

**C.E.S.A.R – CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS
DO RECIFE**

ERIC DE OLIVEIRA CAVALCANTI

**FIRESCRUM: FERRAMENTA DE APOIO À GESTÃO ÁGIL DE
PROJETOS UTILIZANDO SCRUM**

RECIFE

2009

ERIC DE OLIVEIRA CAVALCANTI

**FIRESCRUM: FERRAMENTA DE APOIO À GESTÃO DE PROJETOS
UTILIZANDO SCRUM**

Dissertação apresentada ao programa de Mestrado em Engenharia de Software do Centro de Estudos e Sistemas Educacionais do Recife – C.E.S.A.R, como requisito para a obtenção do título de Mestre em Engenharia de Software.

Orientação: Prof. Dr Jones Albuquerque

Co-Orientação: Profa. Msc Teresa Maciel

RECIFE

2009

**C.E.S.A.R – CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS
DO RECIFE**

**Título da dissertação
ERIC DE OLIVEIRA CAVALCANTI**

Dissertação apresentada ao programa de Mestrado em Engenharia de Software do Centro de Estudos e Sistemas Educacionais do Recife- C.E.S.A.R., como requisito para a obtenção do título de Mestre em Engenharia de Software.

Data de aprovação:
____ / ____ / 2009.

Banca examinadora:

Prof.(a).Dr.(a) ***
instituição

Prof.(a).Dr.(a) ***
instituição

Prof.(a).Dr.(a)
instituição

Dedicatória

Dedico esta dissertação primeiramente a Deus, pois sem Ele, nada seria possível e não estaríamos aqui reunidos, desfrutando, juntos, destes momentos tão importantes.

A minha esposa Sheyla Cavalcanti, pelo incentivo, carinho, cooperação e apoio.

Ao meu filho Artur Cavalcanti, presente dado por Deus durante a construção desta obra, dando um novo sentido para minha vida, e aumentando a motivação para concluir este trabalho.

Agradecimentos

A minha esposa Sheyla Cavalcanti, pelo carinho e incentivo.

Aos meus familiares, pelo apoio.

Ao meu orientador, Prof. Dr Jones Albuquerque por ser um grande mestre, incentivador e amigo. Por todo seu apoio e entusiasmo prestados neste trabalho.

A minha Co-Orientadora, Teresa Maciel, por todo seu apoio prestado. Pelo convite em apresentar o fruto deste trabalho em um evento que trouxe uma visibilidade sem igual ao mesmo.

Ao Prof. Dr Silvio Meira, por acreditar neste trabalho e pelo apoio prestado.

A Aísa Pereira, por acreditar e incentivar este trabalho.

A todos que colaboraram e colaboraram no desenvolvido do FireScrum.

A todos os educadores e profissionais do CESAR.EDU pelo aprendizado, amizade e incentivo.

Epígrafe

"O futuro têm muitos nomes. Para os incapazes o inalcançável, para os medrosos o desconhecido, para os valentes a oportunidade."

Victor Hugo

SUMÁRIO

1	INTRODUÇÃO	15
1.1	TEMA	15
1.2	MOTIVAÇÃO.....	17
1.2.1	Motivação de Mercado	17
1.2.2	Motivação Técnica	21
1.3	PROBLEMA	22
1.3.1	Objetivo geral	24
1.3.2	Objetivos específicos	24
1.4	CONTRIBUIÇÕES	24
1.5	ESTRUTURA DA DISSERTAÇÃO.....	25
2	ESTADO DA ARTE	26
2.1	METODOLOGIAS ÁGEIS	26
2.2	SCRUM	26
2.2.1	Termos e conceitos	27
2.2.2	Papéis e Responsabilidades	28
2.2.3	Ciclo de Vida	28
2.3	RICH INTERNET APPLICATION.....	32
2.3.1	AJAX	33
2.3.2	Adobe Flash	34
2.3.3	Adobe Flex	34
2.3.4	Adobe Integrated Runtime	34
2.3.5	Laszlo	34
2.3.6	Microsoft Windows Presentation Foundation	35
2.3.7	Microsoft Silverlight	35
2.3.8	JavaFx.....	35
3	SOLUÇÃO PROPOSTA – FIRESCRUM	36
3.1	CARACTERÍSTICAS GERAIS	36
3.2	FUNCIONALIDADES	37
3.3	ARQUITETURA	51
3.3.1	Arquitetura do <i>front-end</i>	53
3.3.2	Arquitetura do <i>back-end</i>	58
3.4	REQUISITOS TECNOLÓGICOS MÍNIMOS PARA UTILIZAÇÃO DA FERRAMENTA	59
4	METODOLOGIA	60
4.1	ANÁLISE DE FERRAMENTAS	61
4.2	PROTÓTIPOS E VALIDAÇÕES COM USUÁRIOS	62

4.3	PARTICIPAÇÃO DE LISTAS DE DISCUSSÕES.....	64
4.4	SITE DA FERRAMENTA.....	64
5	CONSIDERAÇÕES FINAIS.....	66
5.1	EVOLUÇÃO DO TRABALHO.....	67
	REFERÊNCIAS.....	68
	ANEXO A – E-MAILS SOBRE O FIRESCRUM.....	72

Resumo

As metodologias ágeis surgiram no final da década de 90 como uma alternativa aos métodos tradicionais de desenvolvimento de software. Tais metodologias propõem uma nova abordagem para o desenvolvimento, eliminando gastos com documentação excessiva e burocrática, enfatizando a comunicação, colaboração com o cliente e as atividades que trazem valor imediato na produção de software com qualidade.

Com o avanço da adoção das metodologias ágeis no desenvolvimento de software mundial, a procura por ferramentas de apoio a esta nova forma de desenvolver e gerenciar software surge com o objetivo de aumentar o potencial competitivo das empresas, otimizar o processo e viabilizar uma necessidade do cenário globalizado atual que é o desenvolvimento distribuído de software.

Sendo o Scrum apontado pelas pesquisas como a metodologia ágil que vem sendo a mais adotada pelas empresas, este trabalho apresenta uma proposta de ferramenta de apoio à gestão de projetos que utilizam o Scrum, onde alinhada com os princípios ágeis, propõem-se em não burocratizar o processo, tendo como foco principal a simplicidade e usabilidade.

Palavras-chave

Metodologias Ágeis. Scrum. Gestão de Projetos. RIA. Desenvolvimento de Software Distribuído. Ferramenta.

Abstract

Agile methodologies appeared at the end of 90s as an alternative to traditional methods of software development. These methodologies propose a new approach to development, eliminating excessive expenses with bureaucratic documentation, emphasizing the communication, collaboration with the customer, and the activities that bring immediate value to the production of high quality software.

With the advanced adoption of agile methodologies in the software development world, there is a high demand for tools to support this new way of developing and managing software. These tools increase the competitive potential of companies, optimize their process, and make possible a current need of the global scene, that it is the distributed development of software.

As Scrum is currently pointed by researches as the most adopted agile methodology among companies, this work proposes a tool to manage projects using Scrum, which is aligned with the agile principles, intends to avoid bureaucratizing the process, and focuses on simplicity and usability.

Key-words

Agile methodologies. Scrum. Project Management. RIA. Distributed Software Development. Tool.

Lista de ilustrações

Figura 1. Aumentos específicos maior ou igual a 10% por implementar a abordagem Ágil.....	18
Figura 2. Quais os percentuais de projetos ágeis obtiveram sucesso na perspectiva das organizações.....	19
Figura 3. Qual a metodologia ágil as empresas estão seguindo.....	19
Figura 4. Número de visitas ao site oficial do FireScrum.	21
Figura 5. Exemplos de artefatos e quadros físicos.....	23
Figura 6. Visão geral do processo do Scrum.....	29
Figura 7. Exemplo de <i>Burndown</i> do Produto exibindo o consumo de <i>story points</i> de cada Sprint.....	31
Figura 8. Exemplo de <i>Burndown</i> da <i>Sprint</i> exibindo o consumo de horas das tarefas durante a <i>Sprint</i>	32
Figura 9. Estrutura modular do FireScrum	38
Figura 10. Tela principal do módulo <i>Core</i> do FireScrum.	39
Figura 11. Imagem do módulo <i>TaskBoard</i> do FireScrum.....	40
Figura 12. Imagem do módulo <i>Planning Poker</i> do FireScrum.	40
Figura 13. Imagem do módulo <i>Test Management</i> do FireScrum.	41
Figura 14. Imagem do módulo <i>Bug Tracking</i> do FireScrum.	42
Figura 15. Imagem do módulo <i>Desktop Agent</i> do FireScrum.....	42
Figura 16. Tela de cadastro de Produto.	43
Figura 17. Tela de seleção do Produto.	44
Figura 18. Tela de cadastro do Item de <i>Backlog</i>	45
Figura 19. <i>Backlog</i> do Produto (<i>Product Backlog</i>) na tela principal.....	45
Figura 20. Priorizando os Itens de <i>Backlog</i> do Produto.	46
Figura 21. Tela de cadastro de <i>Sprint</i>	47
Figura 22. Definindo o Selected Backlog.	47
Figura 23. Tela do Cadastro de Tarefa	48
Figura 24. <i>Sprint Backlog</i>	49
Figura 25. <i>Mockup</i> da tela do <i>Sprint Burndown</i>	50
Figura 26. <i>Mockup</i> da tela do <i>Product Burndown</i>	50
Figura 27. <i>Mockup</i> da tela de Impedimentos.....	51
Figura 28. Arquitetura macro.....	53
Figura 29. Percentual de penetração do Adobe Flash Player.	54

Figura 30. Mapeando funcionalidades dos papéis para camadas do <i>Cairngorm</i>	55
Figura 31. Elementos do <i>Cairngorm</i> e fluxo de eventos.....	57
Figura 32. Diagrama da arquitetura do <i>back-end</i>	59
Figura 33. Protótipos em papel	63
Figura 34. Validação com usuários	63
Figura 35. Site oficial do FireScrum.	65
Figura 36. Informações de acesso providas pelo Google Analytics.	65

Lista de tabelas

Tabela 1. Metodologia - Atividades e Objetivos	60
Tabela 2. Análise comparativa de funcionalidades.	61

ABREVIATURAS

Sigla	Significado
XP	Extreme Programming
RIA	Rich Internet Applications
ROI	Return of Investment
FDD	Feature Driven Development
TI	Tecnologia da Informação
RPC	Remote Procedure Call
HTML	HyperText Markup Language
DHTML	Dynamic HTML
XML	Extensible Markup Language
AJAX	Asynchronous Javascript And XML
AIR	Adobe Integrated Runtime
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
WPF/E	Windows Presentation Foundation/Everywhere
API	Application Program Interface
BLI	Backlog Item
AMF	Action Message Format
MVC	Model-View-Controller
GPL	General Public License

1 INTRODUÇÃO

1.1 TEMA

À medida que as organizações tornam-se cada vez mais dependentes da indústria do software, ficam mais evidentes os problemas relacionados ao processo de desenvolvimento de sistemas: alto custo, alta complexidade, dificuldade de manutenção, e uma disparidade entre as necessidades dos usuários e o produto desenvolvido (SOMMERVILLE 2003).

Durante a evolução dos processos de Engenharia de Software, a indústria se baseou nos métodos tradicionais de Engenharia de Software, que definiram por muitos anos os padrões para a criação de software no meio acadêmico e empresarial. Acreditando que o processo aplicado pelo modelo tradicional seja um dos motivos para a ocorrência dos fracassos dos projetos de software que aconteceram ao longo da história (STANDISH 1994, 2004), um segmento crescente da Engenharia de Software vem defendendo a adoção de processos mais simplificados conhecidos como metodologias ágeis ou métodos ágeis, que visam à desburocratização das atividades associadas ao desenvolvimento de software e a valorização de outros princípios. Segundo Fowler, metodologias ágeis são orientadas a pessoas ao invés de serem orientadas a processos. Ele afirma que o objetivo dos métodos de engenharia tradicional é de definir um processo que irá funcionar bem, independentemente de quem os estiverem utilizando. Entretanto, segundo ele, metodologias ágeis afirmam que nenhum processo jamais será equivalente à habilidade da equipe de desenvolvimento e, portanto, o papel do processo é dar suporte à equipe de desenvolvimento e seu trabalho (FOWLER 2001).

O processo de desenvolvimento de software ágil foi influenciado pelas melhores práticas da indústria Japonesa, particularmente pelos princípios de desenvolvimento *Lean* (POPPENDIECK 2005) implementados em empresas como a Toyota (LIKER 2004) e a Honda (HOLFORD e EBRAHIMI 2007).

No final da década de 90 surgiu uma série de abordagens ágeis como *Adaptive Software Development*, *Crystal*, *Dynamic Systems Development*, *Extreme Programming* (XP), *Feature Driven Development*, e *Scrum* (BOEHM 2006). Na contramão do grande fluxo da indústria de software as metodologias ágeis concentram-se nos fatores humanos e em entregar valor ao cliente, e não em processos. Larman sugere que, ao contrário do cenário numa linha de produção em massa, o software não é algo previsível ou imune a mudanças e que desenvolver software é como desenvolver novos produtos (LARMAN 2003). Poppenick ressalta que desenvolver é como criar uma receita, enquanto produzir é seguir os passos de uma receita (POPPENDIECK 2003). Larman enfatiza que o desenvolvimento é um processo de aprendizado, que envolve tentativas e erros, e que, como a manufatura previsível, não pode ser comparada ao software, e dificilmente as práticas e valores enraizados nesse paradigma trazem algum benefício (LARMAN 2003).

Com o objetivo principal de desburocratizar o processo, não criar uma sobrecarga de trabalho nas informações necessárias para o acompanhamento e aumentar a colaboração e visibilidade do time, as metodologias ágeis sugerem a utilização de artefatos simples, muitas vezes construídos através de planilhas e quadro de tarefas baseados em cartões, murais, cartolinhas e pôsteres (MOUNTAIN 2009).

Porém, conforme avaliado por Cockburn, ferramentas devem ser consideradas como pontos importantes para a obtenção de resultados rápidos na construção do software, onde a colaboração não deve acontecer apenas dentro do time de desenvolvimento, mas também por toda a organização, com usuários experientes e patrocinadores do projeto (COCKBURN 2004).

Um desafio a ser considerado é a dificuldade de se trabalhar com metodologias ágeis usando um ambiente de times remotos, onde a presença física e reuniões face a face que são uma das características dessas metodologias não são possíveis, e muito menos a visualização dos quadros de tarefas. Outro ponto importante é que sem o uso de uma ferramenta o processo de consolidação de métricas e histórico pode se tornar bastante custoso.

Alinhado aos princípios ágeis, o *Scrum* foi criado em 1996 por Ken Schwaber e Jeff Sutherland, como um método que aceita que o desenvolvimento de software é imprevisível e formaliza a abstração, sendo aplicável a ambientes voláteis. O *Scrum* se destaca dos demais métodos ágeis pela ênfase dada ao gerenciamento do projeto. Há atividades de monitoramento e *feedback*, em geral, através de reuniões rápidas e diárias com toda a equipe, visando identificação e correção de deficiências e/ou impedimentos no desenvolvimento (SCHWABER e BEEDLE 2002).

Nesse contexto, esta Dissertação propõe a definição, especificação e o desenvolvimento do FireScrum, uma ferramenta *open source* definida para suportar as atividades do *Scrum* de modo automático com o objetivo de estimular sua adoção, como também, proporcionar seu uso por equipes locais e remotas. Adicionalmente, o uso da ferramenta evita a perda de histórico, propiciando o levantamento de métricas sem demandar esforço excessivo do time.

1.2 MOTIVAÇÃO

1.2.1 Motivação de Mercado

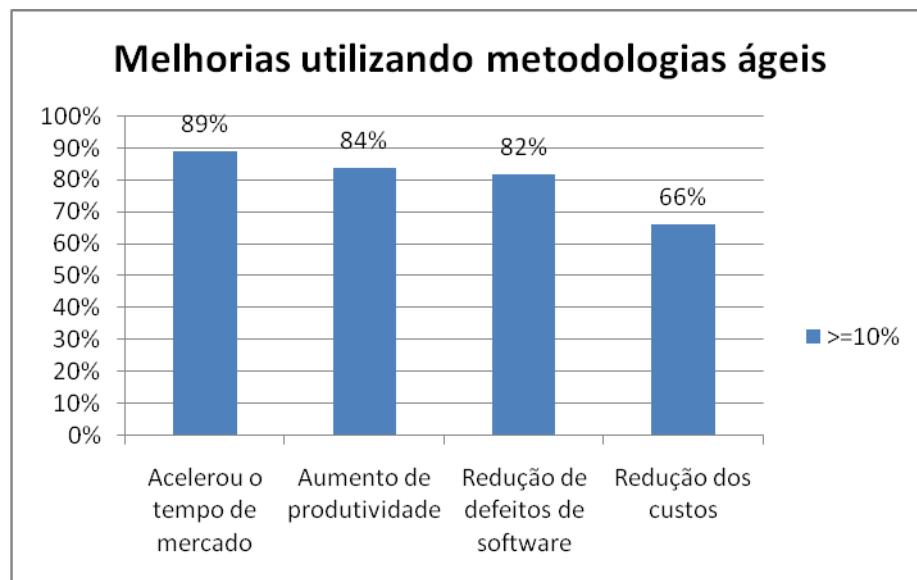
No CHAOS *Report* de 1994 (STANDISH 1994), 365 gerentes executivos de TI representando pequenas, médias e grandes empresas dos maiores segmentos da indústria colaboraram com a avaliação de 8.380 projetos. Dentre estes projetos apenas 16% obtiveram sucesso, 31,1% foram cancelados, 52,7% custaram mais que 189% de sua estimativa inicial. Os participantes também declararam as principais razões para esses resultados e apenas seis fatores representam juntos 57,8% das possíveis causas: *requisitos incompletos*, *falta de envolvimento de usuários*, *mudanças de requisitos e especificações*, *falta de apoio de negócios*, *falta de recursos* (STANDISH 1994).

