos os times participam das estimativas.

Com a estimativa baseada em pontos de função dando bons resultados, viabilizou-se a mensuração da velocidade do projeto, como mostra a tabela 40. Com isso, a equipe ganhou uma grande confiança no cumprimento dos prazos e pôde agir corretivamente quando se viu desviando do planejado.

Tabela 40 - Velocidade: ações tomadas na iteração final

Velocidade	✓	A velocidade de desenvolvimento é mensurada.
	✓	A velocidade é utilizada para ações corretivas.
	✓	Seu registro é utilizado para melhorar futuras estimativas.

O gráfico *burndown* foi totalmente implementado, como mostra a tabela 41. 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.

Tabela 41 - Gráfico *Burndown*: ações tomadas na iteração final.

Gráfico <i>Burndown</i>	✓	O time tem.
	✓	É visível para toda a equipe.
	✓	É atualizado, no mínimo, diariamente pelo <i>Scrum Master</i> .
	✓	O time age corretivamente caso o gráfico <i>Burndown</i> mostre que o desenvolvimento está fora do cronograma.

A figura 31 mostra o Gráfico *Burndown* da iteração final. Percebe-se que a equipe passou boa parte do tempo bastante próxima do planejado e conseguiu terminar o projeto dois dias antes do prazo final.



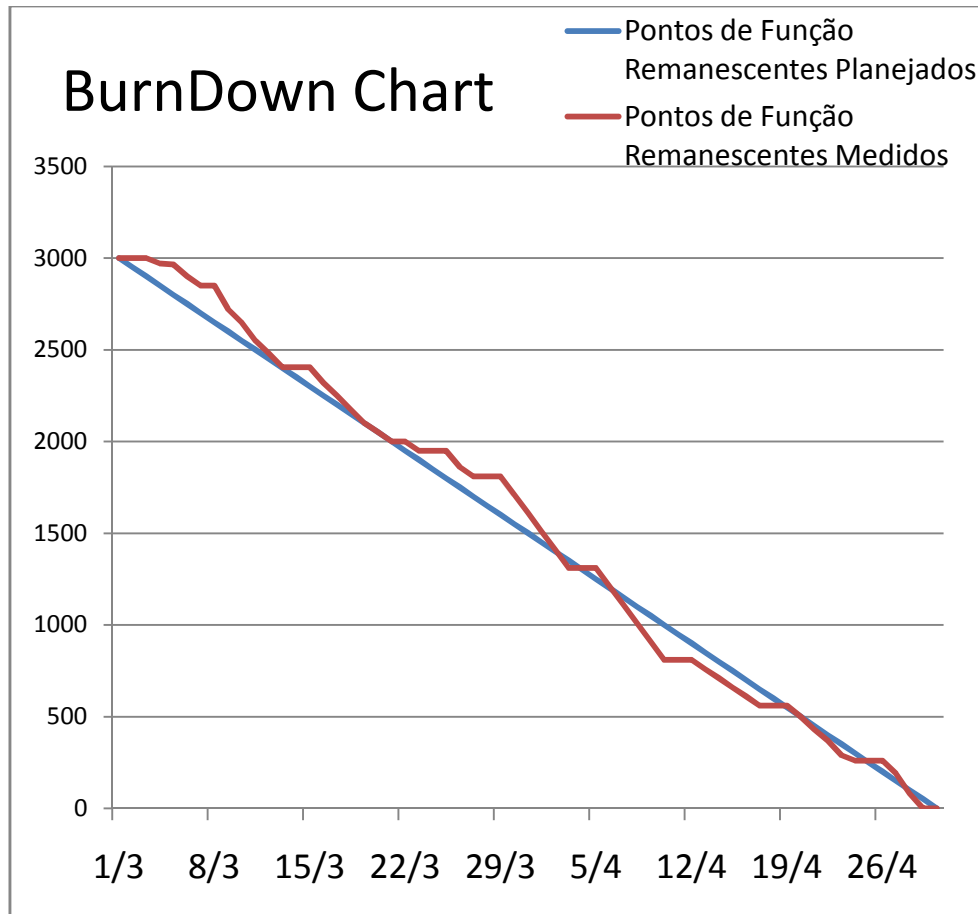


Figura 31 - Gráfico *Burndown* da terceira iteração

#### 4.6. Fase de avaliação: análise dos resultados

Esta etapa final do processo de pesquisa-ação tem dois objetivos principais: verificar e mensurar os resultados das ações no contexto organizacional da pesquisa e suas conseqüências; 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 no IDE “Eclipse”, versão 3.2.1. A codificação completa do projeto foi bastante grande, tendo gerado 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 configura um sistema grande 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 (Tabela 14 do Capítulo 2) e enviadas aos participantes da equipe de implantação. Essas pessoas deveriam ler cada afirmação e responder qual seu grau de concordância à 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 o mesmo 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 mostrada na tabela 42. Os resultados são apresentados na tabela 43.

Tabela 42 - 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.

Tabela 43 – 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

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) apresentaram um bom grau de concordância, o que leva a 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 32 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.

