

# Ferramenta Open-Source para Apoio ao Uso do Scrum por Equipes Distribuídas

Eric Cavalcanti<sup>34</sup>, Teresa M. de Medeiros Maciel<sup>1234</sup>, Jones Albuquerque<sup>134</sup>

<sup>1</sup>Universidade Federal Rural de Pernambuco – Recife, Brasil

<sup>2</sup>Universidade Federal de Pernambuco – Recife, Brasil

<sup>3</sup>C.E.S.A.R – Centro de Estudos e Sistemas Avançados do Recife – Recife, Brasil

<sup>4</sup>INES – Instituto Nacional para Engenharia de Software

eric@cesar.org.br, tmmm@cin.ufpe.br, joa@deinfo.ufrpe.br

**Abstract.** *This paper presents FireScrum, a project planning and management tool designed to support distributed Scrum teams. FireScrum is an open source tool developed based on Rich Internet Application (RIA) technology and can be especially useful across distributed teams. This paper comprises an overview of distributed Scrum teams, FireScrum architecture proposal, its functionalities and comparative analysis with other tools. Status of FireScrum and results of its validation is discussed.*

**Resumo.** *Este artigo apresenta o FireScrum, uma ferramenta de apoio ao planejamento e gerenciamento de projetos com Scrum, sendo especialmente útil para apoio a equipes distribuídas, sendo open-source e baseada em tecnologia Rich Internet Application (RIA). Este artigo compreende uma visão geral do ciclo Scrum, a proposta arquitetural do FireScrum e uma análise funcional comparativa com outras ferramentas. Resultados de validações realizadas são analisados.*

## 1. Contextualização

A adoção de abordagens ágeis tem sido crescente na indústria de software, apresentando resultados positivos em termos de prazo, custo e qualidade [Versionone 2008]. Tais abordagens visam à desburocratização das atividades [Schwaber 2004], e empregam princípios como ciclos iterativos e entregas rápidas de software funcionando [Beck, K at all 2001]. A essência desse movimento é calcado na agilidade, flexibilidade e habilidade de comunicação [Highsmith, J. 2004].

Dentre as metodologias ágeis propostas, o Scrum tem se apresentado como uma das mais adotadas, o se observa em pesquisas que apontam sua adoção por cerca de 70% entre empresas que usam metodologias ágeis [Versionone 2008].

Assim como outras metodologias ágeis, o Scrum enfatiza a interação do time através de comunicação constante e *face-to-face*. Por outro lado, desenvolvimento por equipes remotas tem apresentado um crescimento contínuo, particularmente visível desde a última década [Damian & Moitra 2006]. Entre as razões para a adoção por times distribuídos, pode-se ressaltar a redução de custos de locomoção, a facilidade de

envolvimento de especialistas em áreas-chave de, além da flexibilidade para criação e desativação de equipes de trabalho. Neste contexto, o modelo de Scrum de Scrum distribuído é recomendado pela ScrumAlliance ([www.scrumalliance.com](http://www.scrumalliance.com)), o qual divide o trabalho em partes dentro “times Scrum” isolados, eliminando a maioria das dependências entre os mesmos [Sutherland et al 2007]. Se ferramentas de gerenciamento tradicional de projetos podem comprometer o apelo visual e simplista defendido pela metodologia, o uso de artefatos não automatizados como cartões e murais, podem representar um desafio à adoção da metodologia por equipes remotas [Mountain 2009]. Neste contexto uma boa iniciativa pode ser o desenvolvimento de ferramentas de apoio ao Scrum em equipes distribuídas [Cristal 2008].

Inserido neste contexto, este artigo apresenta uma aplicação open-source, FireScrum, desenvolvida para facilitar a adoção do Scrum por times distribuídos contemplando apoio ao planejamento e controle de projetos, utilizando-se fortemente de design centrado no usuário e tecnologias RIA.2.

O FireScrum foi concebido como resultado de um trabalho de conclusão de curso e sua proposta aperfeiçoada pela dissertação de um mestrado profissionalizante. Com base na definição arquitetural e funcional, a aplicação foi implementada por cerca de 60 estudantes de pós-graduação, sendo atualmente é estudo de caso de 4 dissertações em andamento.