Versões mais recentes do relatório mostram uma tendência de melhora na qualidade dos projetos, porém a situação ainda requer atenção, pois os dados de 2004 indicam que só 29% dos projetos são finalizados com sucesso e o percentual de projetos que excedem os custos ou o prazo, mantém-se tão elevado quanto antes (STANDISH 2004).

Diante deste quadro, as metodologias ágeis indicam que, diferentemente das atividades de engenharia Civil ou Mecânica, onde as metodologias tradicionais foram inspiradas, o processo de desenvolvimento de software na grande maioria das vezes é imprevisível. Em vez da abordagem tradicional onde as mudanças são evitadas, uma vez que não estavam contempladas no plano inicial, e as pessoas podem não saber o que fazer quando este plano falha, as metodologias ágeis consideram mudanças como bem-vindas e utilizam a abordagem adaptativa no desenvolvimento de software (FOWLER 2001).

Em uma pesquisa realizada pela VersionOne e divulgada em Agosto de 2008 (VERSIONONE 2008), onde receberam mais de 2.300 respostas com um alcance global correspondente a 80 países, foi solicitado aos participantes informarem a estimativa de aumento dos benefícios igual ou maior a 10% que foram adquiridos na utilização de metodologias ágeis conforme ilustra a Figura 1.

Figura 1. Aumentos específicos maior ou igual a 10% por implementar a abordagem Ágil.

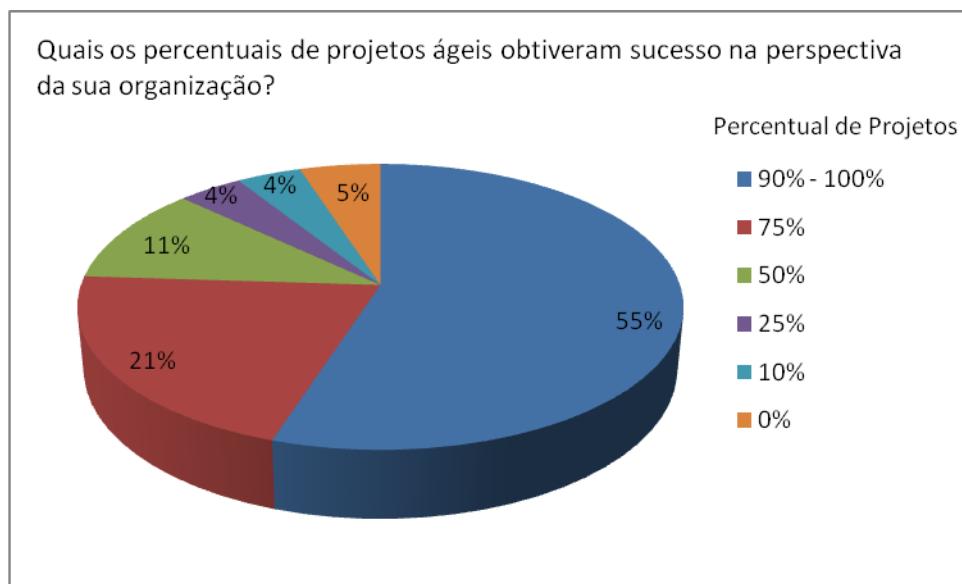


Fonte: (VERSIONONE 2008)

Como observado na Figura 1, 89% informaram que obtiveram um aumento no tempo de mercado, 84% informaram que obtiveram um aumento de produtividade, 82% obtiveram uma redução nos defeitos de software e 66% obtiveram uma redução dos custos. Na pergunta sobre “Quais os percentuais de projetos ágeis obtiveram

sucesso na perspectiva da sua organização?", podemos observar que 55% obtiveram sucesso entre 90% e 100% dos projetos conforme apresentado na Figura 2.

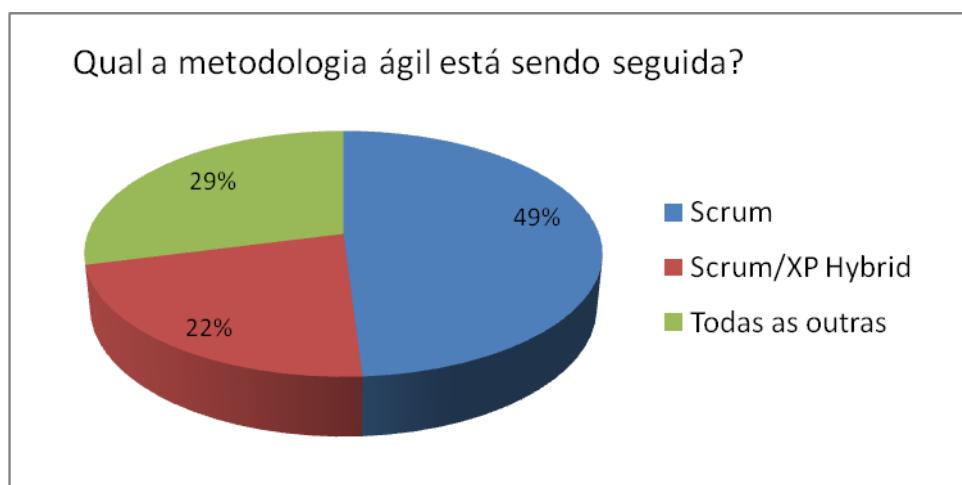
Figura 2. Quais os percentuais de projetos ágeis obtiveram sucesso na perspectiva das organizações.



Fonte: (VERSIONONE 2008)

Na mesma pesquisa quando perguntado “Qual a metodologia ágil está sendo seguida?”. O *Scrum* encontra-se em uma posição de destaque conforme apresenta a Figura 3.

Figura 3. Qual a metodologia ágil as empresas estão seguindo.



Fonte: (VERSIONONE 2008)

No cenário globalizado atual a necessidade em desenvolver software de forma distribuída é cada vez mais evidente, tendo como principal objetivo redução de custos e contratação de mão-de-obra especializada. Segundo Scott W. Ambler, uma pesquisa realizada pelo site Dr.Dobb's em fevereiro de 2008 aponta que um pequeno número de empresas já utiliza metodologias ágeis de forma bem sucedida com equipes de mais de 200 pessoas de forma distribuída, e muitas outras estão tentando com equipes de 50 ou mais pessoas (AMBLER 2008).

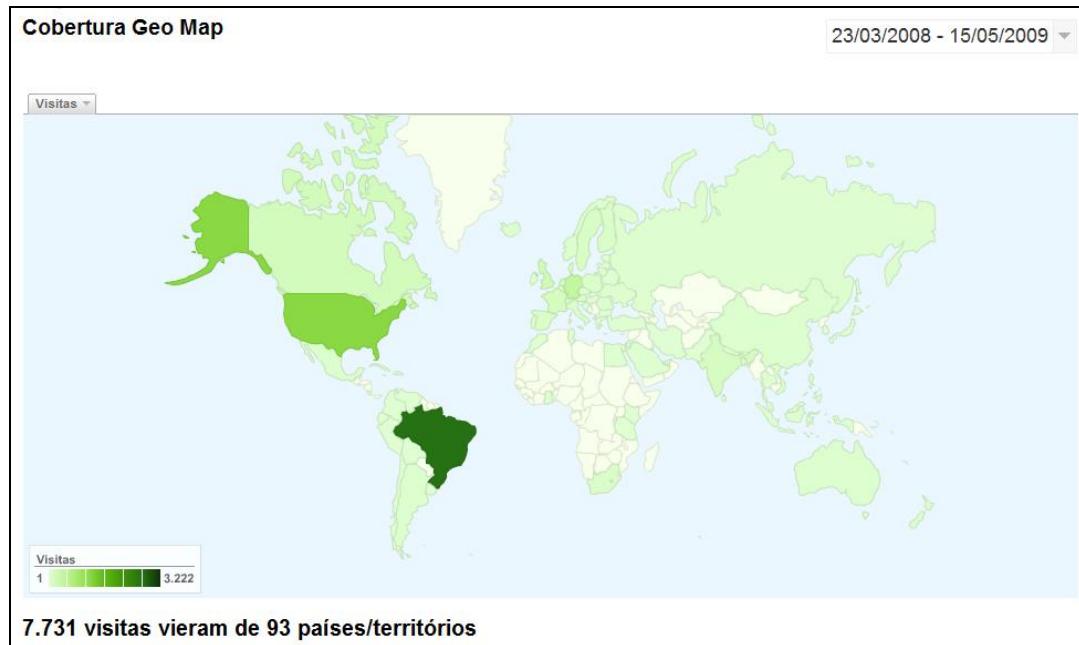
Com o crescimento do uso das metodologias ágeis em todo o mundo é natural o surgimento de ferramentas de apoio ao processo, com o objetivo de aumentar a produtividade, consolidar informações, realizar a manutenção de histórico e proporcionar uma visibilidade para todos os envolvidos no projeto.

Martin Fowler identifica um ponto muito importante, onde ferramentas que impõe uma estrutura simplificada são preferíveis, uma vez que elas se encaixam melhor no que a equipe deseja fazer (FOWLER 2001). Desta forma, produzir uma interface com usuário eficiente, e que se aproxime ao máximo da simplicidade na utilização de artefatos simples como quadros e cartões é essencial para tornar a ferramenta alinhada com os princípios ágeis.

Diante deste cenário, durante o mestrado profissional do C.E.S.A.R, uma equipe de alunos decidiu iniciar o desenvolvimento de uma ferramenta de apoio à gestão de projetos utilizando *Scrum*, nomeada de *FireScrum* (FIRE 2007). Posteriormente em uma disciplina de pós-graduação do Centro de Informática da UFPE (Cin), o Firescrum representou o estudo de caso de uma equipe de alunos que concentraram no desenvolvimento do módulo *Taskboard*.

Em 23 de abril de 2008 a proposta da ferramenta FireScrum foi apresentada à comunidade durante o *Workshop – Uso do SCRUM em ambientes CMMI* promovido pelo *Software Process Improvement Network Recife* (SPIN-RECIFE 2008), sendo anunciado o lançamento oficial do site da ferramenta. Desde seu lançamento até 15 de maio de 2009 o site recebeu 7.731 visitas vindas de 93 países/territórios como ilustra a Figura 4.

Figura 4. Número de visitas ao site oficial do FireScrum.



Fonte: (ANALYTICS 2009)

Desta forma é possível perceber o nível de interesse mundial pelo FireScrum, que também foi alavancado por comentários em blogs de membros ativos da comunidade ágil e que estavam presentes no workshop como Camilo Bardusco da Globo.com (BARDUSCO 2008) e Boris Gloger da Sprint-IT, especialista em Scrum (GLOGER 2008, 2009). Fatos que corroboram a expectativa e o nível de interesse por este trabalho.

1.2.2 Motivação Técnica

Uma vez que alguns dos principais objetivos das metodologias ágeis são de diminuir a burocracia e aumentar a comunicação entre o time - onde muitas vezes são incentivados a utilizar artefatos simples como murais de cartões - a proposta de uma ferramenta de apoio precisa ter como foco principal a facilidade de uso, disponibilidade, usabilidade, elevando a experiência do usuário.

Segundo Carl Zetie, uma das mais longas tensões entre negócios e TI é representada pelo tradicional impasse entre clientes ricos, com interfaces poderosas, e clientes magros, com interfaces simples. Negócios preferem clientes ricos porque

os usuários ganham interatividade, produtividade, e satisfação do usuário, mas os custos em desenvolvimento, gerenciamento, distribuição e suporte ao usuário recaem largamente em TI (ZETIE 2006).

Diante deste cenário, estudos, tecnologias específicas e arquiteturas bem definidas devem ser realizadas e utilizadas para o suprimento destes desafios.

Apesar da existência no mercado de ferramentas que suportem o *Scrum*, a falta de ferramentas *free* ou *open source* com funcionalidades importantes e a preocupação com usabilidade e simplicidade, ou até mesmo foco no *Scrum*, é evidente.

O propósito do FireScrum é fornecer as funcionalidades existentes em ferramentas comerciais e de grande aceitação pelo mercado, proporcionar novas funcionalidades inexistentes em outras ferramentas, e que são consideradas importantes para o ciclo de vida do *Scrum*, além de viabilizar o uso desta abordagem por times distribuídos. Ainda o fato de ser *open source*, contribui para que pequenas e médias empresas possam utilizar uma ferramenta com a maioria das funcionalidades existentes em ferramentas comerciais, sem precisar arcar com o ônus financeiro para sua aquisição.

1.3 PROBLEMA

O uso de artefatos simples como cartões, murais, cartolinhas e pôsteres (MOUNTAIN 2009) nas metodologias ágeis, muitas vezes podem não ser suficientes para o cenário globalizado e competitivo atual, onde cada dia projetos envolvendo times, usuários e clientes distribuídos fisicamente tornam-se realidade. Onde podemos identificar os seguintes problemas que dificultam a adoção do *Scrum*:

- Acompanhamento do quadro: apresentar informações sobre o andamento do projeto através de um quadro físico para membros de um time distribuído não é a melhor forma; uma vez que, eles geralmente não se encontram fisicamente próximos;

- Atualização manual das informações do quadro: dado que as informações são apresentadas visualmente no quadro, a atualização manual destas pode consumir um tempo significativo;
- Dificuldade para gerar relatórios: há um esforço para reunir os dados presentes no quadro com o intuito de calcular dinamicamente indicadores como quantidade de horas trabalhadas, gráficos de *burndown* e linhas de tendência;
- Manter histórico do projeto: como parte das informações do projeto encontra-se no quadro, em cartões, por exemplo, ao final do projeto, para criar um histórico deste, faz-se necessário a digitalização de toda esta informação.

A Figura 5 ilustra alguns dos problemas com quadro físicos e o uso de cartões como: um amontoado de cartões que são difíceis de consolidar, consultar e manter histórico; o quadro físico dificulta a visualização por membros de um time distribuído; a geração manual de gráficos de *burndown*; e a queda ou a perda de cartões.

Figura 5. Exemplos de artefatos e quadros físicos



1.3.1 Objetivo geral

Esta Dissertação tem como objetivo propor uma ferramenta para suportar o gerenciamento de projetos ágeis utilizando o *Scrum*, com foco em simplicidade e facilidade de uso, suprindo necessidades como apoio a equipes remotas, geração automática de gráficos, obtenção de métricas e históricos.

1.3.2 Objetivos específicos

Uma vez que artefatos simples como cartões, murais, cartolinhas e pôsteres são realidade em projetos utilizando metodologias ágeis, um dos desafios torna-se a construção de uma ferramenta que se aproxime da facilidade da utilização de tais artefatos. Para isso, é necessária a aplicação de um processo de desenvolvimento centrado no usuário tendo como objetivo alcançar a simplicidade e facilidade de uso.

A arquitetura deve contemplar a utilização de padrões abertos como *Web Services Soap*, padrão *RPC* (Remote Procedure Call) consolidado e implementado por diversas tecnologias, com o objetivo de fornecer extensibilidade e interoperabilidade com outras aplicações. *Frameworks* e padrões arquiteturais disseminados no mercado devem ser utilizados com o objetivo de aumentar a produtividade, robustez, facilidade de manutenção e customização da ferramenta.

1.4 CONTRIBUIÇÕES

As seguintes contribuições são esperadas:

- Projeto e implementação de uma ferramenta de suporte ao processo *Scrum* possibilitando seu uso por equipes distribuídas geograficamente, possibilitando, por exemplo, a realização de reuniões de planejamento e estimativas utilizando a técnica de *Planning Poker* através de recursos de áudio e vídeo em tempo real;
- Facilitar a adoção do *Scrum* por empresas de desenvolvimento de software, especialmente as pequenas e médias empresas que não

querem onerar a adoção de práticas ágeis de desenvolvimento de software com custo adicional de ferramentas;

- Apresentação e aplicação do conceito *RIA* (*Rich Internet Application*) e tecnologias relacionadas;
- Arquitetura de uma aplicação utilizando a tecnologia *RIA* Adobe Flex;

Produção de uma ferramenta de forma colaborativa no modelo *open source*, envolvendo estudantes, professores e profissionais.