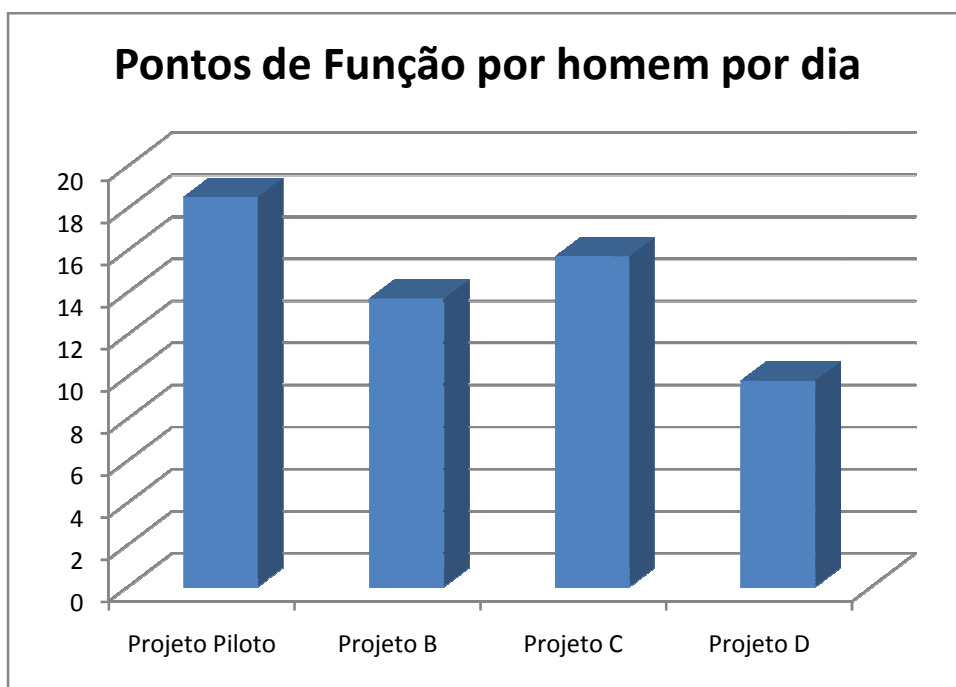


Figura 32 - Produtividade por projeto

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 estava sendo analisados. Também devemos 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 33 mostra a classificação dos projetos, seguindo o modelo usado em Schwaber & 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, é cedo para afirmar que o *Scrum* melhora 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.

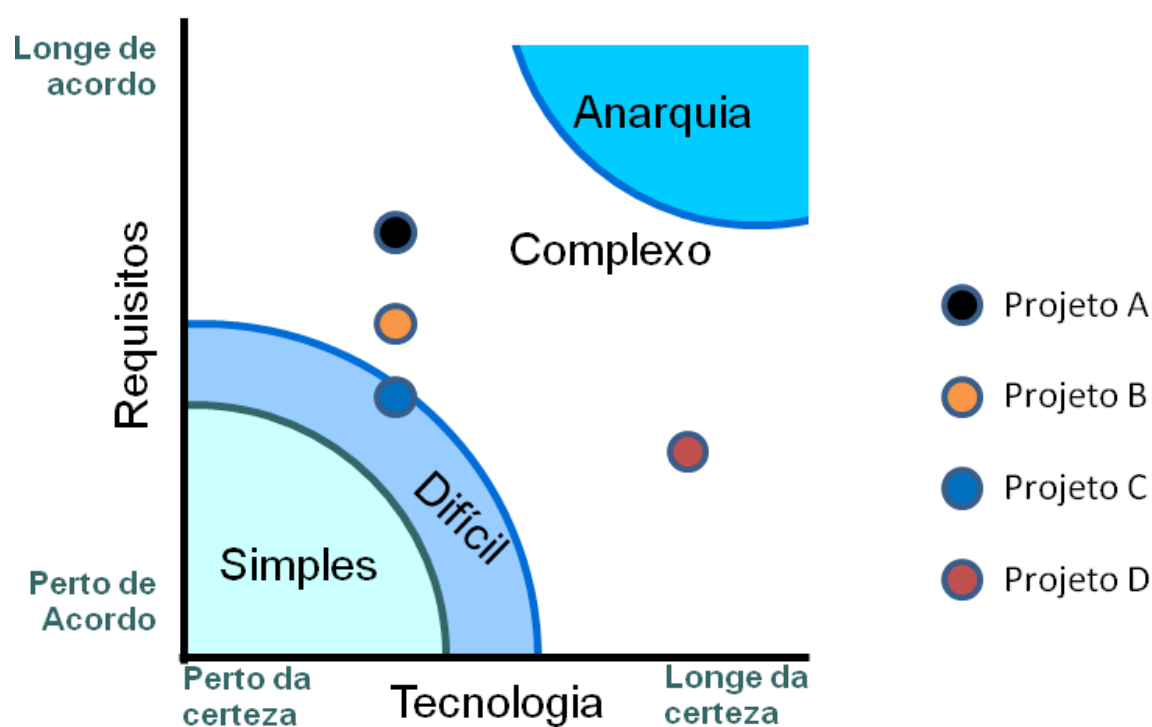


Figura 33 - Classificação dos projetos

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 tem 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 na tabela 44.