## 2. Metodologias Ágeis e Scrum

O termo “Metodologias Ágeis” tornou-se popular em 2001 quando especialistas em processos de software estabeleceram princípios comuns, sendo criada a Aliança Ágil e o estabelecido o “Manifesto Ágil” [Beck, K et al. 2001].

Alinhado aos princípios ágeis, 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 1986]. O primeiro desenvolvimento de software com Scrum foi realizado em 1993 por Jeff Sutherland na Easel Corporation [Sutherland 2004], e junto com Ken Schwaber formalizaram o Scrum [Sutherland 2007], como processo de desenvolvimento na OOPSLA [Schwaber 1997].

O Scrum compreende 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]. A proposta é baseada em um ciclo iterativo e incremental, onde cada iteração é planejada de acordo com a prioridade definida pelo cliente.

O Scrum implementa um esqueleto iterativo e incremental, através de três papéis principais [Schwaber 2004]. *Product Owner* (PO), que estabelece objetivos do produto, define e prioriza as funcionalidades, ou *itens de backlog* e participa ativamente do desenvolvimento, validando o produto de cada *sprint*; *Scrum Master* (SM), cujo papel é facilitar o trabalho do time, removendo os impedimentos levantados pelo time e apoiando o mesmo no uso do Scrum; *Time*, responsável pelo desenvolvimento dos *itens de backlog*, define como transformar o *product backlog* em incremento de funcionalidades, gerenciando seu próprio trabalho. São responsáveis coletivamente pelo sucesso da iteração e conseqüentemente pelo projeto como um todo.

## 2.1 O Ciclo Scrum

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 ser de duas a quatro semanas, conhecidas como *sprint*. Ao final de cada *sprint* é entregue um incremento do produto tendo passado por todo o processo de desenvolvimento, auditoria e teste.