1.5 ESTRUTURA DA DISSERTAÇÃO

Esta Dissertação está organizada em 5 capítulos, conforme descrição a seguir.

No Capítulo 2 é apresentado o estado da arte, descrevendo a fundamentação teórica e conceitos necessários para realização deste trabalho. São apresentados os princípios que fundamentam as metodologias ágeis, como surgiu o *Scrum*, quais seus princípios, papéis e ciclo de vida. O conceito de *RIA* é apresentado, os aspectos incluídos em aplicações deste tipo, assim como tecnologias existentes no mercado.

No Capítulo 3 é apresentada a solução proposta. Um pouco do histórico da evolução da ferramenta, características, funcionalidades e módulos, arquitetura e suas divisões, tecnologias utilizadas e requisitos mínimos de sistema.

No Capítulo 4 é apresentada a metodologia utilizada para identificar quais funcionalidades serão fornecidas pelo presente trabalho, e como serão abordadas.

Finalmente, no Capítulo 5 são apresentadas as considerações finais, evolução do trabalho e conclusões.

2 ESTADO DA ARTE

2.1 METODOLOGIAS ÁGEIS

O termo “Metodologias Ágeis” tornou-se popular em 2001 quando dezessete (17) especialistas em processos de desenvolvimento de software representando as metodologias *Scrum*, *Extreme Programming* (XP) e outros, estabeleceram princípios comuns compartilhados por todos esses métodos. Foi então criada a Aliança Ágil e o estabelecimento do “Manifesto Ágil” (AGILEMANIFESTO 2001). Os conceitos chave do “Manifesto Ágil” são:

- Indivíduos e interações sobre processos e ferramentas;
- Software executável sobre documentação;
- Colaboração do cliente sobre negociação de contratos;
- Respostas rápidas a mudanças sobre seguir planos.

2.2 SCRUM

O *Scrum* foi criado inicialmente como um *framework* para gerenciamento de projetos na indústria convencional e publicado no artigo *The New New Product Development Game* (TAKEUCHI e NONAKA 1986). O primeiro desenvolvimento de software com *Scrum* foi realizado em 1993 por Ken Jeff Sutherland na Easel Corporation (SUTHERLAND 2004), e junto com Ken Schwaber formalizaram o *Scrum* (SUTHERLAND e SCHWABER 2007) como processo de desenvolvimento na OOPSLA (SCHWABER 1997). O *Scrum* aceita que o desenvolvimento de software é imprevisível e formaliza a abstração, sendo aplicável a ambientes voláteis, se destacando dos demais métodos ágeis pela ênfase dada ao gerenciamento do projeto. Há atividades de monitoramento e *feedback*, em geral, através de reuniões rápidas e diárias com toda a equipe, visando à identificação e correção de deficiências e/ou impedimentos no desenvolvimento (SCHWABER 2002).

O *Scrum* é fundamentado na produção de software focada no produto, onde a participação do cliente/usuários é sempre ativa. Realizado de forma iterativa e incremental, o *Scrum* busca sempre, o mais cedo possível, adicionar funcionalidades que trazem maior valor de negócio para o cliente. Segundo Schwaber, muitas pessoas responsáveis por gerenciar projetos são educadas a usar uma abordagem determinística que utilizam planos detalhados com gráficos de *Gantt* e agendas de trabalho (SCHWABER 2004). A proposta do *Scrum* é exatamente o oposto, onde o cronograma é orientado ao produto que será produzido em cada iteração, que é planejada de acordo com a prioridade funcional definida pelo cliente e tendo o gerente com o papel de guiar o projeto para que siga o caminho mais efetivo. Estas iterações devem ter duração de duas a quatro semanas para atender rapidamente às necessidades do cliente. Nesta situação o prazo final não está claramente definido. No entanto, como o cliente recebe produtos constantes de acordo com sua própria orientação, há uma redução dos conflitos pela cumplicidade no processo.

2.2.1 Termos e conceitos

Alguns termos e conceitos serão introduzidos com o objetivo de um melhor entendimento do *Scrum*.

- **Product Backlog** – O *Product Backlog* é a lista de requisitos para um sistema, apresentada como uma lista priorizada de Itens do *Product Backlog*. Ela inclui tanto requisitos funcionais como requisitos não funcionais;
- **Sprint** – Uma *Sprint* é uma iteração onde um incremento das funcionalidades do produto são implementadas;
- **Sprint Backlog** – Representa Itens do *Product Backlog* selecionados para serem produzidos em uma *Sprint*, junto com as tarefas que devem ser concluídas para realizar o objetivo da *Sprint*,
- **Story Points** - Unidade de tamanho relativo utilizado na estimativa de requisitos como uma alternativa a unidade de tempo. *Points* é uma medida de complexidade a/ou tamanho de um requisito. São chamados *Story*

Point, pois são muito utilizados em conjunto com o termo *User Story*, no qual é muito utilizado em metodologias ágeis para as especificações de requisitos.

2.2.2 Papéis e Responsabilidades

O *Scrum* implementa um esqueleto iterativo e incremental, através de três (3) papéis principais (SCHWABER 2004):

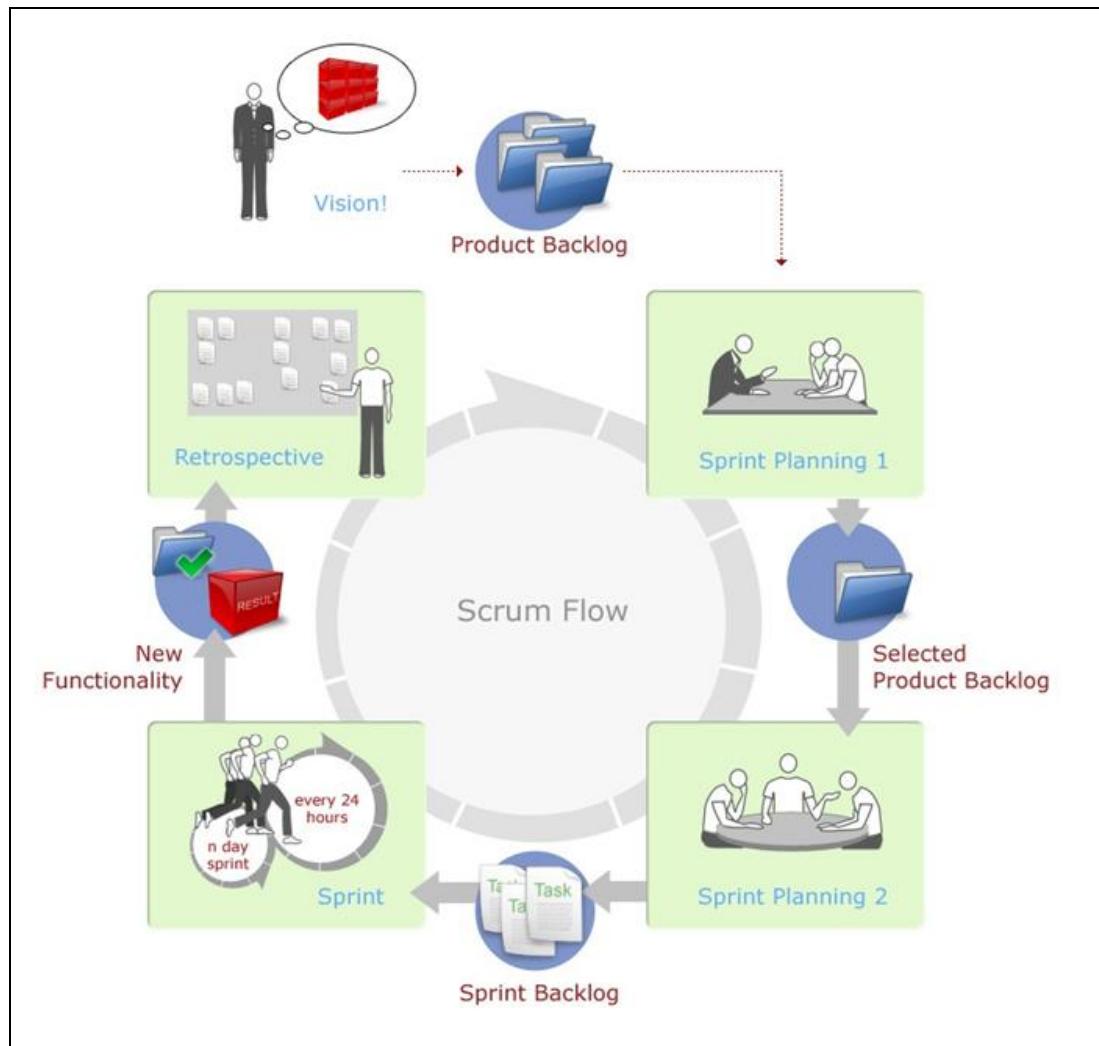
1. **Product Owner** – define as funcionalidades do produto, os itens do *Product Backlog*, que é a lista de funcionalidades do produto, o valor de negócios para cada item do *Backlog*, prioriza tais itens, aceita ou rejeita o resultado do trabalho. Pode ser o cliente, uma pessoa que represente o cliente, alguém de marketing e que deve estar disponível durante todo o projeto;
2. **Scrum Master** – assegura que as práticas do *Scrum* estão sendo executadas, remove os impedimentos levantados pelo time, protege o time de interferências externas, garante a colaboração entre os diversos papéis e funções. É aconselhável que o *Scrum Master* não assuma papel ou atividades do time, para evitar a falta de tempo para exercer suas responsabilidades, prejudicando assim o processo;
3. **Time** – estima e desenvolve as funcionalidades do produto; define como transformar o *Product Backlog* em incremento de funcionalidades numa iteração gerenciando seu próprio trabalho sendo responsáveis coletivamente pelo sucesso da iteração e consequentemente pelo projeto como um todo. A estimativa de esforço para cada item do *Product Backlog* geralmente são realizadas por unidades abstratas tal como *Story Points*.

2.2.3 Ciclo de Vida

No *Scrum*, um projeto se inicia com uma visão do produto que será desenvolvido (SCHWABER 2004). O ciclo de vida de projetos *Scrum* é definido em iterações que podem durar de duas a quatro semanas, conhecidas como *Sprint*. Ao

final de cada *Sprint* deve ser entregue um incremento do produto. Este incremento deve estar pronto, ou seja, tenha passado por todo o processo de desenvolvimento, auditoria e testes. O fluxo de desenvolvimento detalhado do *Scrum* é mostrado na Figura 6.

Figura 6. Visão geral do processo do Scrum.



Fonte: (SPRINT-IT 2006)

Antes do início da *Sprint* é realizada a reunião de planejamento, chamada de *Sprint Planning Meeting*, onde o *Product Owner* e o time decidem em conjunto o que será desenvolvido durante a *Sprint*. Esta reunião é dividida em duas partes, onde a primeira, chamada *Sprint Planning 1*, o *Product Owner* apresenta os requisitos de maior valor e prioriza aqueles que devem ser implementados primeiro. Neste momento o time pode argumentar que, por motivos técnicos, alguns itens precisam

ser implementados primeiro, e que devem ser considerados na priorização. Considerando sua capacidade de produção, o time colaborativamente com o *Product Owner* define o que será desenvolvido na próxima *Sprint* (*Selected Product Backlog*) e qual o objetivo da mesma, sendo esta ultima uma breve declaração que ilustre o foco do trabalho durante a *Sprint*. Na segunda parte da reunião, chamada de *Sprint Planning 2*, e onde o presença do *Product Owner* não é mais necessária, o time define as tarefas que serão necessárias para a realização de cada item do *Selected Product Backlog* que foram definidos na reunião anterior. Existem várias discussões com relação às estimativas de tarefas, se devem ser em horas ou dias. Sendo em horas, a indicação é que as tarefas não podem ser maiores do que 16 horas para serem concluídas, e sendo em dias todas as tarefas não podem levar mais que um dia. Caso ultrapassem esses limites, devem obrigatoriamente ser quebradas, em tarefas menores, sempre respeitando o limite. O objetivo maior dessa restrição é aumentar a precisão das estimativas e diminuir as dependências.

Durante a execução da *Sprint*, cada membro do time escolhe quais tarefas desejam realizar, ou seja, tarefas nunca devem ser atribuídas. Deve existir uma atualização diária da estimativa de trabalho restante, onde qualquer membro do time pode adicionar, mudar ou apagar tarefas que ainda não foram realizadas. Também deve ser realizada diariamente uma reunião de revisão, chamada *Scrum Daily Meeting*, que é conduzida pelo *Scrum Master*, com duração máxima de 15 minutos, estando todos de pé e não necessitando ser numa sala de reunião, cada membro do time responde a três perguntas: O que eu fez no projeto desde a última reunião? O que irá fazer até a próxima reunião? Quais são os impedimentos? As respostas não devem ser direcionadas para o *Scrum Master*, e sim para todo o time, mostrando o compromisso perante todos. Novas funcionalidades não podem ser adicionadas durante a *Sprint* e sim adicionadas ao *Product Backlog* para serem desenvolvidas nas próximas. Caso a mudança seja extremamente necessária e não possa esperar pelo final da *Sprint*, a mesma é abortada, sendo necessário realizar um novo planejamento.

Ao término da execução da *Sprint*, o time apresenta os resultados obtidos ao *Product Owner* que então valida se os objetivos da *Sprint* foram alcançados. Nesta apresentação, definida como *Sprint Review*, todos os *stakeholders* podem participar.

Tal apresentação deve ser informal, sem o uso de *slides*, sendo realizada a demonstração das novas funcionalidades ou arquitetura.

Em seguida o *Scrum Master* conduz a reunião de retrospectiva, chamada *Sprint Retrospective*, onde todos devem participar (time, *Product Owner* e *Stakeholders*), onde são levantados pontos do que foi bom durante a *Sprint* e o que se pode melhorar para a próxima.

Todo o ciclo então é reiniciado até a conclusão do produto final. Para monitoramento do andamento do projeto dois gráficos são gerados o *Product Burndown*, Figura 7, e o *Sprint Burndown*, Figura 8, que representam o quanto de funcionalidades foram entregues do produto e da *Sprint* respectivamente na linha do tempo.

Figura 7. Exemplo de *Burndown* do Produto exibindo o consumo de *story points* de cada Sprint.

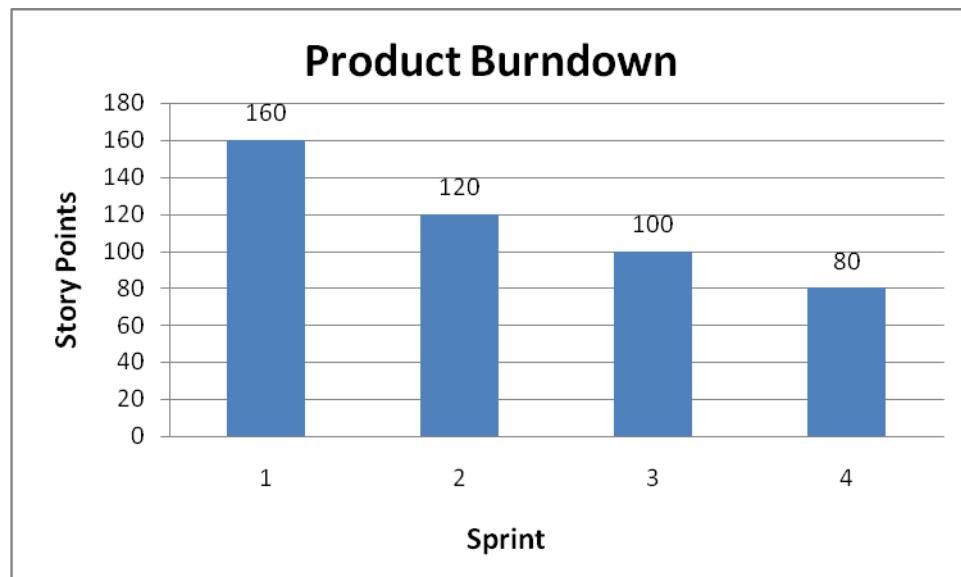
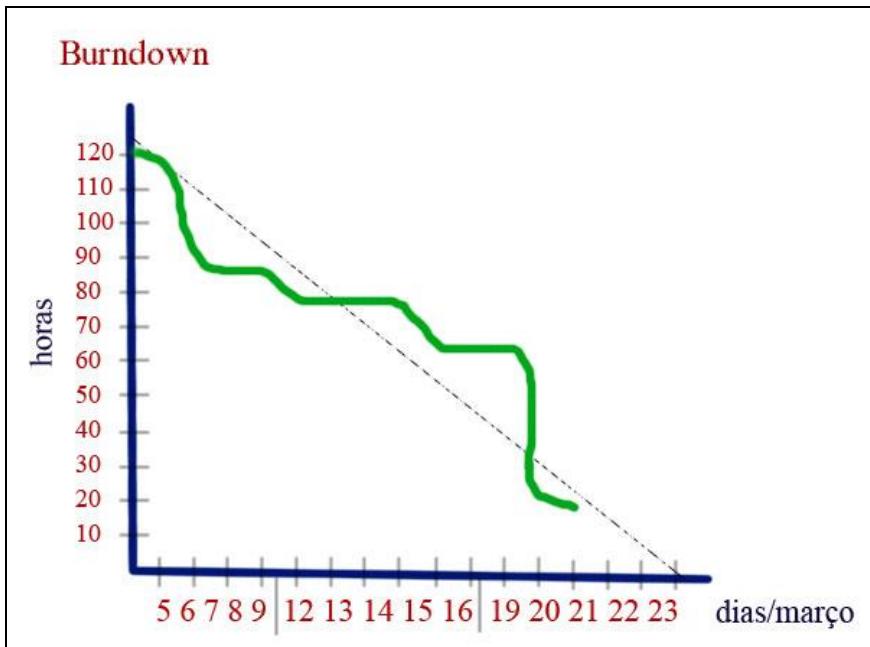


Figura 8. Exemplo de *Burndown* da *Sprint* exibindo o consumo de horas das tarefas durante a Sprint.