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

Tabela 44 - Análise das práticas gerenciais do *Scrum*

Item	Vantagens / Facilidades	Desvantagens / Dificuldades
<i>Backlog</i> 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 à desencontros de horários dos membros da equipe.
<i>Sprint</i>	A divisão do projeto em <i>Sprints</i> aumentou a motivação do time.	Devido à inexperiência da equipe, alguns <i>Sprints</i> tiveram duração muito longa.
Planejamento do <i>Sprint</i>	Sua realização melhorou o planejamento do <i>Sprint</i> .	As primeiras reuniões foram pouco produtivas devido a inexperiência da equipe.
<i>Backlog</i> do <i>Sprint</i>	Sua boa elaboração é crucial para o sucesso ou fracasso do <i>Sprint</i> .	Seu superdimensionamento causou um atraso nos primeiros <i>Sprints</i> .
Revisão do <i>Sprint</i>	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 <i>Sprint</i> .
<i>Backlog</i> de Impedimentos	É uma boa ferramenta para cobrar a resolução de problemas conhecidos, diminuindo os riscos do projeto.	Tende a ser negligenciado pelo próprio <i>Scrum Master</i> , 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 <i>Burndown</i>	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.
<i>Scrum Master</i>	É imprescindível para tirar dúvidas quanto ao processo de desenvolvimento e liderar as cerimônias do <i>Scrum</i> .	Muitas vezes o time o vê como um gerente do projeto, deixando para ele atribuições que deveriam ser do próprio time.

Uma terceira análise foi realizada em um segundo seminário no sentido de buscar lições aprendidas. Assim como no estudo de Rayhan & 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 contra-senso.
- 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 auto-gerenciamento. 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.
- 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.

A tabela 45 mostra a quarta análise realizada, que foi a comparação entre o desenvolvimento de novos produtos com *Scrum* e o antigo processo de desenvolvimento de produtos.

Tabela 45 – 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 auto-gerencia. Todos os membros do time assumem os riscos do projeto e tem comprometimento com seu sucesso.
Contato com o cliente	Somente o gerente do projeto interagiu 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 ficada 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.

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”.

## 5. Conclusão

Neste capítulo será apresentada a conclusão do trabalho, assim como suas limitações e recomendações para pesquisas futuras.

### 5.1 Considerações finais

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 atribuía ao mesmo. A equipe participante da pesquisa sentiu diretamente quatro dos benefícios do *Scrum*, são eles:

- 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);
- aumento de produtividade da equipe.

Além disso, não foi possível investigar se houveram aumentos da qualidade do produto, da satisfação de clientes (e a diminuição de 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. Com isso, se completaria a análise dos benefícios mapeados na pesquisa bibliográfica. Além disso, outra pesquisa interessante 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*.



A equipe também discutiu e avaliou as práticas gerenciais e os artefatos do *Scrum*. Para eles, o método foi 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. Isso corrobora a pesquisa realizada, mostrando que o *Scrum* pode ser empregado como um método adequado para a gestão de projetos de software.

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, 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.

Após a implantação do processo de desenvolvimento de produtos e gestão de projetos na equipe piloto, a cúpula da empresa ficou convencida de que nos próximos meses este deve se consolidar como o processo padrão de desenvolvimento de produtos da empresa. A equipe também considerou que a empresa está pronta para adotar o *Scrum* no desenvolvimento de todos os seus produtos.

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 auto-gerenciamento. 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 administração em todo o processo de implantação do novo método. Sem este apoio, seria muito mais difícil o sucesso desta empreitada, 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.

## **5.2 Limitações e recomendações para pesquisas futuras**

O presente estudo abordou a gestão de desenvolvimento de produtos em uma empresa de base tecnológica. Pela própria natureza da pesquisa-ação, não se buscou criar um conhecimento universal acerca do tema. Pelo contrário, buscou-se gerar uma teoria emergente desenvolvida a partir de uma síntese da prática e da teoria de gestão de desenvolvimento de produtos.

Para isto é interessante a replicação desta pesquisa-ação em outras empresas da mesma natureza, que observem problemas na gestão de desenvolvimento de produtos. 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 seria investigar 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 de trabalho uma análise mais aprofundada da questão da medição de velocidade a estimativas para trabalhos com *Scrum*. Este foi um dos pontos mais críticos deste trabalho e isto pode ocorrer em outras empresas. É notável que este assunto possui um campo de pesquisa muito grande a ser explorado.

## REFERÊNCIAS

ABRAHAMSSON, P.; SALO, O.; RONKAINEN, J.; WARSTA, J. Agile Software Development Methods – Review and Analysis. Espoo, VTT Publications, p. 478, 2002.

AMBLER, S. Agile Modeling, Wiley Computer Publishing. New York, 2002.

ASSOCIAÇÃO BRASILEIRA DAS EMPRESAS DE SOFTWARE, 2008. Disponível em: <http://www.abes.org.br/templ2.aspx?id=305&sub=305>. Acesso em: março/2008.

BARTON, B.; CAMPBELL, E. Implementing a Professional Services Organization Using Type C Scrum. System Sciences, p. 275, 2007.

BATES, C.; YATES, S. Scrum down a software engineer and a sociologist explore the implementation of an agile method. International workshop on Cooperative and human aspects of software engineering, 2008.

BERCZUK, S. Back to basics: The Role of Agile Principles in Success with an Distributed Scrum Team. Agile Conference, p. 382-388, 2007.