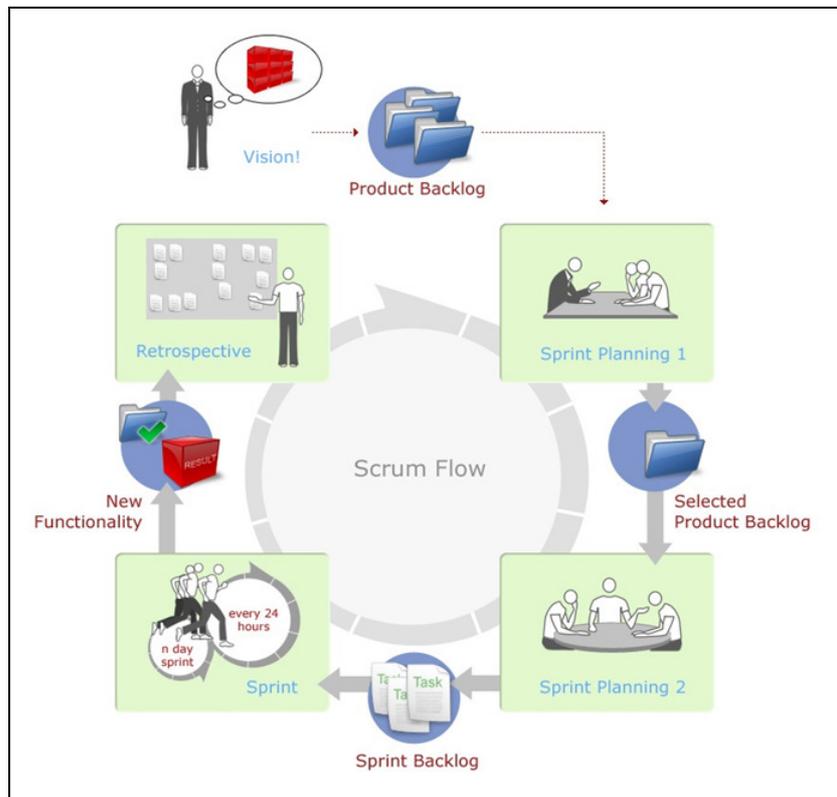


Figura 1. Visão geral do processo do Scrum

No início de cada *sprint* é realizada a reunião de planejamento (*Sprint Planning Meeting*), onde o Time colaborativamente com o PO define o objetivo da próxima *sprint* e os itens que serão desenvolvidos na mesma (*Selected Product Backlog*). Em seguida, o Time define as tarefas necessárias para a realização de cada item selecionado para a *sprint*. As tarefas não podem levar mais de 16 horas para ser executadas ou terem mais de 1 dia de duração. O próprio time escolhe quais tarefas desejam realizar, ou seja, tarefas nunca devem ser atribuídas. Diariamente é realizada uma reunião de revisão (*Scrum Daily Meeting*), conduzida pelo SM, com duração máxima de 15 minutos, estando todos de pé, para acompanhamento e identificação de impedimentos.

Ao término da execução da *sprint*, o Time reúne-se com o PO (*Sprint Review*) para demonstrar os resultados obtidos e para que o PO valide se os objetivos da *sprint* foram alcançados. Em seguida, o SM conduz a *Sprint Retrospective*, onde são levantados pontos de melhoria. 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* e o *Sprint Burndown*, que representam o quanto de funcionalidades foram entregues do produto e da *sprint* respectivamente na linha do tempo.

## 2.2 Scrum Distribuído

Assim como outras metodologias ágeis, o Scrum enfatiza a interação do time através de comunicação constante e *face-to-face*. Por outro lado, desenvolvimento por equipes remotas tem apresentado um crescimento contínuo, particularmente visível desde a última década [Damian & Moitra 2006]. Entre as razões para a adoção por times distribuídos, pode-se ressaltar a redução de custos de locomoção, a facilidade de envolvimento de especialistas em áreas-chave de, além da flexibilidade para criação e desativação de equipes de trabalho. Neste contexto, o modelo de Scrum de Scrum distribuído é recomendado pela ScrumAlliance ([www.scrumalliance.com](http://www.scrumalliance.com)), o qual divide o trabalho em partes dentro “times Scrum” isolados, eliminando a maioria das dependências entre os mesmos [Sutherland et al 2007]. Se ferramentas de gerenciamento tradicional de projetos podem comprometer o apelo visual e simplista defendido pela metodologia, o uso de artefatos não automatizados como cartões e murais, podem representar um desafio à adoção da metodologia por equipes remotas [Mountain 2009]. Neste contexto uma boa iniciativa pode ser o desenvolvimento de ferramentas de apoio ao Scrum em equipes distribuídas [Cristal 2008].

## 3. O FireScrum

O FireScrum é uma ferramenta que reúne um conjunto de aplicações integradas para suportar equipes que utilizam o Scrum como base para o desenvolvimento de seus projetos, sendo especialmente útil para equipes que trabalham remotamente.

A idéia do FireScrum foi concebida por um trabalho de dissertação de mestrado, motivada por lacunas existentes em propostas similares e pela demanda por ferramentas de apoio ao Scrum. Durante o desenvolvimento da dissertação, as funcionalidades definidas foram implementadas por cerca de 60 pós-graduandos da disciplina de Engenharia de Software na Universidade Federal de Pernambuco.

Os desafios enfrentados pelo Fire Scrum não foram poucos. O uso de post-its e murais exigia uma interface simples, mas muito próxima do real, o que de certa forma a tornava complexa de ser definida e implementada. A interação “*face-to-face*”, defendida pelas metodologias ágeis, demandava recursos multimídia para garantir interação entre o time mesmo que remoto. A arquitetura deveria ser modularizada para não criar dependências entre seus módulos. Além disso, a ferramenta deveria ser open-source.