Conforme apresentado, o ciclo de vida do *Scrum* e seus artefatos podem ser considerados simples e de fácil implantação em projetos. Sendo assim, uma ferramenta que venha a apoiar o processo proposto pelo *Scrum* precisa compartilhar dos mesmos princípios, em caso contrário poderá ser considerada burocrática e inadequada. Então além das funcionalidades que devem ser fornecidas por uma ferramenta, deve-se existir uma preocupação com a forma como tais funcionalidades são apresentadas. Desta forma, para suprir a necessidade de implementar *interfaces* que possibilitem aumentar as experiências interativas com os usuários, a utilização de tecnologias que surgiram para este propósito devem ser consideradas.

2.3 RICH INTERNET APPLICATION

O termo *Rich Internet Application (RIA)* apareceu originalmente em um *white paper* da Macromedia, agora Adobe, em 2002 (ALLAIRE 2002). No entanto, termos com significados similares datam de antes, como: *Rich Web Clients* e *Rich Web Applications*. O objetivo de *R/As* é criar um novo modelo de aplicações web que supere as limitações do modelo até então vigente. Este consiste em normalmente construir clientes web baseados em HTML ditos magro, uma vez que a maior parte

do processamento acontece em servidores remotos. Apesar de esta abordagem ter se mostrado bem sucedida, ela possui limitações que ao longo do tempo tornaram-se cada vez mais críticas em função de uma maior demanda dos usuários por maior riqueza na interface das aplicações, na manipulação de conteúdo e dados na web. Esta necessidade impulsionou buscas por modelos mais ricos para aplicações web que combinassem a riqueza na manipulação de dados e possibilidades das interfaces de aplicações desktop com os benefícios das aplicações web. Portanto, *Rich Internet Applications (RIA)* são aplicações web que possuem características e funcionalidades de aplicações desktop tradicionais.

Aspectos incluídos em Aplicações de Internet Rica:

- Cada interação do usuário não necessita uma requisição e resposta do servidor web;
- Dados podem ser obtidos do servidor sem redesenhar a página inteira;
- Combina o melhor do desktop, web e comunicação;
- Utiliza a banda de rede de forma eficiente transmitindo somente a porção de dados alterada.

A seguir são apresentadas algumas tecnologias para Aplicações de Internet Rica encontradas no mercado.

2.3.1 AJAX

Acrônimo em língua inglesa de *Asynchronous Javascript And XML*. Ajax é baseado em ferramentas já familiares pelos desenvolvedores Web: HTML, DHTML, XML e JavaScript. Que podem tornar páginas mais interativas com o usuário, utilizando-se de solicitações assíncronas de informações é possível atualizar o conteúdo de uma página sem a necessidade de carregá-la novamente;

2.3.2 Adobe Flash

Antes chamado Macromedia Flash é uma plataforma multimídia originalmente adquirida pela Macromedia e atualmente desenvolvida e distribuídos pela Adobe Systems. Desde a sua introdução em 1996, o Flash se tornou um método popular para adicionar animação e interatividade a páginas web; Flash é comumente usado para criar animações, anúncios, e os vários componentes em páginas web, para integrar vídeos em páginas web e, mais recentemente, para desenvolver Aplicações de Internet Rica;

2.3.3 Adobe Flex

Antes chamado de Macromedia Flex e depois rebatizado como Adobe Flex pela Adobe, é o nome de uma tecnologia lançada em Março de 2004 pela Macromedia, que suporta o desenvolvimento de aplicações ricas para a Internet, baseadas na plataforma do Macromedia Flash;

2.3.4 Adobe Integrated Runtime

O Adobe Integrated Runtime (AIR), é um ambiente de execução multi-plataforma para a construção de Aplicações de Internet Ricas usando o Adobe Flash, Adobe Flex, HTML, ou Ajax, que pode ser implementado como uma aplicação desktop;

2.3.5 Laszlo

Laszlo é uma plataforma de código aberto desenvolvida pela Laszlo Systems Inc.. O servidor OpenLaszlo compila programas escritos em linguagem LZX (uma mistura de tags XML e JavaScript), com em DHTML (também conhecido como AJAX) ou Adobe Flash bytecode, que atualmente suporta Flash7 e Flash8. O servidor, que inicialmente era um software proprietário, teve seu código aberto em Outubro de 2004 graças a uma licença pública. Ele é a única Aplicação Rica de Internet que é capaz de compilar em dois diferentes runtimes a partir do mesmo código base;

2.3.6 Microsoft Windows Presentation Foundation

O Windows Presentation Foundation (ou WPF), inicialmente chamado de Avalon, é um subsistema gráfico no *.NET Framework 3.0* (inicialmente chamado de WinFX), que usa uma linguagem de marcação, conhecida como XAML para desenvolvimento de interfaces ricas;

2.3.7 Microsoft Silverlight

Inicialmente lançado em 2007 como WPF/E. O Silverlight junto com o WPF surgiu como uma iniciativa da Microsoft para entrar no mercado Web 2.0/RIA. Apesar do Silverlight oficialmente apenas suportar a plataforma Windows e MAC OS, uma iniciativa *open source* chamada Moonlight que estende o suporte para a plataforma Linux e Unix/X11 está sendo patrocinada pela Microsoft em parceria com a NOVELL.

2.3.8 JavaFx

Iniciativa da Sun Microsystems baseada em java para a criação e disponibilização de Aplicações de Internet Rica que podem ser executadas em vários ambientes *desktop*, *web* e em dispositivos móveis.

3 SOLUÇÃO PROPOSTA – FIRESCRUM

É uma ferramenta de apoio à gestão de projetos utilizando Scrum desenvolvida no modelo open source, sob a licença GPL, através de um processo centrado no usuário e utilizando tecnologias que elevam o nível de experiência dos usuários no uso das interfaces.

O início do desenvolvimento do FireScrum se deu durante o mestrado profissional do C.E.S.A.R em Outubro de 2007, na disciplina de Fábrica de Software, sendo o FireScrum o produto proposto a ser desenvolvido pela fábrica FIRE (FIRE 2007).

Em Maio de 2008, o FireScrum representou o estudo de caso de uma das fábricas, na disciplina de Engenharia de Software da pós-graduação do Centro de Informática da UFPE (Cin), onde foi implementada a primeira versão do módulo *Taskboard*.

Em Abril de 2009, na mesma disciplina do Cin na UFPE, reformulada para um modelo de desenvolvimento de software ágil utilizando o Scrum e apoiada pelo INES (Instituto Nacional de Ciência e Tecnologia para Engenharia de Software), o FireScrum passou a ser evoluído junto com a implementação de novos módulos (INES 2009).

Neste capítulo será apresentado o FireScrum. Também serão discutidas suas principais características, funcionalidades, arquitetura, e as tecnologias utilizadas para seu desenvolvimento.

3.1 CARACTERÍSTICAS GERAIS

O FireScrum é uma ferramenta *open source* de apoio ao gerenciamento ágil de projetos utilizando o *Scrum*. Desenvolvida utilizando conceitos da *Web 2.0* e de *Rich Internet Applications*, a ferramenta possui um esforço grande de usabilidade para garantir a simplicidade do seu uso. O FireScrum é uma aplicação Web, o que significa que a mesma é acessível através de um *browser*, podendo ser utilizada em um ambiente de Internet ou Intranet. Tendo ainda as seguintes características:

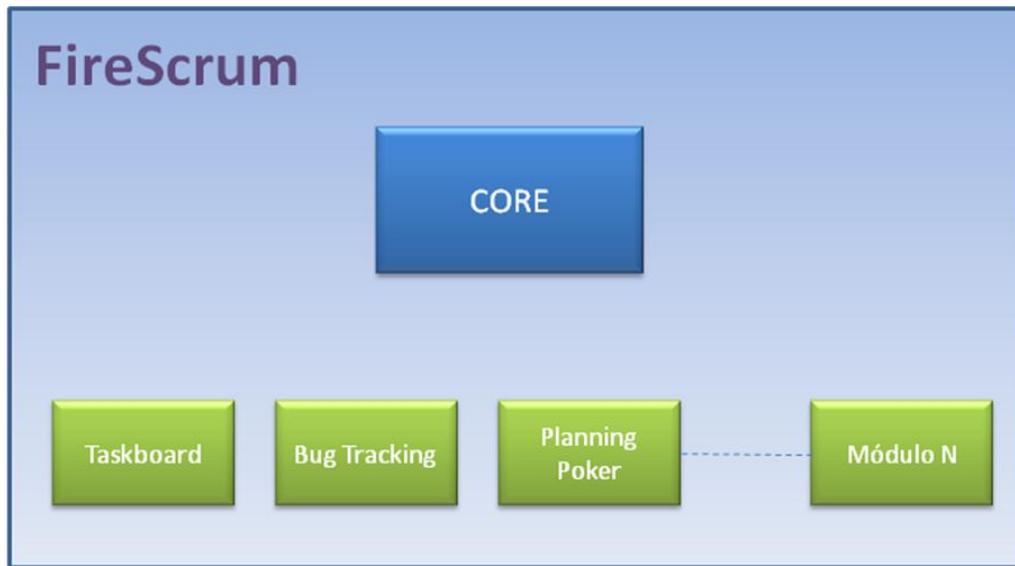
- Foco no *Scrum* – diante do crescimento do *Scrum* no mercado se destacando das demais metodologias ágeis e por sua natureza gerencial;
- Foco em Usabilidade – propõe uma interface rica e ao mesmo tempo simplista, prática e focada, alinhando-se ao formato de ferramentas desejáveis para serem utilizadas em metodologias ágeis;
- Extensível – Através de uma arquitetura modular e extensível é possível a criação de novos módulos, expandindo assim suas funcionalidades;
- API de integração – através de uma API (*Application Program Interface*) é possível realizar a integração do FireScrum com ferramentas e aplicações de terceiros.

O FireScrum por ser uma aplicação de internet rica (RIA) combina o melhor das funcionalidade das interfaces de aplicações desktop com a alta disponibilidade, baixo custo de distribuição de aplicações *web*, o melhor da interatividade, e da comunicação multimídia.

3.2 FUNCIONALIDADES

Conforme a Figura 9 a estrutura modular do FireScrum permite que novas funcionalidades sejam adicionadas conforme a necessidade de uso, ou quando novos módulos forem criados ao longo do tempo.

Figura 9. Estrutura modular do FireScrum



As funcionalidades do FireScrum estão divididas em módulos. Isso facilita ainda mais seu uso uma vez que apenas os módulos desejáveis podem ser utilizados, evitando assim que o FireScrum se torne uma aplicação complexa e burocrática para os usuários.

Os seguintes módulos estão sendo produzidos para o FireScrum:

- Core - Módulo principal que é responsável por toda a parte operacional básica do *Scrum*. Dentre as funcionalidades disponíveis no módulo core destacam-se: (1) controle de acesso, (2) cadastro de usuários, (3) cadastro de mais de um projeto (tem-se um projeto para cada produto), (4) criação de itens de *backlog*, (5) priorização de itens de *backlog*, (6) criação de *sprints*, (7) associação de itens de *backlog* a *sprints*, (8) criação de tarefas para um dado item de *backlog*, (9) alocação de usuário em determinada tarefa, (10) gráfico de *Burndown* da *Sprint* e (11) gráfico de *Burndown* do Produto. Este é o único módulo que não é opcional uma vez que serve de base para os demais. A Figura 10 apresenta uma imagem do módulo Core.

Figura 10. Tela principal do módulo *Core* do FireScrum.

Product Backlog			+BLI	+Task
Uncommitted Backlog Items	Backlog Effort	Business Value		
Como jornalista, desejo publicar notícias	8	700		
Como redator, desejo autorizar ou não uma notícia	5	900		
Como agência de publicidade, desejo consultar e gerenciar campanhas	8	500		
Como jornalista, desejo publicar vídeos	30	200		
Como gerente de marketing, desejo que os anúncios sejam automaticamente gerenciados	21	500		
Como jornalista, desejo publicar automaticamente notícias	40	200		
Como jornalista, desejo publicar chamadas para ação	5	400		
Como gerente de marketing, desejo ter anúncios personalizados	15	500		
Como jornalista, desejo cadastrar uma galeria de imagens	13	300		
Como gerente de marketing, desejo exibir previsões de vendas	20	400		
Como gerente de marketing, desejo fornecer o histórico de vendas	15	300		
Como administrador, desejo configurar quais seções serão exibidas	40	900		
Como redator, desejo saber estatísticas de publicações	9	200		
Como administrador, desejo definir as permissões de acesso	21	500		
Como gerente de marketing, desejo exibir os resultados da estratégia	21	100		
Como gerente de marketing, desejo saber estatísticas de vendas	5	400		

Committed Backlog		
Sprint 1	Backlog Effort	Business Value
Como gerente, desejo consultar páginas mais visitadas	15	900
Como jornalista, desejo cadastrar banners	21	700
Como jornalista, desejo cadastrar enquetes	13	800
Como designer, desejo criar hotsites	30	600

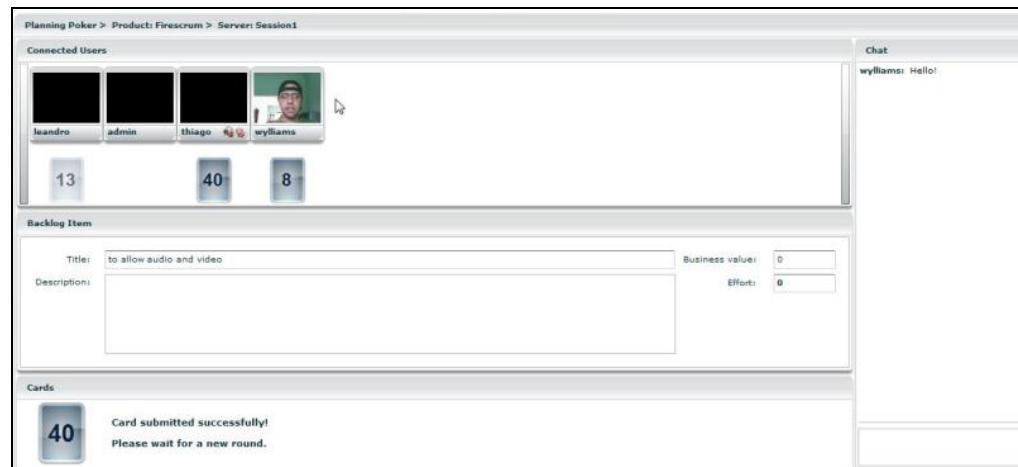
- TaskBoard - O objetivo deste módulo é reproduzir o quadro físico utilizado em projetos reais baseados em *Scrum*. O intuito é permitir que, através do quadro, os usuários da aplicação possam manipular cartões virtuais de forma simples e, além disso, possibilitar uma boa visibilidade do andamento do projeto. Dentre as funcionalidades disponíveis no módulo *TaskBoard* destacam-se: (1) visualizar os itens de *backlog* para a *sprint* selecionada, (2) priorizar os itens de *backlog* da *sprint* selecionada, (3) cadastrar tarefas (representadas por cartões virtuais) para um dado item de *backlog* informando um responsável e uma estimativa de esforço, (4) mudar o estado das tarefas (*Todo*, *Impeded*, *In Progress* e *Done*) arrastando-as pelo quadro, (5) cadastrar impedimentos, (6) associar impedimentos a tarefas ou itens de *backlog*, (7) efetuar *zoom in* ou *zoom out* no quadro, (8) aplicar filtros visuais para destacar tarefas que possuam uma dada informação e (9) visualizar o percentual de progresso da *sprint* corrente. A Figura 11 apresenta uma imagem do módulo *TaskBoard*.