BERCZUK, S.; HARRISON, N.; HENNEY, K.; KERIEVSKY, J.; RISING, L.; SCHWABER, K.; WOOLF, B. What's so eXtreme about doing things right. International Conference on Object-Oriented Programming, Systems, Languages and Applications, p. 26-30, 2003.

BRASIL. Lei complementar nº 123, de 14 de dezembro de 2006. República Federativa do Brasil. Institui o Estatuto Nacional da Microempresa e da Empresa de Pequeno Porte. Brasília, 14 de dezembro de 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: agosto/2008.

BRUEGGE, B.; SCHILLER, J. Word Spotting in Scrum Meetings. Database and Expert Systems Application, p. 125-129, 2008.

CARDOSO, C. UML na prática. 1ª edição, Ciência Moderna, 2003.

CARNEVALLI, J.; CAUCHICK MIGUEL, P. Revisão, análise e classificação da literatura sobre o QFD – tipos de pesquisa, dificuldades de uso e benefícios do método. Gest. Prod., São Carlos, v. 14, n. 3, p. 557-579, set.-dez, 2007.

CARVALHO, B. V. MELLO, C. H. P. Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil Scrum. In: SIMPO 2009 - XII Simpósio de Administração da Produção, Logística e Operações Internacionais, São Paulo - SP. 2009.

CARVALHO, M. M.; MACHADO, S. A.; PIZYSIEZNIG FILHO, J.; RABECHINI JR, R. Fatores críticos de sucesso em empresas de base tecnológica. Produto & Produção, vol. 4, número especial, p. 47-59, abril 2000.

CLARK, K. B.; WHEELWRIGHT, S. C. Managing new product and process development: text and cases. New York: The Free Press, 896p, 1993.

COHN, M. Mountain Goat Software. 2008. Disponível em: <http://www.mountaingoatsoftware.com/scrum>. Acesso em: março/2009.

COOPER, R. G.; KLEINSCHMIDT, E. What makes a new product a winner: success factors at project level. The Journal of Product Innovation Management, USA, v. 4, n.3, p.175-189, 1987.

COUGHLAN, P.; COUGHLAN, D. Action Research for operations management International Journal of Operations & Production Management, vol. 22, n. 2, p. 220-240, 2002.

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. Dissertação de mestrado. Universidade Federal de Campina Grande. Centro de Ciências e Tecnologia. Campina Grande, 2003.

DOERNHOEFER, M. Surfing the net for software engineering notes. ACM SIGSOFT

Software Engineering Notes – vol. 29, n. 5; 2004.

DOTPROJECT OPEN SOURCE PROJECT MANAGEMENT TOOL. Disponível em <http://www.dotproject.net/>. Acesso em: 22/05/2009.

DYBÅ, T.; DINGSØYR, T. Empirical Studies of Agile Software Development: A Systematic Review. *Information and Software Technology*, vol. 50, n. 9-10, p. 833-859, 2008.

EDWARDS, M. Overhauling a Failed Project Using Out of the Box Scrum. *Agile Conference*, p. 413-416, 2008.

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, *Proceedings of 4th International Conference of Technology Policy and Innovation*, Curitiba, Brazil, 2000.

FOWLER, M. Put Your Process on a Diet, *Software Development*, December 2000, disponível em <http://www.sdmagazine.com/articles/2000/0012/>. Acesso em: Julho/2003.

FRANCO, M. A. S. Pedagogia da Pesquisa-Ação. *Educação e Pesquisa*, São Paulo, v. 31, n. 3, p. 483-502, 2005.

FRASER, S.; RISING, L.; AMBLER, S.; COCKBURN, A.; ECKSTEIN, J.; HUSSMAN, D.; MILLER, R.; STRIEBECK, M.; THOMAS, D. A fishbowl with piranhas coalescence, convergence or divergence. *Dynamic Languages Symposium*, p. 937-939, 2006.

GREGÓRIO, M.; FALCÃO, R.; VAZ, F.; MARCONI, C.; LINS NETO, R.; FANTINI, W.; ALENCAR, R.; MENEZES, D.; BEZERRA, S. Os Sete Pecados na Aplicação de Processos de Software. *UNIBRATEC – União Brasileira dos institutos de tecnologia*, 2007. Disponível em [http://www.unibratec.com.br/revistacientifica/n2\\_artigos/n2\\_gregorio\\_mla.pdf](http://www.unibratec.com.br/revistacientifica/n2_artigos/n2_gregorio_mla.pdf). Acesso em: janeiro/2009.

GRIFFIN, A. PDMA Research on new product development practices: Updating trends and benchmarking best practices. *Journal of Product Innovation Management*, v.14, n. 6, p. 429-458, 1997.

HENSGEN, P.; PARRA, L.; RUSSAK P.; MCGEE C; FLEURY P.; MADRIGAL G.; BAUMGART T.; WILLIGHAGEN E. *Umbrello UML Modeller Manual*. KDE Docs, 2003.

HIGHSMITH, J. Agile Project Management Creating Innovative Products. *The Journal of Product Innovation Management*, vol. 22, n. 4, p. 371-373, 2005.

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.

JUDY, K.; KRUMINS-BEENS, I. Great Scrums Need Great Product Owners - Unbounded Collaboration and Collective Product Ownership. *Hawaii International Conference on System Sciences*, p. 462-462, 2008.