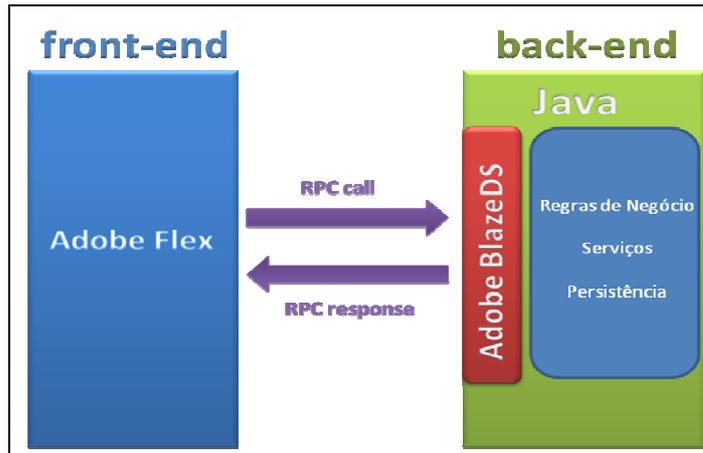
O FireScrum foi desenvolvido utilizando conceitos da Web 2.0 e de *Rich Internet Applications*, apresentando foco em usabilidade para garantir a simplicidade do seu uso. É uma aplicação Web, o que significa que a mesma é acessível remotamente através de um browser, podendo ser utilizada em um ambiente de Internet ou Intranet, apresentando uma arquitetura modular e extensível.

### 3.1 Arquitetura

A arquitetura 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 pela parte da aplicação responsável pela interface do usuário e seu comportamento. É acessível através de um browser com o Flash Player instalado. Este requisito se dá em função do uso do Adobe Flex, *framework* de código aberto que permite a criação e manutenção de *Rich Internet Applications*. O *back-end* é desenvolvido utilizando a plataforma Java que contém a regra de negócio, os serviços e o mecanismo de

persistência. 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 que possibilita a comunicação entre aplicações desenvolvidas em Adobe Flex/Air utilizando um protocolo binário denominado AMF(*Action Message Format*) [Adobe 2008].

Na Figura 2 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 2. Arquitetura Macro do FireScrum**

#### **Arquitetura do *front-end***

O Adobe Flex foi é um poderoso framework para desenvolvido 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 Web. Na arquitetura do *front-end* foi adotado o Cairngorm, que é uma micro-arquitetura para desenvolvimento de aplicações de internet rica para o Adobe Flex. O Cairngorm utiliza o padrão arquitetural MVC onde a separação de papéis e responsabilidades são bem evidentes[Cairngorm].

#### **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 pela existência de uma infraestrutura 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.

### **4. Funcionalidades do FireScrum**

As funcionalidades do FireScrum foram definidas com base em um trabalho de benchmarking onde foram consideradas

estão organizadas em módulos integrados, mas independentes entre si. O módulo Core é considerado o principal da ferramenta, sendo auto-suficiente para contemplar todo o ciclo de um projeto Scrum. Os demais são considerados módulos de apoio, excluindo a necessidade de ferramentas de terceiros.

A seguir são apresentadas as principais funcionalidades divididas em módulos.

#### 4.1 Core

Responsável por toda a parte operacional básica do Scrum. Dentre as funcionalidades, disponíveis destacam-se: controle de acesso, cadastro de usuários, cadastro de projeto criação de *itens de backlog*, priorização de *itens de backlog*, criação de *sprints*, associação de *itens de backlog* a *sprints*, criação de tarefas para um dado *item de backlog*, alocação do membro do time a determinada tarefa, geração do gráfico de *burndown* da *sprint* e gráfico de *burndown* do produto.

#### 4.2 TaskBoard

Reproduz o quadro físico ou mural utilizado em projetos reais baseados em Scrum. O intuito é permitir que, através do quadro, os usuários da aplicação possam manipular post-its virtuais de forma simples. Dentre as funcionalidades disponíveis no módulo TaskBoard destacam-se: visualizar os *itens de backlog* para a *sprint* selecionada, priorizar os itens de *backlog* da *sprint* selecionada, cadastrar tarefas (representadas por post-its virtuais) para um dado *item de backlog* informando um responsável e uma estimativa de esforço, mudar o estado das tarefas arrastando-as pelo quadro, cadastrar impedimentos, associar impedimentos a tarefas ou *itens de backlog*, efetuar zoom in ou zoom out no quadro, aplicar filtros visuais para destacar tarefas que possuam uma dada informação e visualizar o percentual de progresso da *sprint* corrente.



Figura 3 Tela do Módulo TaskBoard do FireScrum.