Figura 11. Imagem do módulo *TaskBoard* do FireScrum.



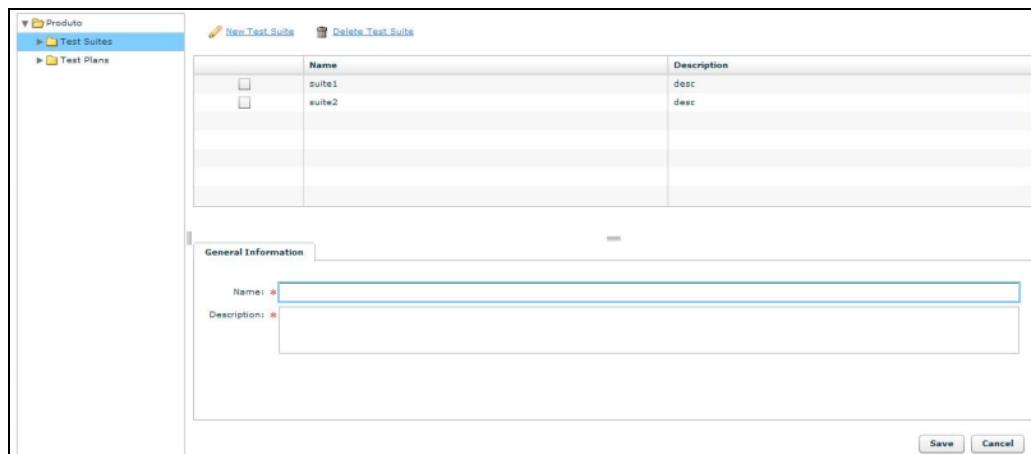
- Planning Poker – O objetivo deste módulo é viabilizar a técnica de estimativa *planning poker* ser realizada por times, *product owner* e *stakeholders* geograficamente distribuídos. Dentre as funcionalidades disponíveis neste módulo destacam-se: (1) visualizar os itens do *backlog* a serem estimados, (2) definir os valores das estimativas a serem utilizados, (3) possibilidade de compartilhar áudio e vídeo dos participantes, (4) *chat* entre os participantes e (5) gerenciar o processo de estimativa entre os participantes relativo ao item de *backlog* escolhido. A Figura 12 apresenta uma imagem do módulo Planning Poker.

Figura 12. Imagem do módulo Planning Poker do FireScrum.



- Test Management – O objetivo deste módulo é possibilitar a criação de casos de testes e seu gerenciamento, evitando assim a utilização de ferramentas externas para tal finalidade. Dentre as funcionalidades disponíveis neste módulo destacam-se: (1) criação de plano de teste, (2) criação de casos de teste, (3) criação de *job* de execução de testes, (4) associar casos de teste a itens do *backlog*, (5) registro dos resultados dos testes, (6) criar biblioteca de casos de testes (suítes) e (7) disponibilizar relatórios dos testes. A Figura 13 apresenta uma imagem do módulo *Test Management*.

Figura 13. Imagem do módulo *Test Management* do FireScrum.



- Bug Tracking – O objetivo deste módulo é fornecer recursos para o registro de *bugs* e todo o ciclo de vida necessário para a solução dos mesmos, evitando assim a utilização de ferramentas externas para tal finalidade. Dentre as funcionalidades disponíveis neste módulo destacam-se: (1) registrar um *bug*, (2) associar um item de *backlog* a um *bug*, (2) definir um responsável pelo *bug*, (3) adicionar notas ao *bug*, (4) anexar arquivos ao *bug*, (5) enviar e-mails para as pessoas envolvidas no *bug*, (6) manter histórico do *bug* e (7) relatórios com filtro sobre os *bugs*. A Figura 14 apresenta uma imagem do módulo *Bug Tracking*.

Figura 14. Imagem do módulo Bug Tracking do FireScrum.

- Desktop Agent – Este módulo que se trata de uma aplicação desktop, tem como objetivo facilitar o acesso as funcionalidades do FireScrum, uma vez em que o mesmo quando executado estará disponível através da bandeja do Sistema Operacional, onde o usuário terá acesso a tais funcionalidades através de um menu suspenso. Dentre as funcionalidades disponíveis neste módulo destacam-se: (1) visualizar todas as tarefas, (2) visualizar tarefas do usuário, (3) editar tarefas, (4) receber alertas do sistema, (5) reportar bugs e (6) fornecer um chat entre os integrantes do time. A Figura 15 apresenta uma imagem do módulo *Desktop Agent*.

Figura 15. Imagem do módulo *Desktop Agent* do FireScrum.

FireScrum

All Tasks

Task	Description	Owner	Spent today	Remaining	Status
t1	dfsfd	admin			
t2	sdfsdfsd				
sdf	dfsdfsdf	admin			
t3 ed	gdfgdgf_ed	admin			
Task 2	sdfsdfsdf				
task 4	dfsdf				
task 5	dfsdfsdf	admin			
Editing	new description	admin			

TO DO
 DOING
 DONE
 IMPEDED
 Allocate
 Deallocate

Refresh
 Cancel

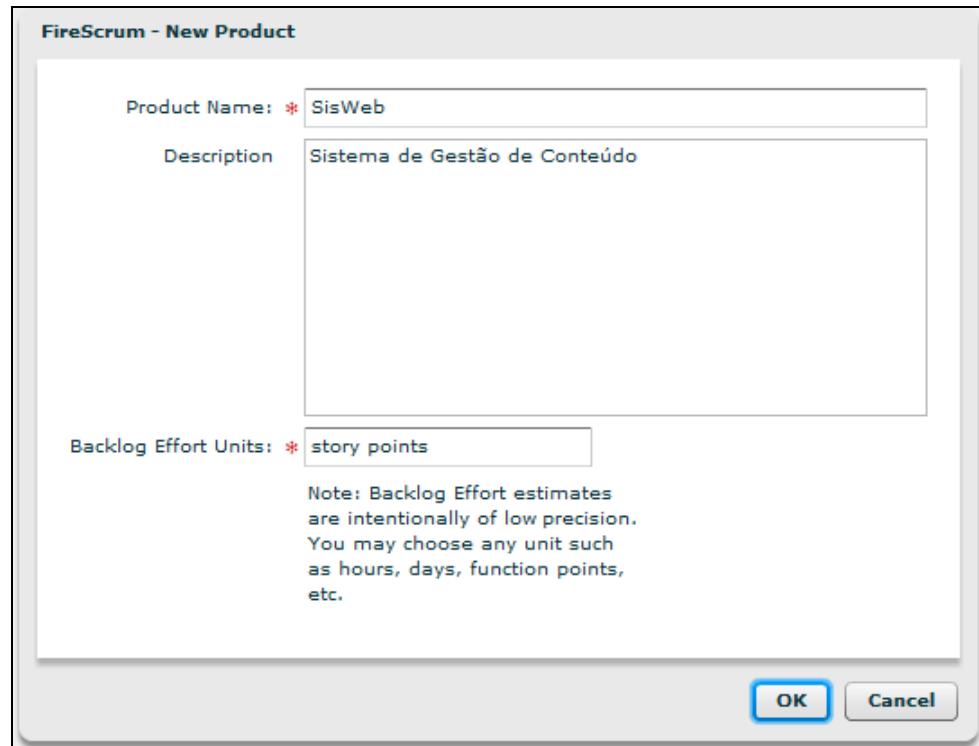
Uma vez que o módulo *Core* é considerado o módulo principal do sistema e responsável por toda a parte operacional básica do *Scrum*, o mesmo deve ser auto-suficiente para contemplar todo o ciclo de vida básico de um projeto *Scrum*, sendo possível utilizá-lo de forma independente em projetos *Scrum*. Os demais módulos são considerados módulos de apoio, contemplando necessidades e ferramentas que possam tornar o FireScrum uma ferramenta ainda mais atrativa, excluindo a necessidade de ferramentas de terceiros que apóiem o desenvolvimento de software como um todo.

A seguir serão analisadas algumas das funcionalidades principais do módulo *Core*.

- **Cadastro de Produto**

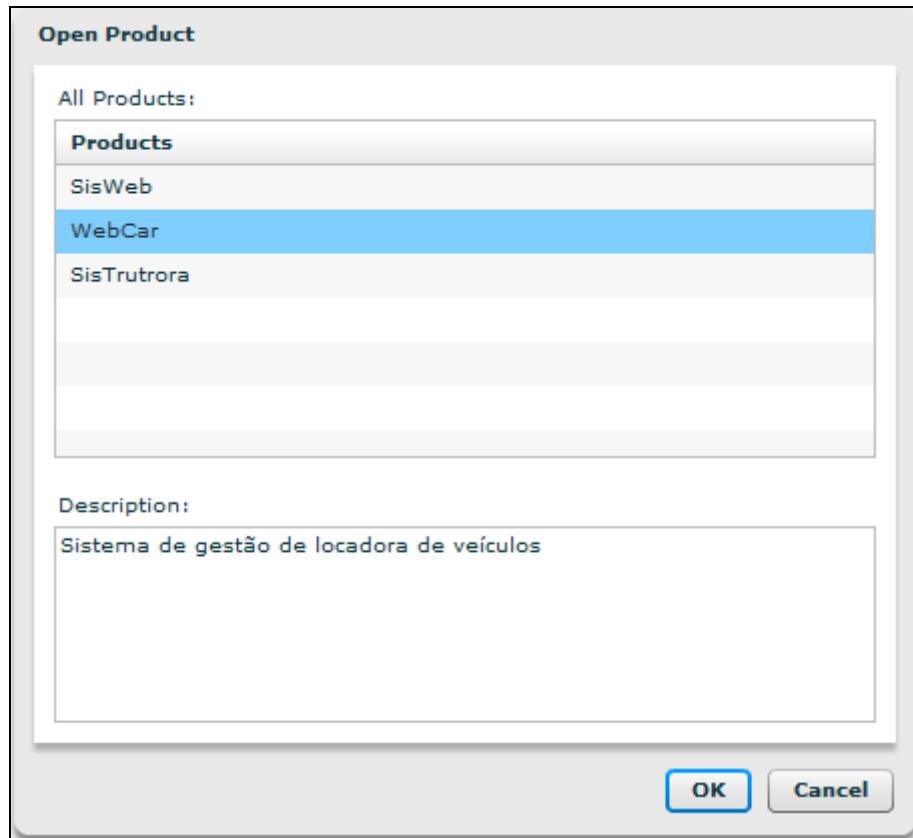
No FireScrum o software a ser desenvolvido é chamado *Produto* (mesma nomenclatura adota pelo *Scrum*). Sendo solicitado um **nome**, uma **descrição** e a **unidade de esforço ou tamanho** que será estimada pelo time para cada *Item de Backlog do Produto* como mostra a Figura 16.

Figura 16. Tela de cadastro de Produto.



Uma vez que o produto é cadastrado o mesmo passa a ser automaticamente o produto corrente do sistema. O usuário poderá alternar entre os produtos já cadastrados na tela *Open Product* conforme figura 17.

Figura 17. Tela de seleção do Produto.



- **Cadastro de Item de Backlog**

Uma vez o produto corrente selecionado é possível cadastrar *Itens de Backlog* para o mesmo, o que representará o *Backlog do Produto*. A Figura 18 mostra a tela de cadastro de Item de *Backlog* onde são solicitados o **título**, uma **descrição**, o **valor de negócio** deste item que deve ser informado pelo *Product Owner* e o **esforço** para ser produzido que deve ser informado pelo *Time*.

Figura 18. Tela de cadastro do Item de *Backlog*.

FireScrum - Backlog Item

Title: * Como editor, quero aprovar as matérias publicadas

Description:

- As matérias são publicadas pelos jornalistas
- O editor tem um prazo máximo de 2 horas para aprovar a matéria
- Caso o prazo expire a matéria será publicada automaticamente

Business value: 800

Effort: 8 story points

OK Cancel

Uma vez os itens de *backlog* do produto cadastrado, o FireScrum exibe o *Product Backlog* na tela principal do sistema conforme Figura 19.

Figura 19. *Backlog* do Produto (*Product Backlog*) na tela principal.

Product Backlog				+BLI	+Task
Uncommitted Backlog Items	Backlog Effort	Task Hours	Business Value		
Como jornalista quero publicar entrevistas	8		700		
Como editor, desejo adicionar revisão a ui	3		600		
Como marketing, quero publicar banners i	3		600		
Como marketing, quero publicar patrocina	5		400		
Como usuário, quero assinar uma RSS da	8		600		
Como usuário, quero pesquisar no conteú	13		900		
Como editor, quero estatísticas de publicaç	20		900		
Como administrador, quero definir perfis	8		100		
Como jornalista, quero publicar um álbum	13		300		
Como jornalista, quero publicar vídeos	8		500		
Como administrador, quero cadastrar os us	13		400		
Como jornalista, quero publicar notícias	5		700		
Como editor, quero aprovar as matérias p	8		800		

- **Priorizar os Itens de Backlog**

Os Itens de *Backlog* são priorizados levando-se em conta sua ordem de apresentação no *Backlog* do Produto, os itens no topo da lista são os que possuem maior prioridade, sendo os itens no final da lista aqueles que possuem menor prioridade. Para definir esta ordem o FireScrum proporciona o recurso de arrastar e soltar (*drag and drop*) para os itens possibilitando assim seu reordenamento conforme Figura 20.

Figura 20. Priorizando os Itens de *Backlog* do Produto.

Product Backlog				+BLI	+Task
Uncommitted Backlog Items	Backlog Effort	Task Hours	Business Value		
Como editor, quero estatísticas de publicações	20		900		
Como editor, desejo adicionar revisão a uma matéria	3		600		
Como marketing, quero publicar banners	3		600		
Como marketing, quero publicar patrocínios	5		400		
Como usuário, quero assinar uma RSS da mídia	13	8	900	600	
Como usuário, quero pesquisar no conteúdo	13		900		
Como jornalista, quero publicar entrevistas	8	8	100	700	
Como administrador, quero definir perfis	8		100		
Como jornalista, quero publicar um álbum	13		300		
Como jornalista, quero publicar vídeos	8		500		
Como administrador, quero cadastrar os usuários	13		400		
Como jornalista, quero publicar notícias	5		700		
Como editor, quero aprovar as matérias publicadas	8		800		

- **Cadastro de Sprint**

No cadastro de uma *Sprint* são definidos o nome, descrição, data inicial, data final e o objetivo da mesma conforme Figura 21.

Figura 21. Tela de cadastro de *Sprint*.

FireScrum - New Sprint

Sprint Name: * Sprint 1

Description: Primeira Sprint

Start Date: 23/04/2009

End Date: 06/05/2009

Goals: Disponibilizar o módulo administrativo

OK Cancel

- **Definindo o Selected Backlog**

Para definir o Selected Backlog, itens do Backlog que serão produzidos na próxima iteração, é preciso arrastar os Itens de Backlog do Produto para a Sprint conforme Figura 22.

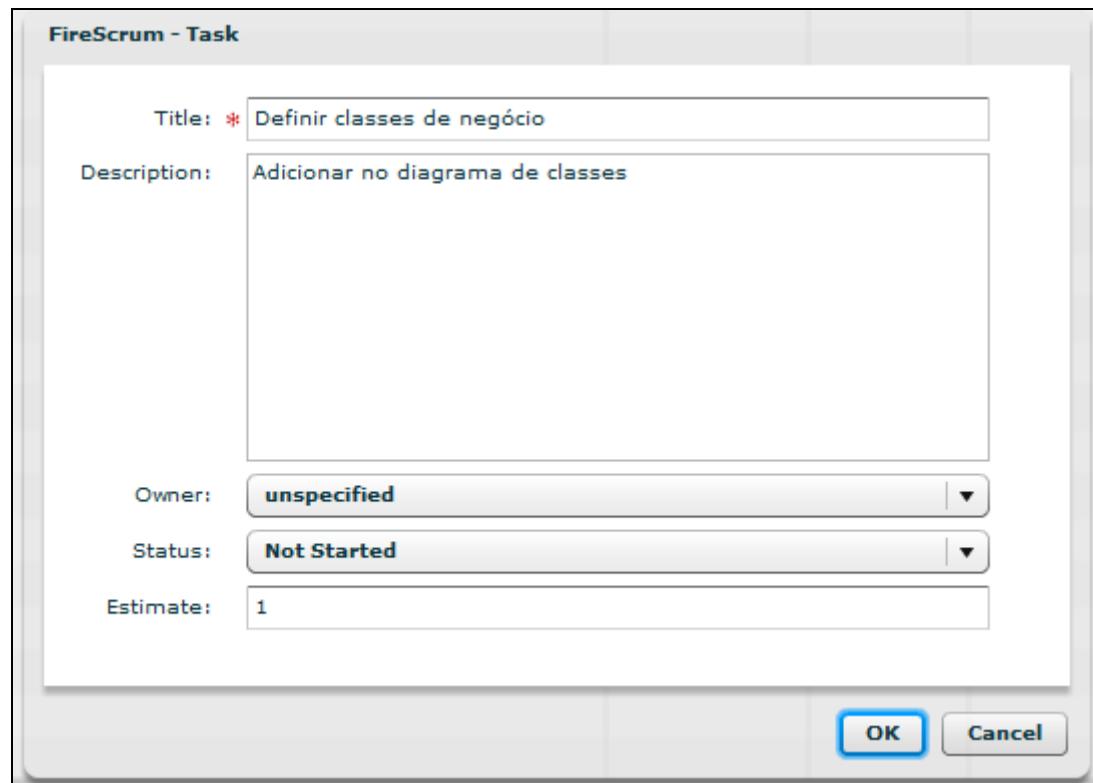
Figura 22. Definindo o Selected Backlog.