JOHNSON, J. *The Chaos Report*. The Standish Group International, Inc. West Yarmouth, 1995.

JOHNSON, J. *The Chaos Report*. The Standish Group International, Inc. West Yarmouth, 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.

KEENAN, F. Agile Process Tailoring and problem analysis (APTLY). 26th International Conference on Software, 2004, Estados Unidos.

KNIBERG, H.; FARHANG, R. Bootstrapping Scrum and XP under Crisis. *Agile Conference*, p. 436-444, 2008, Estados Unidos.

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: março/2008.

LASTRES, H. M. M.; CASSIOLATO, J. E.; ARROIO, A. Sistemas de inovação e desenvolvimento: mitos e realidade da economia do conhecimento global. *In*: LASTRES, H. M. M.; CASSIOLATO, J. E.; ARROIO, A. (org.) Conhecimento, Sistemas de Inovação e Desenvolvimento, Rio de Janeiro: Ed. UFRJ, 2005.

LEDMITH, A. Management of new product development in small electronics firms. *Journal of European Industrial Training*, p. 137-148, 2000.

LEWIS, J.; NEHER, K. Over the Waterfall in a Barrel - MSIT Adventures in Scrum. *Agile Conference*, p. 389-394, 2007.

LUKANUSKI, M.; MILANO, M.; BRUIN, J.; ROCHFORD, M.; BOSMAN, R. Agile or awkward surviving and flourishing in an agilesrum project. *Conference on Human Factors in Computing Systems*, 2008.

LYON, R.; EVANS, M. Scaling Up Pushing Scrum out of its Comfort Zone. *Agile Conference*, p. 395-400, 2008.

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, p. 311-327, 2003.

MAHNIC, V.; ZABKAR, N. Measurement repository for Scrum-based software development process. *World Scientific and Engineering Academy and Society*, 2008.

MANIFESTO ÁGIL, 2001. Disponível em: <http://www.manifestoagil.com.br/>. Acesso em: março/2009.

MANN, C.; MAURER, F. A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. *Agile Development Conference*, IEEE Computer Society, p. 70-79, 2005.

MAR, K.; SCHWABER, K. Scrum With XP. 2001. Disponível em <http://www.controlchaos.com/XPKane.htm>. Acesso em : Julho/2003.

MARÇAL, A.; FREITAS, B.; SOARES, F.; BELCHIOR, A. Mapping CMMI Project Management Process Areas to SCRUM Practices. Software Engineering Workshop, p. 13-22, 2007.

MARCHENKO, A.; ABRAHAMSSON, P. Scrum in a Multiproject Environment an Ethnographically-Inspired Case Study on the Adoption Challenges. Agile Conference, p. 15-26, 2008.

MAURER, F.; MELNIK, G. Agile methods moving towards the mainstream of the software industry. 28th International Conference on Software Engineering, 2006.

MAURER, F.; MELNIK, G. Agile Methods: Crossing the Chasm. 29th International Conference on Software Engineering, 2007.

MCGAUGHEY, R. E. Benchmarking business-to-business electronic commerce, Benchmarking. An International Journal, v. 9, n. 5, p. 471-484, 2002.

MINAS GERAIS. Decreto N° 43.442 de 17 de julho de 2003. Estado de Minas Gerais. Altera e consolida a legislação referente ao Programa de Apoio Financeiro ao Desenvolvimento de Médias, Pequenas e Microempresas de Base Tecnológica. Belo Horizonte, 17 de julho de 2003. Disponível em: [http://www.fazenda.mg.gov.br/empresas/legislacao\\_tributaria/decretos/d43442\\_2003.htm](http://www.fazenda.mg.gov.br/empresas/legislacao_tributaria/decretos/d43442_2003.htm). Acesso em: março/2009.

MOORE, R.; REFF, K.; GRAHAM, J.; HACKERSON, B. Scrum at a Fortune 500 Manufacturing Company. Agile Conference, p. 175-180, 2007.

MUNDIM, A. P. F.; ROZENFELD, H.; AMARAL, D. C.; SILVA, S. L.; GUERRERO, V.; HORTA, L. C. 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.

PDMA-Product Development and Management Association. Disponível em:



<http://www.pdma.org>. Acesso em: novembro/2008.

PMI - Project Management Institute. Disponível em: <http://www.pmi.org/>. Acesso em: novembro/2008.

PRESSMAN, R. Software Engineering: A Practitioner's Approach. 6ª ed., Mc Graw Hill, 2005.

PORTER, M. Vantagem Competitiva. Criando e sustentando um desempenho superior. 13 ed., Rio de Janeiro: Campus, 1998.

RAYHAN, S.; HAQUE, N. Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup. Agile Conference, p. 351-355, 2008.

RISING, L.; JANOFF, N. S. The Scrum Software Development Process for Small Teams, IEEE Software, vol. 17, n. 4, 2000.

ROZENFELD, H.; FORCELLINI, F. A.; AMARAL, D. C.; TOLEDO, J. C.; SILVA, S. L.; ALLIPRANDINI, D. H.; SCALICE, R. K. Gestão de desenvolvimento de produtos: uma referência para a melhoria do processo. São Paulo: Saraiva, 2006.

SALO, O.; ABRAHAMSSON, P. Agile methods in European embedded software development organisations. IET Software, vol. 2, n. 1, p. 58-64, 2008.

SANDERS, D. Using Scrum to manage student projects. Journal of Computing Sciences in Colleges, vol. 23, n. 1, 2007.

SCHWABER, K.; BEEDLE, M. Agile Software Development with SCRUM. Prentice Hall, 2002.

SCHWABER, K. SCRUM Development Process. 1995. Disponível em: <http://jeffsutherland.com/oopsla/schwapub.pdf>. Acesso: Julho/2003.

SCHMITZ, H. Global competition and local cooperation: Success and failure in the Sinos Valley, Brazil. World Development, vol. 27, n. 9, p. 1627-1650, 1992.

SCOTLAND, K.; BOUTIN, A. Integrating Scrum with the Process Framework at Yahoo!. Europe Agile Conference, p. 191-195, 2008.

SEBRAE/IPT. MPES de base tecnológica: conceituação, formas de financiamento e análise de casos brasileiros. Relatório de Pesquisa, São Paulo, jul. 2001.

SENGE, P. The Fifth Discipline: the Art and Practice of the Learning Organization. New York: Currency, 1990.

SMITS, H.; PSHIGODA, G. Implementing Scrum in a Distributed Software Development Organization. Agile Conference, p. 371-375, 2007.

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, vol. 15, n.6, p. 520-533, 1998.

SULAIMAN, T.; BARTON, B.; BLACKBURN, T. AgileEVM - Earned Value Management in Scrum Projects. Agile Conference, 2006.

SUTHERLAND, J. Future of Scrum Parallel Pipelining of *Sprints* in Complex Projects. Agile Conference, p. 90-99, 2005.

SUTHERLAND, J.; JAKOBSEN, C.; JOHNSON, K. Scrum and CMMI Level 5. The Magic Potion for Code Warriors. Agile Conference, p. 272-278, 2007.

SUTHERLAND, J.; JAKOBSEN, C.; JOHNSON, K. Scrum and CMMI Level 5. The Magic Potion for Code Warriors. Proceedings of the Hawaii International Conference on System Sciences, p. 466-466, 2008.

SUTHERLAND, J.; SCHOONHEIM, G.; RUSTENBURG, E.; RIJK, M. Fully Distributed Scrum - The Secret Sauce for Hyperproductive Offshored Development Teams. Agile Conference, p. 339-344, 2008.

SUTHERLAND, J.; VIKTOROV, A.; BLOUNT, J.; PUNTIKOV, N. Distributed Scrum. Agile Project Management with Outsourced Development System Sciences.

Proceedings of the Hawaii International Conference on System Sciences, p. 274-285, 2007.

TAKEUCHI, H.; NONAKA, I. The New New Product Development Game, Harvard Business Review, p. 137-146, Jan-Feb. 1986.

TAKEUCHI H.; NONAKA I. Hitotsubashi on Knowledge Management. Singapore: John Wiley & Sons (Asia), 2004.

THIOLLENT, M. Metodologia da pesquisa-ação. 14ª edição, São Paulo: Cortez Editora, 2005.

TOLEDO, J. C.; SILVA, S. L.; MENDES, G. H. S.; JUGEND, D. A gestão do processo de desenvolvimento de produto em empresas de base tecnológica de pequeno e médio portes. Relatório de Pesquisa FAPESP, GEPEQ/DEP/UFSCar, São Carlos, 2005.

TOLEDO, J. C.; SILVA, S. L.; MENDES, G. H. S.; JUGEND, D. Fatores críticos de sucesso no gerenciamento de projetos de desenvolvimento de produto em empresas de base tecnológica de pequeno e médio porte. Gest. Prod., vol.15, no.1, 2008.

TRAC OPEN SOURCE PROJECT. Disponível em: <http://trac.edgewall.org/>. Acesso em: maio/2009.

UY, E.; IOANNOU, N. Growing and Sustaining an Offshore Scrum Engagement. Agile Conference, p. 345-350, 2008.

UY, E.; ROSENDAHL, R. Migrating from SharePoint to a Better Scrum Tool. Agile Conference, p. 506-512, 2008.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software. 7ª edição, Editora Érica, 2007.

VRIENS, C. Certifying for CMM Level 2 and ISO 9001 with XP & Scrum. Agile Development Conference, p. 120-124, 2003.