Product Backlog				+BLI	+Task	Committed Backlog				-Sprint
Uncommitted Backlog Item	Backlog	Task	Business	Committed Backlog Items				Backlog	Task	Business
Como editor, desejo adicionar...	3		600	Como editor, desejo adicion...	3		600			
Como marketing, quero p...	3		600	Como marketing, quero pub...	3		600			
Como editor, quero estatís...	20		900	Como editor, quero estatís...	20		900			
Como administrador, quer...	13		400	Como administrador, quer...	13		400			
Como jornalista, quero pu...	8		500	Como jornalista, quero pu...	8		500			

- **Cadastro de Tarefas**

Uma ou mais tarefas, atividades necessárias para construir os itens do *backlog*, podem ser cadastradas para cada item do *Backlog* conforme Figura 23, onde são informados o título da tarefa, descrição, responsável, status e a estimativa para conclusão da tarefa que é comumente reportada em horas.

Figura 23. Tela do Cadastro de Tarefa



- **Sprint Backlog**

Uma vez que as tarefas são cadastradas para cada item de *backlog* da *Sprint*, passamos a ter o *Sprint Backlog* conforme Figura 24.

Figura 24. Sprint Backlog

Sprint 1				
Committed Backlog Items	Backlog Effort	Task Hours	Business Value	
▼ Como editor, desejo adicionar revisão a um artigo	3		600	
Implementar		6		
Executar testes		2		
▼ Como marketing, quero publicar banners de negócios	3		600	
Definir classes de negócio		1		
Implementar classes de negócio		2		
Executar testes		2		
Implementar CRUD		6		
▼ Como editor, quero estatísticas de publicações	20		900	
Implementar		8		
Executar testes		2		

- **Gráficos de Burndown**

Os gráficos de *Burndown* podem ser gerados a partir do histórico das atualizações das tarefas e conclusão da *Sprint*. Como estas funcionalidades ainda não foram implementadas no FireScrum as Figuras 25 e 26 apresentam dos *mockups* do *Sprint Burndown* e do *Product Burndown* respectivamente.

Figura 25. Mockup da tela do *Sprint Burndown*

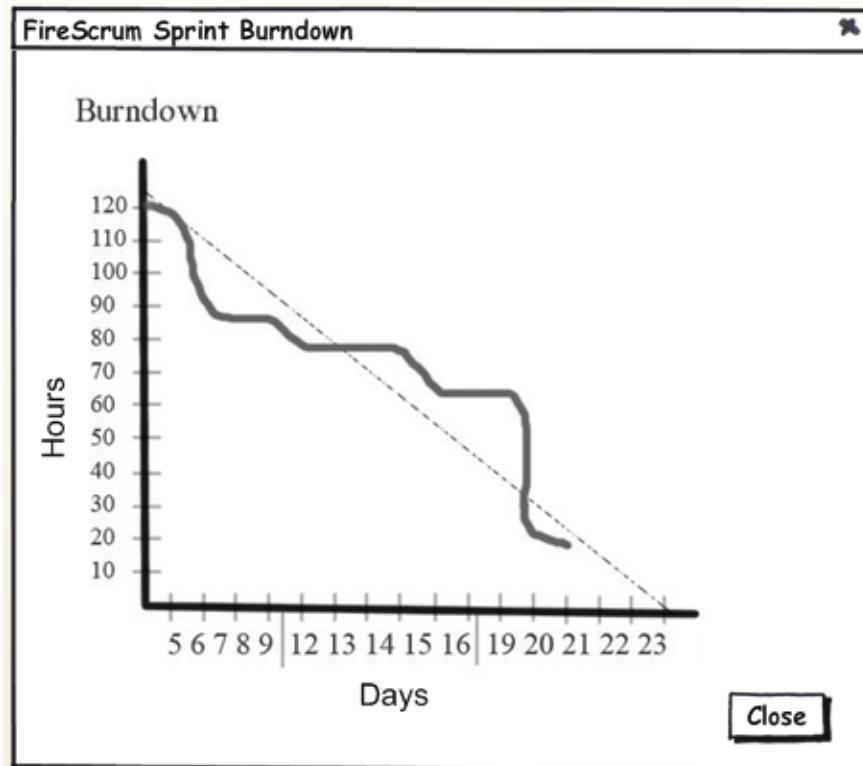
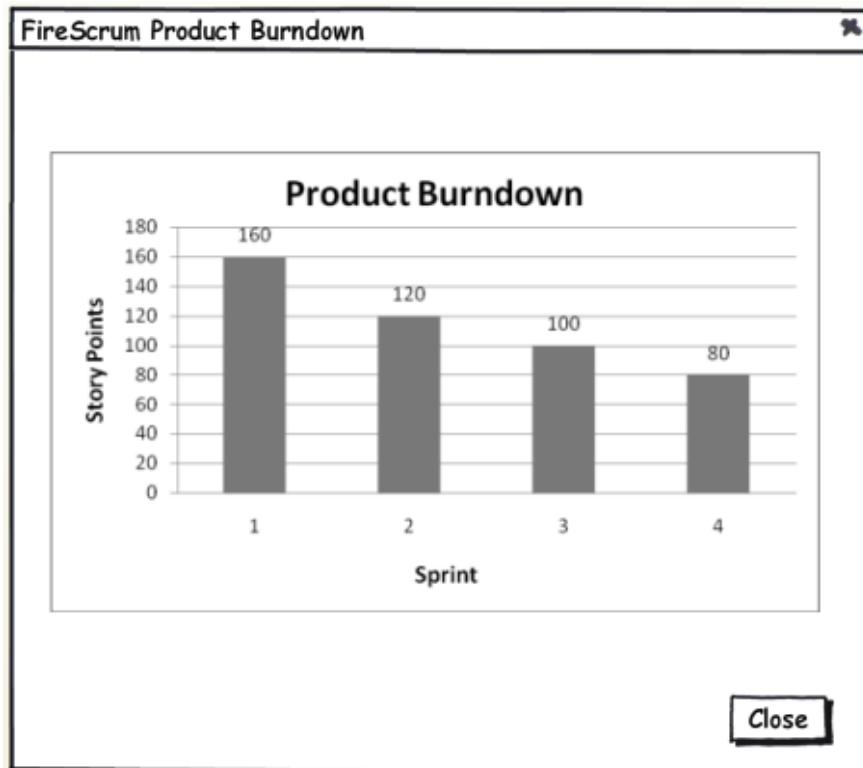


Figura 26. Mockup da tela do *Product Burndown*



- **Impedimentos**

Os impedimentos podem ser cadastrados por qualquer membro do time. A Figura 27 apresenta o *mockup* desta funcionalidade.

Figura 27. *Mockup* da tela de Impedimentos

Date	Impediment	Reporter	Resolved
20/03/2009	Falta de informações do PO	Eric	<input type="checkbox"/>
22/03/2009	Feedback do usuário	Jamerson	<input checked="" type="checkbox"/>
25/03/2009	Erro no componente de terceiros	Tarciana	<input type="checkbox"/>

3.3 ARQUITETURA

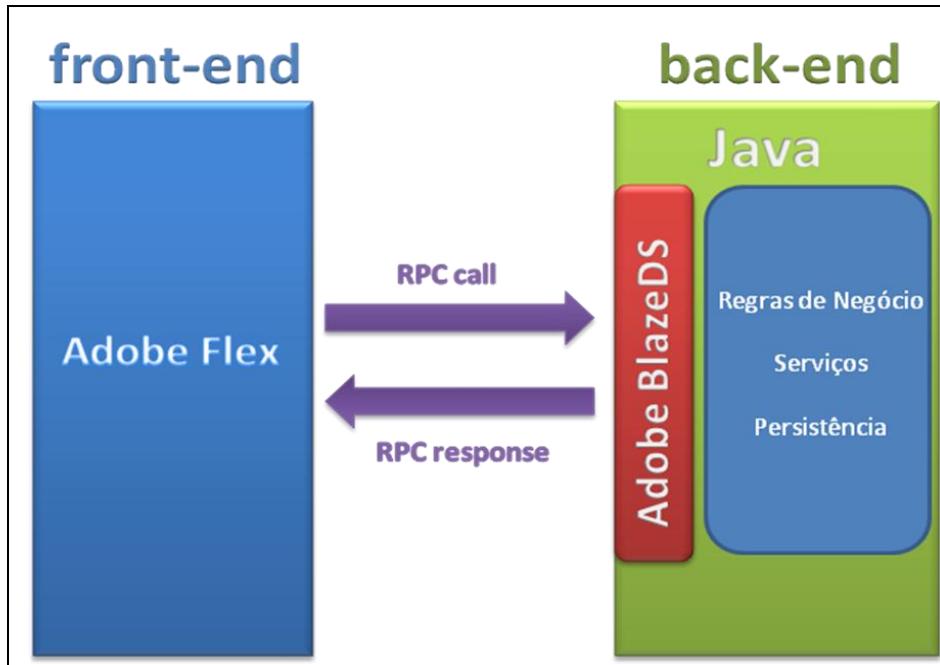
Conforme apresentado do Capítulo 3.2, a estrutura modular do FireScrum permite que novas funcionalidades sejam adicionadas conforme a necessidade de uso, ou quando novos módulos forem criados ao longo do tempo. Para que isso seja possível é necessária a criação de uma arquitetura que contemple esta modularidade proposta pelo FireScrum.

A arquitetura macro do FireScrum está dividida em duas partes: o *front-end* e o *back-end*. O *front-end*, também conhecido como camada de apresentação, é composto por toda parte da aplicação responsável pela interface do usuário e seu comportamento. Sendo acessível através de um browser com o Flash Player instalado. Este requisito se dá em função do uso do Adobe Flex que é uma *framework* de código aberto da Adobe que permite a criação e manutenção de *Rich Internet Applications*.

Já o *back-end* é desenvolvido utilizando a plataforma Java que contém a regra de negócio, os serviços e o mecanismo de persistência da aplicação. Tais serviços são acessados remotamente por cada usuário pelo *browser* através do *front-end*. Para que seja possível a comunicação entre o Adobe Flex e o Java de forma eficiente, foi utilizado o BlazeDS que é uma tecnologia da Adobe (BLAZEDS 2008) que possibilita a comunicação entre aplicações desenvolvidas em Adobe Flex/Air utilizando um protocolo binário denominado AMF(*Action Message Format*).

Na Figura 28 é apresentada a arquitetura macro da aplicação onde os serviços Java do *back-end* são expostos através do BlazeDS e consumidos pelo *front-end* Adobe Flex. A comunicação é realizada através de chamadas RPC (*Remote Procedure Call*) utilizando o formato de mensagem AMF.

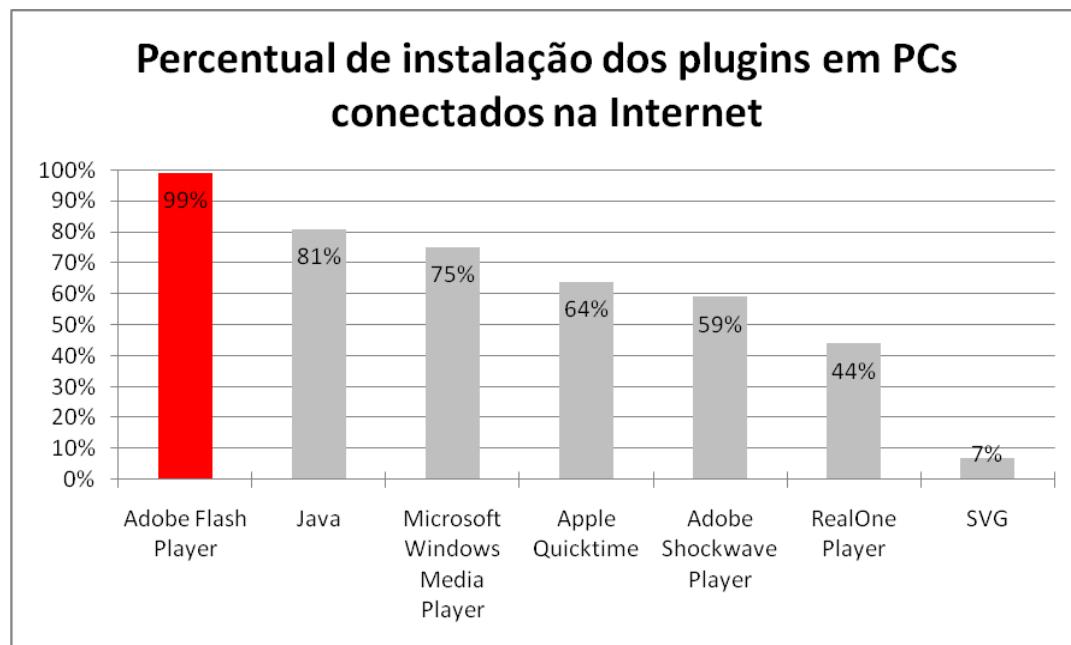
Figura 28. Arquitetura macro



3.3.1 Arquitetura do *front-end*

O Adobe Flex é um poderoso *framework* para o desenvolvimento de aplicações de internet ricas (RIA), e por ter o Flash Player como seu ambiente de execução no qual segundo a Adobe está instalado em 99% dos PCs conectados na internet conforme a Figura 29, o torna uma excelente opção para desenvolvimento de aplicações web multiplataforma.

Figura 29. Percentual de penetração do Adobe Flash Player.



Fonte: (Adobe 2009)

Na arquitetura do *front-end* foi adotado o *Cairngorm*, que é uma micro-arquitetura (coleções de padrões de projetos) para desenvolvimento de aplicações de internet rica para o Adobe Flex, desenvolvido pela Consultoria da Adobe, e que encoraja as melhores práticas de desenvolvimento e potencializa os fundamentos do framework Flex, facilitando seu uso para médias e grandes equipes de engenheiros que trabalham em projetos de médio e grande porte, assim como em aplicações de missão crítica (CAIRNGORM 2008).

Segundo Webster e Tanner, o *Cairngorm* aborda três áreas chaves que a Consultoria da Adobe tem recomendado como melhor para seus clientes e parceiros ao longo do tempo (WEBSTER e TANNER 2008):

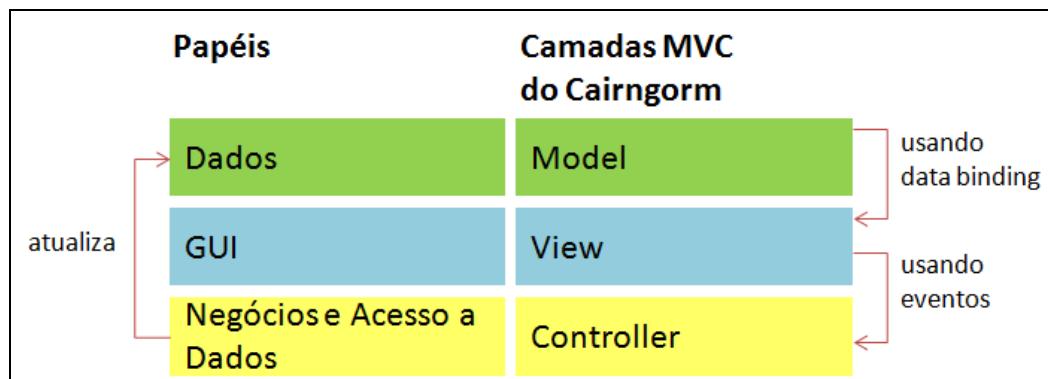
1. Manipular interações do usuário no cliente;
2. Encapsular lógica de negócios e interações com o servidor;
3. Gerenciar o estado no cliente e a representação deste estado na interface do usuário.