## ANEXO A – APRESENTAÇÃO DOS ARTIGOS CODIFICADOS

Tabela 46 - Apresentação dos artigos codificados

Código	Autores	Título do Artigo
1	MANN, C. & MAURER, F.	A case study on the impact of scrum on overtime and customer satisfaction
2	FRASER, S.; RISING, L.; AMBLER, S.; COCKBURN, A.; ECKSTEIN, J.; HUSSMAN, D.; MILLER, R.; STRIEBECK, M.; THOMAS, D.	A fishbowl with piranhas coalescence, convergence or divergence
3	SALO, O.; ABRAHAMSSON, P.	Agile methods in European embedded software development organisations
4	MAURER, F.; MELNIK, G.	Agile methods moving towards the mainstream of the software industry
5	MAURER, F.; MELNIK, G.	Agile Methods: Crossing the Chasm
6	LUKANUSKI, M.; MILANO, M.; BRUIN, J.; ROCHFORD, M.; BOSMAN, R.	Agile or awkward surviving and flourishing in an agilescrum project
7	KEENAN, F.	Agile Process Tailoring and probLem anaLYsis (APTLY)
8	HIGHSMITH, J.	Agile Project Management Creating Innovative Products
9	SULAIMAN, T.; BARTON, B.; BLACKBURN, T.	AgileEVM - Earned Value Management in Scrum Projects
10	BERCZUK, S.	Back to basics: The Role of Agile Principles in Success with an Distributed Scrum Team
11	KNIBERG, H.; FARHANG, R.	Bootstrapping Scrum and XP under Crisis
12	VRIENS, C.	Certifying for CMM Level 2 and ISO9001 with XP& Scrum
13	PAASIVAARA, M.; DURASIEWICZ, S.; LASSENIUS, C.	Distributed Agile Development: Using Scrum in a Large Project
14	SUTHERLAND, J.; VIKTOROV, A.; BLOUNT, J.; PUNTIKOV, N.	Distributed Scrum Agile Project Management with Outsourced Development
15	DYBÅ, T. & DINGSØYR, T.	Empirical studies of agile software development A systematic review
16	SUTHERLAND, J.; SCHOONHEIM, G.; RUSTENBURG, E.; RIJK, M.	Fully Distributed Scrum The Secret Sauce for Hyperproductive Offshored Development Teams
17	SUTHERLAND, J.	Future of Scrum Parallel Pipelining of <i>Sprints</i> in Complex Projects
18	JUDY, K.; KRUMINS-BEENS, I.	Great Scrums Need Great Product Owners Unbounded Collaboration and Collective Product Ownership
19	UY, E.; IOANNOU, N.	Growing and Sustaining an Offshore Scrum Engagement
20	BARTON, B.; CAMPBELL, E.	Implementing a Professional Services Organization Using Type C Scrum

21	SMITS, H.; PSHIGODA, G.	Implementing Scrum in a Distributed Software Development Organization
22	RAYHAN, S.; HAQUE, N.	Incremental Adoption of Scrum for Successful Delivery of an IT Project in a Remote Setup
23	SCOTLAND, K.; BOUTIN, A.	Integrating Scrum with the Process Framework at Yahoo! Europe
24	MARÇAL, A.; FREITAS, B.; SOARES, F.; BELCHIOR, A.	Mapping CMMI Project Management Process Areas to SCRUM Practices
25	MAHNIC, V.; ZABKAR, N.	Measurement repository for Scrum-based software development process
26	UY, E.; ROSENDAHL, R.	Migrating from SharePoint to a Better Scrum Tool
27	LEWIS, J.; NEHER, K.	Over the Waterfall in a Barrel - MSIT Adventures in Scrum
28	EDWARDS, M.	Overhauling a Failed Project Using Out of the Box Scrum
29	LYON, R.; EVANS, M.	Scaling Up Pushing Scrum out of its Comfort Zone
30	SUTHERLAND, J.; JAKOBSEN, C.; JOHNSON, K.	Scrum and CMMI Level 5 The Magic Potion for Code Warriors
31	SUTHERLAND, J.; JAKOBSEN, C.; JOHNSON, K.	Scrum and CMMI Level 5 The Magic Potion for Code Warriors
32	MOORE, R.; REFF, K.; GRAHAM, J.; HACKERSON, B.	Scrum at a Fortune 500 Manufacturing Company
33	BATES, C.; YATES, S.	Scrum down a software engineer and a sociologist explore the implementation of an agile method
34	MARCHENKO, A.; ABRAHAMSSON, P.	Scrum in a Multiproject Environment An Ethnographically-Inspired Case Study on the Adoption Challenges
35	DOERNHOEFER, M.	Surfing the net for software engineering notes
36	RISING, L.; JANOFF, N.	The Scrum software development process for small teams
37	CRISTAL, M.; WILDT, D.; PRIKLADNICKI, R.	Usage of SCRUM Practices within a Global Company
38	SANDERS, D.	Using Scrum to manage student projects
39	BERCZUK, S.; HARRISON, N.; HENNEY, K.; KERIEVSKY, J.; RISING, L.; SCHWABER, K.; WOOLF, B.	What's so eXtreme about doing things right
40	BRUEGGE, B.; SCHILLER, J.	Word Spotting in Scrum Meetings

Tabela 47 - Apresentação dos periódicos, anos de publicação, base da dados e métodos dos artigos codificados

Código	Journal	Ano	Base da dados	Tipo de Estudo
1	Agile Conference	2005	IEEE	Estudo de Caso
2	Dynamic Languages Symposium	2006	ACM	Teórico-conceitual

3	IET Software	2008	AIP	Survey
4	28th International Conference on Software Engineering.	2006	ACM	Teórico-conceitual
5	29th International Conference on Software Engineering.	2007	ACM	Teórico-conceitual
6	Conference on Human Factors in Computing Systems.	2008	ACM	Teórico-conceitual
7	26th International Conference on Software.	2004	ACM	Teórico-conceitual
8	The Journal of Product Innovation Management	2005	ACM	Revisão de Literatura
9	Agile Conference.	2006	IEEE	Experimental
10	Agile Conference	2007	IEEE	Estudo de Caso
11	Agile Conference	2008	IEEE	Estudo de Caso
12	Agile Development Conference	2003	IEEE	Estudo de Caso
13	Global Software Engineering	2008	IEEE	Estudo de Caso
14	System Sciences	2007	IEEE	Estudo de Caso
15	SINTEF ICT, S.P. Andersensv.	2008	Science Direct	Revisão de Literatura
16	Agile Conference	2008	IEEE	Estudo de Caso
17	Agile Conference	2005	IEEE	Teórico-conceitual
18	Hawaii International Conference on System Sciences	2008	IEEE	Teórico-conceitual
19	Agile Conference	2008	IEEE	Estudo de Caso
20	System Sciences	2007	IEEE	Teórico-conceitual
21	Agile Conference	2007	IEEE	Estudo de Caso
22	Agile Conference	2008	IEEE	Estudo de Caso
23	Agile Conference	2008	IEEE	Estudo de Caso
24	Software Engineering Workshop	2007	IEEE	Teórico-conceitual
25	World Scientific and Engineering Academy and Society.	2008	ACM	Teórico-conceitual
26	Agile Conference	2008	IEEE	Estudo de Caso
27	Agile Conference	2007	IEEE	Estudo de Caso
28	Agile Conference	2008	IEEE	Pesquisa-ação
29	Agile Conference	2008	IEEE	Estudo de Caso
30	Agile Conference	2007	IEEE	Teórico-conceitual
31	Hawaii International Conference on System Sciences	2008	IEEE	Teórico-conceitual
32	Agile Conference	2007	IEEE	Pesquisa-ação
33	International workshop on Cooperative and human aspects of software engineering.	2008	ACM	Teórico-conceitual
34	Agile Conference	2008	IEEE	Estudo de Caso
35	ACM SIGSOFT Software Engineering Notes	2004	ACM	Revisão de Literatura
36	Software, IEEE	2000	IEEE	Teórico-conceitual

37	Global Software Engineering	2008	IEEE	Estudo de Caso
38	Journal of Computing Sciences in Colleges	2007	ACM	Teórico-conceitual
39	International Conference on Object-Oriented Programming, Systems, Languages, and Applications	2003	ACM	Teórico-conceitual
40	Database and Expert Systems Application	2008	IEEE	Teórico-conceitual

Tabela 48 - Apresentação da abordagem, filiação do pesquisador e período de análise dos artigos codificados e os benefícios do *Scrum* neles citados

Código	Abordagem	Filiação do autor	Período de análise	Benefícios do <i>Scrum</i> Citados								
				A	B	C	D	E	F	G	H	I
1	Qualitativa	Univers	Longitudinal	x								
2	Qualitativa	Empresa	Atual									
3	Quantitativa	CP	Atual	x								
4	Qualitativa	Univers	Atual									
5	Qualitativa	Univers	Atual	x	x							
6	Qualitativa	Empresa	Atual									
7	Qualitativa	Univers	Atual									
8	Qualitativa	Empresa	Atual									
9	Quantitativa	Empresa	Atual			x						
10	Qualitativa	Empresa	Atual		x							
11	Qualitativa	Empresa	Atual				x					
12	Qualitativa	Empresa	Atual									
13	Qualitativa	Univers	Atual		x		x	x				
14	Qualitativa	Empresa	Atual						x			
15	Qualitativa	CP	Atual									
16	Qualitativa	Empresa	Atual					x		x	x	
17	Qualitativa	Empresa	Atual	x	x	x	x	x	x	x		
18	Qualitativa	Empresa	Atual		x							
19	Qualitativa	Empresa	Atual									
20	Qualitativa	Empresa	Atual	x	x	x	x	x		x		
21	Qualitativa	Empresa	Atual		x			x				
22	Qualitativa	Empresa	Atual		x							
23	Qualitativa	Empresa	Atual									
24	Qualitativa	Univers	Atual		x	x		x		x		
25	Qualitativa	Empresa	Atual									
26	Qualitativa	Empresa	Atual									

27	Qualitativa	Empresa	Atual									
28	Qualitativa	Empresa	Atual									x
29	Qualitativa	Empresa	Atual				x					
30	Qualitativa	Empresa	Atual	x		x		x		x		
31	Qualitativa	Empresa	Atual	x		x		x		x		
32	Qualitativa	Empresa	Atual		x		x			x		
33	Qualitativa	Univers	Atual	x	x							
34	Qualitativa	CP	Atual									
35	Qualitativa	Empresa	Atual									
36	Qualitativa	Empresa	Atual	x	x			x				
37	Qualitativa	Univers	Atual					x		x		
38	Qualitativa	Univers	Atual		x					x	x	
39	Qualitativa	Empresa	Atual									
40	Qualitativa	Univers	Atual						x	x	x	

Onde Univers = Universidade e CP = Centro de Pesquisa. E onde: A: Aumento da satisfação de clientes (diminuição de reclamações); B: Melhoria na comunicação e aumento da colaboração entre envolvidos. C: Aumento do retorno do investimento do projeto. D: Aumento da motivação da equipe de desenvolvimento. E: Melhoria da qualidade do produto produzido. F: Diminuição dos custos de produção (mão-de-obra). G: Aumento de produtividade da equipe. H: Diminuição no tempo gasto para terminar o projeto (prazo). I: Diminuição do risco do projeto (menor possibilidade de insucesso).