O *Cairngorm* oferece uma micro-arquitetura para ajudar consistentemente a resolver esses desafios de design que são recorrentes. Outros propósitos no uso do *Cairngorm* são identificados como:

- Organizar e partitionar: códigos e pacotes; funcionalidades dos componentes e seus papéis;
- É uma metodologia de “boas práticas” de *design* e desenvolvimento de aplicações em Flex;
- Encorajam desenvolvedores a identificar, organizar, e separar o código baseando-se em seus papéis e responsabilidades.

O *Cairngorm* utiliza o padrão arquitetural MVC onde a separação de papéis e responsabilidades são bem evidentes conforme Figura 30.

Figura 30. Mapeando funcionalidades dos papéis para camadas do *Cairngorm*.



A importância do padrão MVC pode ser diretamente atribuída à separação evidente de três elementos sem que exista a sobreposição de cada uma de suas responsabilidades. A seguir são apresentadas as camadas da arquitetura MVC:

- **View:** renderiza os dados através de *data binding* e lançam eventos baseado nas interações dos usuários;
- **Controller:** Captura os eventos lançados pelas *views* e processam a lógica de negócio e atualizam o *model*;

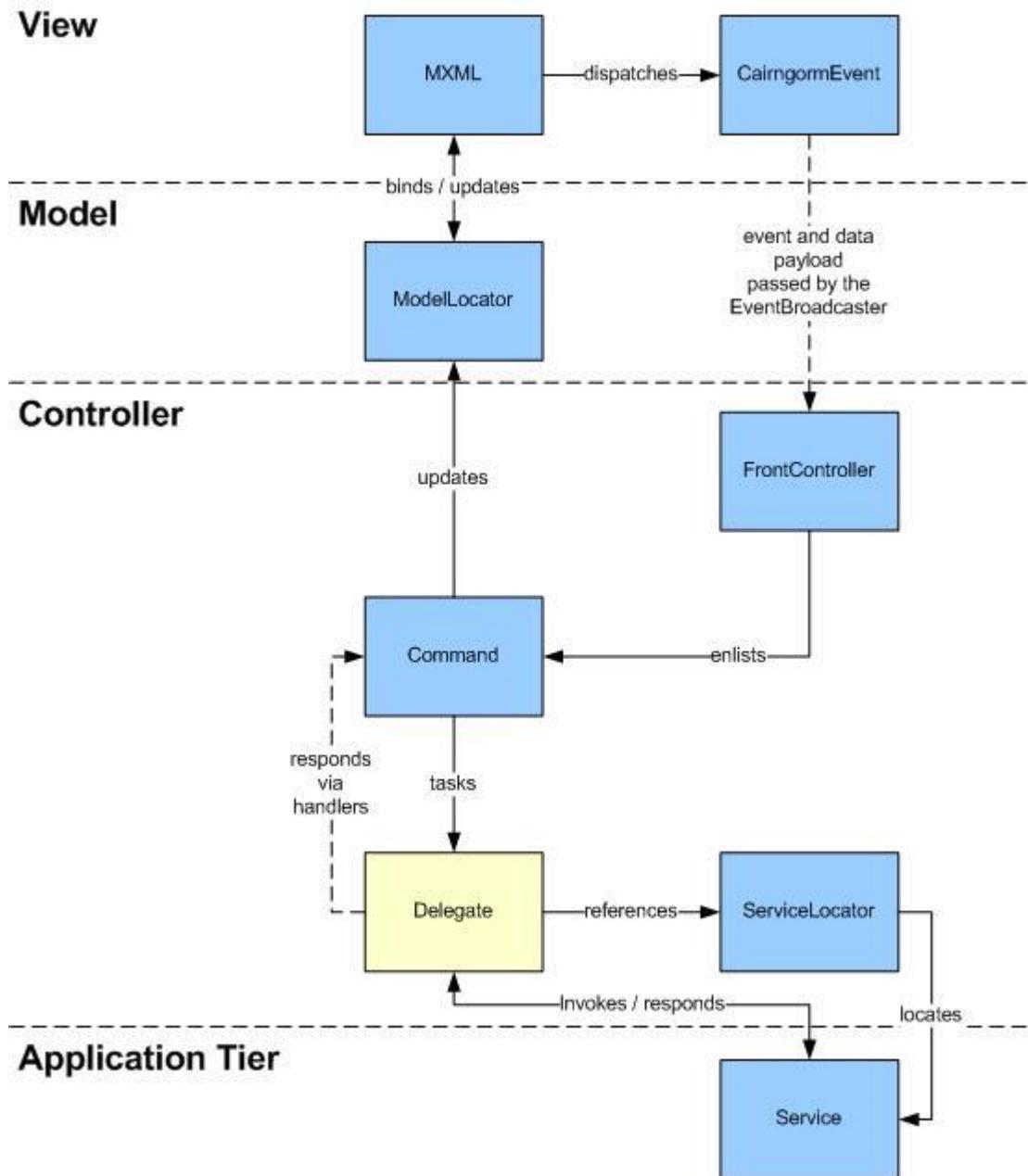
- **Model:** Mantém os dados da aplicação e seus estados. Uma vez atualizado, as *views* que tem ligação com o *model* são também atualizadas para representar este novo estado.

O *Cairngorm* utiliza alguns outros padrões de projetos o que o torna uma micro-arquitetura composta de vários padrões envolvidos pelo padrão MVC. A seguir são apresentados os elementos que compõe o padrão MCV do *Cairngorm*:

- **Model Locator:** Armazena o estado da aplicação e suas variáveis em um único lugar;
- **View:** Renderiza os dados e lançam eventos baseado nas interações do usuário como cliques, arrastar e soltar e etc.;
- **Front Controller:** Recebem os eventos mapeiam-nos para seus respectivos *Commands*;
- **Command:** Manipulam a lógica de negócio, invocam os *Delegates* e/ou outros *Commands*, e atualizam os *Value Objects* e variáveis armazenadas no *Model Locator*;
- **Delegate:** Criado por um *Command*, ele instancia chamadas de procedimentos remotos (HTTP, Web Services, etc) e enviam o resultado de volta para o *Command*;
- **Service:** Define as chamadas de procedimentos remotos (HTTP, Web Services, etc) para conectar com dados armazenados remotamente.

A Figura 31 mostra o fluxo de eventos do *Cairngorm* e onde cada um de seus elementos está inserido no padrão MVC.

Figura 31. Elementos do *Cairngorm* e fluxo de eventos.



Fonte: (CHOPRA 2009)

Conforme a Figura 31 o fluxo do *Cairngorm* acontecesse na seguinte ordem:
 Um evento que contém dados relevantes é lançado pela *View* em um determinado momento, o *FrontController* encaminha o evento para o *Command* apropriado, o *Command* por sua vez invoca um serviço através do *Delegate*, o retorno do serviço é retornado ao *Command* através do próprio *Delegate*, que por sua vez atualiza o

Model via *Model Locator*. Uma vez que a *view* está realizando *data binding* com o *Model*, a mesma é atualizada apresentando o novo estado dos dados.

Para suportar a arquitetura modular do FireScrum o Adobe Flex conta com o componente *Module*. A partir deste componente é possível dividir uma aplicação Adobe Flex em módulos que podem ser carregados dinamicamente pela aplicação principal, onde no caso do FireScrum é o que é chamado de módulo *Core*.

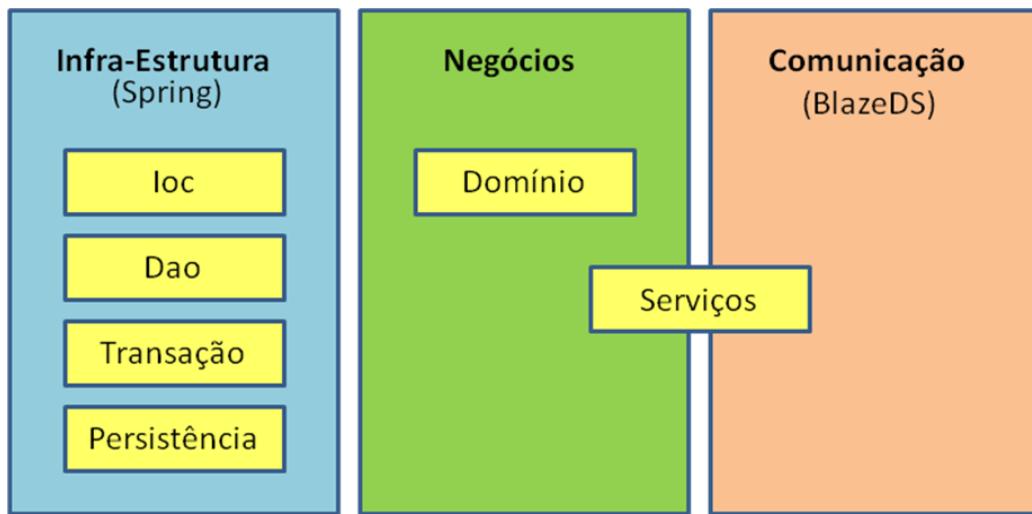
3.3.2 Arquitetura do *back-end*

O *back-end* é composto basicamente por entidades de negócios, serviços e persistência. A escolha pela tecnologia Java para o *back-end* se deu pelo fato da existência de uma grande infra-estrutura fornecida por tal tecnologia para a implementação de aplicação corporativas, além da existência do BlazeDS fornecido pela Adobe como produto *Open Source* e que tem como objetivo integrar aplicações Flex com o Java utilizando o protocolo AMF.

A Figura 32 apresenta o diagrama da arquitetura do *back-end* que está dividido nas seguintes camadas e suas responsabilidades:

- Infra-Estrutura: gerenciada pelo *framework* Spring é responsável pela injeção de dependência, reduzindo assim o acoplamento entre as classes do sistema, controle de persistência e transação das entidades do sistema;
- Negócios: contêm as entidades do domínio do sistema como *Product*, *Backlog Item*, *Tasks* e etc e seus serviços;
- Comunicação: gerenciada pelo BlazeDS é responsável pela publicação dos serviços de negócio através do protocolo AMF da Adobe, viabilizando a comunicação entre o *back-end* e o *front-end*.

Figura 32. Diagrama da arquitetura do *back-end*.



3.4 REQUISITOS TECNOLÓGICOS MÍNIMOS PARA UTILIZAÇÃO DA FERRAMENTA

O FireScrum é uma aplicação Web que para ser utilizada pelo usuário só necessita de um navegador com o plug-in do Adobe Flash Player versão 9 ou superior instalado.

No lado do servidor o FireScrum necessita dos seguintes softwares:

- Java 1.5 ou superior;
- Servidor Web/Container Java (Ex. Tomcat, JBoss e similares);
- Gerenciador de Banco de Dados compatível com Hibernate;
- Servidor RTMP Red 5 (para módulos como o *taskboard* e *planning poker*).

4 METODOLOGIA

Na construção deste trabalho foram realizadas várias atividades para definição e validação das funcionalidades propostas. A Tabela 1 apresenta as atividades realizadas e quais seus objetivos.

Tabela 1. Metodologia - Atividades e Objetivos.

Atividades	Objetivos
Levantamento do Estado da Arte	Entender e se posicionar no contexto a ser inserido
Análise de ferramentas similares existentes no mercado	Identificar oportunidades, tecnologias, pontos positivos e negativos das ferramentas.
Definição Funcional	Definir funcionalidades a serem contempladas pela ferramenta
Implementações de protótipos e validações com usuários	Avaliar se o que está sugerido está sendo bem recebido pelos usuários. Identificar pontos de melhorias.
Participação de listas de discussões nacionais e internacionais sobre o assunto	Identificar necessidades do mercado, perfil do público-alvo, e aceitação no uso de ferramentas.
Criação de site com divulgação de telas e vídeos sobre a ferramenta	Receber <i>feedbacks</i> da comunidade sobre a proposta da ferramenta.
Aperfeiçoamento da ferramenta	Continuar evoluindo a ferramenta de acordo com as necessidades identificadas pelo mercado.

4.1 ANÁLISE DE FERRAMENTAS

Na análise de ferramentas foram avaliadas algumas das mais conhecidas no mercado, que são:

- VersionOne – ferramenta desenvolvida pela VersionOne possuindo uma versão comercial (*Enterprise Edition*) e uma versão gratuita por um ano (*Team Edition*) com limitações de número de usuários e recursos;
- TargetProcess – ferramenta desenvolvida pela TargetProcess possuindo uma versão comercial (*Full Edition*) e uma versão gratuita (*Community Edition*) limitada a 5 usuários;
- ScrumWorks – ferramenta desenvolvida pela Danube Technologies possuindo uma versão comercial (*Pro Edition*) e uma versão gratuita (*Basic Edition*) com recursos mais básicos;
- Agilo for Scrum – ferramenta *open source* desenvolvida pela agile42 GmbH implementada como plug-in para o software wiki e gerenciador de *tickets* chamado Trac.

Dentre as ferramentas avaliadas somente o ScrumWorks e o Agilo for Scrum são voltadas totalmente para o *Scrum*, o que não invalida a utilização das outras para este fim. O que foi percebido é que ao usar nomenclaturas já conhecidas do *Scrum* como, por exemplo, *Product Backlog*, *Backlog Item* e *Sprint*, em vez de nomes mais genéricos, existiram um maior conforto e entendimento de uso por parte dos usuários apresentados a tais ferramentas. A interface mais agradável para os usuários foi a do ScrumWorks, isso justifica a semelhança do FireScrum com a mesma em alguns pontos.

A Tabela 2 apresenta uma análise comparativa de funcionalidades relevantes entre as ferramentas analisadas e o FireScrum:

Tabela 2. Análise comparativa de funcionalidades.

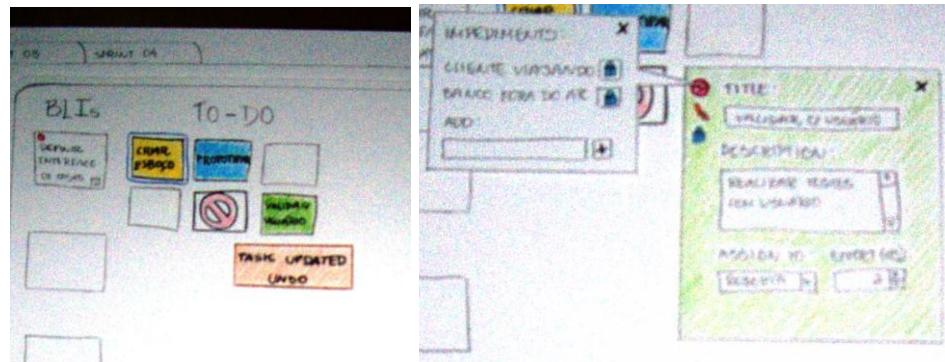
FUNCIONALIDADES	VERSIONONE	TARGETPROCESS	SCRUMWORKS	FIRESCRUM	AGILÓ FOR SCRUM
Product Management	X	X	X	X	X
Sprint Management	X	X	X	X	X
Report and Analytics	X	X	X	X	X
User Management	X	X	X	X	X
Web Services API	X	X	X	X	-
Bug Tracking	X	X	-	X	X
TaskBoard	X	X	X	X	-
Test Management	-	X	-	X	-
Sprint Review	X	-	-	-	-
Sprint Retrospective	X	-	-	-	-
Release Planning	X	X	X	X	X
Impediments Management	-	X	X	X	X
Desktop Agent	-	-	-	X	-
Planning Poker	-	-	-	X	-

Conforme a Tabela 2, é possível identificar duas funcionalidades exclusivas do FireScrum, que são o *Desktop Agent* e o *Planning Poker*. Tais funcionalidades têm como objetivo viabilizar e favorecer a comunicação no uso do *Scrum* em ambientes com times distribuídos.

4.2 PROTÓTIPOS E VALIDAÇÕES COM USUÁRIOS

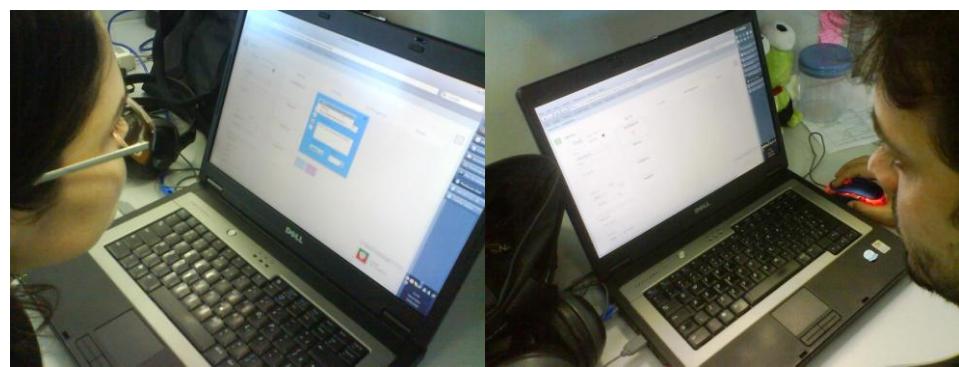
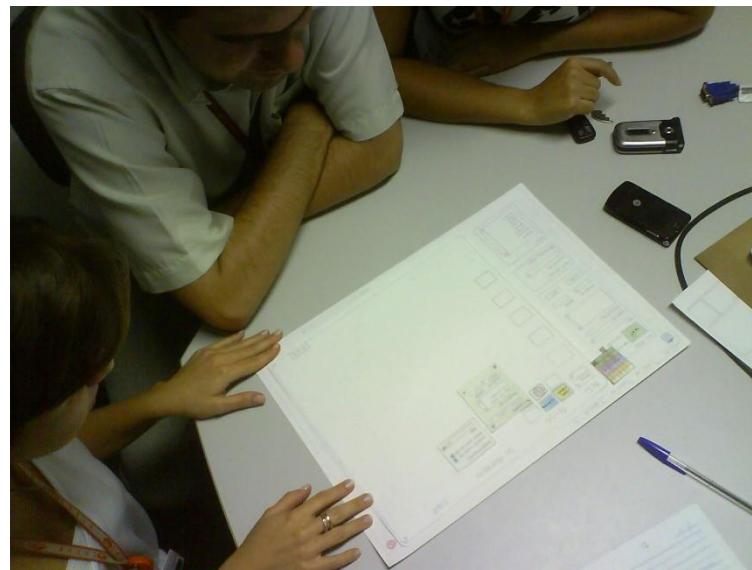
As técnicas de prototipagem e validação com usuários estão sendo de grande importância para a definição de interfaces e seus comportamentos. A figura 33 mostra exemplos de protótipos em papel do *taskboard*.

Figura 33. Protótipos em papel



Uma vez que os protótipos estejam construídos, os mesmos passam por validações dos usuários conforme a figura 34.

Figura 34. Validação com usuários



4.3 PARTICIPAÇÃO DE LISTAS DE DISCUSSÕES

A participação em listas de discussões sobre Scrum é uma forma de acompanhar as discussões em torno da metodologia, assim como o sentimento dos praticantes de *Scrum* com relação à utilização de ferramentas de apoio.

A lista mais importante internacionalmente é a **scrumdevelopment** criada em fevereiro de 2000 e que conta hoje com cerca de 6.700 participantes (YAHOO GROUPS 2000), no qual participam grandes nomes do Scrum como Ken Schwaber, Mike Cohn, e Boris Gloger.

4.4 SITE DA FERRAMENTA

A criação de um site exclusivo para o FireScrum (FIRESCRUM 2007) tem como objetivo identificar o nível de interesse mundial sobre a ferramenta utilizando soluções como o Google Analytics (ANALYTICS 2009), assim como receber *feedbacks* da comunidade, Figura 35 e 36. No anexo A é fornecido trechos de e-mails de pessoas interessadas na ferramenta que entraram em contato através do site.

Figura 35. Site oficial do FireScrum.

The screenshot shows the FireScrum website homepage. At the top, there's a navigation bar with links to HOME, OVERVIEW, SCREENSHOTS, DOWNLOAD, TEAM, and CONTACT US. Below the navigation is a search bar. The main content area features a section titled "FireScrum Workshop" with a summary of the event: "FireScrum workshop occurred on last Monday (June 1) in UFPE. We discussed about Scrum, how real companies use it and the importance of a nice tool to help teams in the journey. Almost 70 people were present. Boris Gloger was there." Below this is an "Agenda" list:

- Introduction (Jones Albuquerque/Silvio Meira) - 10h-10:15h
- How We use Scrum - 10:15h- 10:30h
- FireScrum Presentation (Eric and Scrum teams) - 10:30 - 11:00h
- Experiences in the Market- CESAR (Izabella Lira) - 11:00h-11:20h
- Experiences in the Market - Pitang (Luiz Borba) - 11:20h-11:40h
- Boris Gloger - 11:40-12:30h
- Discussing - 12:30h-13h

Below the agenda, there's a section titled "[View with PicLens]" showing a grid of eight thumbnail images of people at the workshop. To the right of the workshop section are logos for partners: "A PRODUCT OF" with a date "2009", "INES" (a green silhouette of a person's head), "PARTNERS" with "CESAR.edu" (a flame icon), "Centro de Informática UFPE" (a red and white logo), and "caju" (a yellow logo).

Figura 36. Informações de acesso providas pelo Google Analytics.



5 CONSIDERAÇÕES FINAIS

Com a globalização do mercado mundial, a procura por redução de custos e mão-de-obra qualificada potencializa a necessidade de desenvolver software por times geograficamente distribuídos. Ferramentas de apoio ao desenvolvimento de software e gestão de projetos passam a ser fundamentais neste cenário.

Em uma proposta de uma ferramenta de apoio ao *Scrum* mais do que fornecer as funcionalidades que cobrem o processo de desenvolvimento, é preciso estar alinhado com os princípios ágeis, evitando *overhead* desnecessário, sendo simples e objetiva. Diante desta premissa, o uso de recursos tecnológicos para aumentar a qualidade das interfaces gráficas e o nível de experiência com os usuários devem ser considerados.

A adoção do FireScrum por projetos que já utilizam o *Scrum* e aqueles que ainda pretendem iniciar seu uso, pode elevar a produtividade e a qualidade do processo de desenvolvimento de software. Por ser *Open Source*, a ferramenta favorece especialmente pequenas e médias empresas que não querem onerar a adoção de práticas ágeis de desenvolvimento de software com custo adicional de ferramentas.

No cenário globalizado atual a necessidade em desenvolver software de forma distribuída é cada vez mais evidente, tendo como principal objetivo redução de custos e contratação de mão-de-obra especializada. Desta forma o FireScrum pode viabilizar o uso do *Scrum* por equipes geograficamente distribuídas, fornecendo inclusive recursos de compartilhamento de informações, áudio e vídeo em tempo real.

O ciclo de vida do *Scrum* e seus artefatos podem ser considerados simples e de fácil implantação em projetos, sendo assim, uma ferramenta que venha a apoiar o processo proposto pelo *Scrum* precisa compartilhar dos mesmos princípios, em caso contrário poderá ser considerada burocrática e inadequada. Então além das funcionalidades que devem ser fornecidas por uma ferramenta, deve-se existir uma preocupação com a forma como tais funcionalidades são apresentadas. O uso de

tecnologias que elevem as experiências interativas com os usuários, conceitos e técnicas de usabilidade devem ser considerados no desenvolvimento de uma ferramenta para este propósito.

O modelo *Open Source*, o envolvimento de estudantes, professores e profissionais, e o apoio do INES (Instituto Nacional de Ciência e Tecnologia para Engenharia de Software) foram e estão sendo fundamentais para a construção do FireScrum que tem data prevista para o lançamento da primeira release oficialmente pública programada para Julho de 2009.

5.1 EVOLUÇÃO DO TRABALHO

Como evolução do trabalho, após existir a primeira versão funcional da ferramenta, deve-se aplicar estudos de casos, no qual podem apresentar valiosas informações de pontos de melhoria e de novas funcionalidades.

O desenvolvimento de novos módulos também já está sendo previsto como trabalhos futuros. No segundo semestre de 2009, como mais uma iniciativa do INES (Instituto Nacional de Ciência e Tecnologia para Engenharia de Software), os módulos de *Sprint Review* e *Sprint Retrospective* estão planejados para serem desenvolvidos durante a disciplina de Tópicos Avançados de Engenharia de Software da pós-graduação do Centro de Informática da UFPE (Cin).

Algumas funcionalidades planejadas como o uso de uma tela projetada com a tecnologia *multi-touch* pode simular ainda mais o uso do quadro de tarefas (taskboard) físico, o que evitaria o uso de cartões de papel, contribuindo assim também com a natureza. O uso dos recursos de áudio e vídeo disponíveis no módulo *Planning Poker*, também poderá ser utilizado para viabilizar as reuniões diárias com times geograficamente distribuídos, como também adicionados ao módulo *Desktop Agent* para facilitar a comunicação entre os integrantes de um projeto.

Com o lançamento da primeira versão oficial pública, o FireScrum pretende, cada vez mais junto com a comunidade, evoluir para se tornar a melhor ferramenta *open source* de apoio ao *Scrum* disponível no mercado mundial.

REFERÊNCIAS

ADOBE. Flash Player Penetration, 2009. Disponível em:
http://www.adobe.com/products/player_census/flashplayer/. Acesso: 05/04/2009.

AMBLER, Scott, Has Agile Peaked? Let's look at the numbers, 2008. Disponível em:
<http://www.ddj.com/architect/207600615>. Acesso em: 17/05/2009.

ANALYTICS, Google. Disponível em: <http://www.google.com/analytics/>. Acesso: 16/05/2009.

AGILEMANIFESTO, 2001, “Manifesto for Agile Software Development”
<http://agilemanifesto.org/>. Acesso em: 02/04/2008.

ALLAIRE, J. Macromedia Flash MX – A Next-Generation Rich Client. Macromedia white paper, 2002.

BARDUSCO, Camilo. Scrum & CMMi, 2008. Disponível em:
<http://blog.bardusco.com/2008/05/26/scrum-cmmi/>. Acesso em: 29/03/2009.

BECK, Kent, Jeff SUTHERLAND, Martin FOWLER, Alistair COCKBURN, Mike BEEDLE, Arie van BENNEKUM, Ward CUNNINGHAM, James GRENNING, Jim HIGHSMITH, Andrew HUNT, Ron JEFFREIS, Jon KERN, Brian MARICK, Robert C. MARTIN, Steve MELLOR, Ken SCHWABER, Dave THOMAS. “Manifesto for Agile Software Development” 2001. Disponível em: <http://www.agilemanifesto.org>. Acesso em: 2/04/2008.

BECK, Kent. Extreme Programming Explained: Embrace Change. 2nd Edition. Boston: Addison-Wesley, 2004.

BLAZEDS. Overview, 2008. Disponível em:
<http://opensource.adobe.com/wiki/display/blazeds/Overview>. Acesso em: 31/03/2009.

BOEHM, Barry. A View of 20th and 21st Century Software Engineering.|| ICSE06. Shanghai, China: ACM, 2006.

CAIRNGORM, Adobe. About Cairngorm, 2008. Disponível em: <http://opensource.adobe.com/wiki/display/cairngorm/About>. Acesso em: 05/04/2009.

COCKBURN, Alistair. What the Agile Toolbox Contains, 2004. Disponível em: <http://www.stsc.hill.af.mil/crossTalk/2004/11/0411Cockburn.html>. Acesso em: 29/09/2008.

CHOPRA, Chandan. Selecting the Right Flex Application Framework, 2009. Disponível em: <http://chandanchopra.blogspot.com/2009/03/selecting-right-flex-application.html>. Acesso em: 12/04/2009.

FIRE. Factory in Rapid Environment, 2007. Disponível em: <http://fire.cesar.edu.br/>. Acesso em: 05/04/2009.

FIRESCRUM. FireScrum ...the Open Source Scrum Tool, 2007. Disponível em: <http://www.firescrum.com/>. Acesso em: 16/05/2009.

FOWLER, Martin. The New Methodology, 2001. Disponível em: <http://www.martinfowler.com/articles/newMethodology.html>. Acesso em: 29/09/2008.

GLOGER, Boris. 5 min on Scrum | Tools, 2008. Disponível em: <http://scrum4you.wordpress.com/2008/06/03/5-min-on-scrum-tools/>. Acesso em: 29/03/2009

GLOGER, Boris. Scrum Tools | Firescrum | Not a Review, 2009. Disponível em: <http://borisgloger.com/2009/06/17/scrum-tools-firescrum-not-a-review/>. Acesso em: 05/07/2009

HOLFORD, David e Mehran EBRAHIMI. Honda: Approach to Innovation in Aerospace and Automotive/Pick-Up Truck Development: A Dialectical Yet Coherent Firm, 40th Annual Hawaii International Conference on System Sciences (HICSS-40), Big Island, Hawaii, 2007

INES. Inês adota FireScrum - Gerência de Projetos Ágil e Sistematizada, 2009. Disponível em: <http://www.ines.org.br/?p=21>. Acesso em: 05/07/2009

LARMAN, Craig. Agile and Iterative Development: A Manager's Guide. Addison-Wesley Professional, 2003.

LIKER, Jeffrey. **The Toyota way: 14 Management Principles from the World's Greatest Manufacturer**. New York: McGraw-Hill, 2004.

MOUNTAIN GOAT SOFTWARE. Task Board. Disponível em:
<http://www.mountaingoatsoftware.com/task-boards>. Acesso em: 23/02/2009.

POPPENDIECK, Mary e Tom. A History of Lean: From Manufacturing to Software Development, JAOO Conference, Aarhus, Denmark, 2005.

POPPENDIECK, Mary e Tom. Lean Software Development: An Agile Toolkit for Software Development Managers. Addison-Wesley Professional, 2003.

SCHWABER, Ken. **Agile Project Management with Scrum**. Redmond: Microsoft Press, 2004.

SCHWABER, Ken, BEEDLE, Mike. **Agile Software Development with Scrum**. New Jersey: Prentice Hall, 2002.

SCHWABER, Ken *et al.* Scrum Development Process, OOPSLA Business Object Design and Implementation Workshop, Eds. London: Springer, 1997.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Addison-Wesley, 2003.

SPIN-RECIFE. Uso do Scrum em Ambientes CMMI, 2008. Disponível em:
<http://spinrecife.wordpress.com/eventos-spinrecife/>. Acesso em: 05/04/2009.

Sprint-IT. Scrum Checklists, 2006. <http://www.infoq.com/minibooks/scrum-checklists>. Acesso em: 03/05/2009.

STANDISH, The Standish Group International. The CHAOS Report, 1994. Disponível em: http://www.standishgroup.com/sample_research/chaos_1994_1.php. Acesso em: 14/02/2009.

STANDISH, The Standish Group International. The CHAOS Report. Standish Group. 2004. Disponível em: <http://secure.standishgroup.com/reports/reports.php?rid=500>.

SUTHERLAND, Jeff. **Agile Development: Lessons Learned from the First Scrum**, Cutter Agile Project Management Advisory Service: Executive Update, vol. 5, pp. 1-4, 2004.

SUTHERLAND, Jeff, SCHWABER, Ken. The Scrum Papers: Nuts, Bolts, and Origins of an Agile Process, Draft 14/10/2007

TAKEUCHI, Hirotaka, NONAKA, Ikujiro. The New New Product Development Game. **Harvard Business Review**, 86116, 137-146 Janeiro-Fevereiro, 1986.

VERSIONONE, The State of Agile Developmen Survey Results, 2008, http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf. Acesso em: 29/03/2009.

WEBSTER, Steven, TANNER, Leon. Developing Flex RIAs with Cairngorm microarchitecture, 2008. Disponível em: http://www.adobe.com/devnet/flex/articles/cairngorm_pt1.html. Acesso em: 10/04/2009.

YAHOO GROUPS, scrumdevelopment - Scrum Users, 2000. Disponível em: <http://groups.yahoo.com/group/scrumdevelopment/>. Acesso em: 16/05/2009.

ZETIE, Carl. The Rise of Rich Internet Applications, Forrester Research, Inc. April 10, 2006.

ANEXO A – E-MAILS SOBRE O FIRESCRUM

“I was wondering when you’re software will be fully released. I’ve been looking the site over and reading the features of it and feel that it is a great program. I would like to be able to use it for our projects as soon as possible.” *Jackie Park-Cross, CSM - Covenant Eyes, Inc*

“Looking at the screenshots, firescrum seems to be what I’m looking for. When should we expect the first release?” *Martin Filteau*

“I am interested in FireScrum and I just wanted to ask you if you could send me a list of features for it. Also it would be nice if you could tell me which kind of platform and browser it is working with.” *Timm*

“Observamos a sua ferramenta (FireScrum) e ficamos bastante interessados. Poderia, por gentileza, nos passar informações sobre o seu software, bem como um link para que possamos baixá-lo e avaliá-lo?” *Fábio Trierveiler, CRE8 BRASIL / Quality Assurance*

“Parabéns pelo FireScrum! Trabalho no UOL e estamos começando a usar scrum na empresa. Achei o FireScrum muito legal!” *Paula Herculano*

“Gostei muito do que li sobre a ferramenta, pois parece que vocês estão conseguindo manter ela elegante, simples e funcional. Quando haverá uma versão para instalação?” *Aleckssandro*

“When will you have the version online? I want to write something about it, and I want to use it and give it to my customers.” *Boris Gloger Sprint-IT*

“I am the SrumMaster of a small commerical development team and am currently looking for a tool to manage our scrum process. I would like to evaluate FireScrum and want to know when you expect to release the first version? Thanks very much.” *Mike Reynolds*

“I'm interested in your product FireScrum. When do you think it will be released?” *Jim Loving*