#### 4.3 Planning Poker

Viabiliza o uso remoto da técnica de estimativa Planning Poker. Este módulo conta com recursos multimídia para reuniões de estimativas utilizando comunicação via *chat*, vídeo ou texto. Dentre as funcionalidades disponíveis destacam-se: visualizar os itens do backlog a serem estimados, definir os valores das estimativas a serem utilizados, possibilidade de compartilhar áudio e vídeo dos participantes, chat entre os participantes e gerenciar o processo de estimativa entre os participantes.

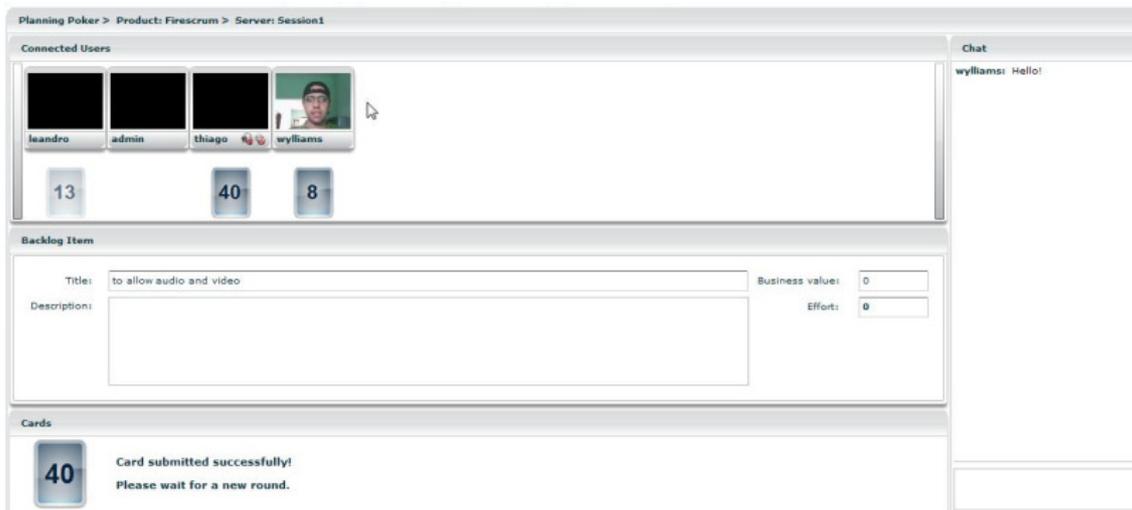


Figura 4 Tela do Módulo Planning Poker do FireScrum.

#### 4.4 Test Management

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: criação de plano de teste, criação de casos de teste, criação de job de execução de testes (Test Runner), associar casos de teste a *itens do backlog*, registro dos resultados dos testes, criar biblioteca de casos de testes e disponibilizar.

#### 4.5 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 necessidade de ferramentas externas para tal finalidade. Dentre as funcionalidades disponíveis destacam-se: registrar bug, associar um item de backlog a um bug, definir um responsável e adicionar notas ao bug, anexar arquivos ao bug, enviar e-mails para envolvidos com o bug, manter histórico e gerar relatórios com filtros sobre bugs.

#### 4.6 Desktop Agent

Facilita 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 destacam-se: visualizar tarefas e tarefas do usuário, editar tarefas, receber alertas do sistema, reportar bugs e fornecer um chat entre os integrantes do time.

### 5. Diferenciais do FireScrum sobre Ferramentas Similares

O FireScrum atualmente se encontra em teste de validação, disponível para um número restrito de avaliadores voluntários, os quais estão sugerindo melhorias emergenciais antes que a ferramenta seja disponibilizada para download. O trabalho de ajustes e validação deve prosseguir até setembro deste ano, quando a ferramenta deverá ser disponibilizada para a comunidade global. No entanto, o código fonte da aplicação se encontra estável, disponível para download a partir do site da ferramenta.

Além do design centrado no usuário, baseado em recursos multimídia, RIA e Adobe Flex, o qual fornece uma interface próxima dos artefatos manuais recomendados pelo

Scru, o FireScrum é uma ferramenta *freeware* e *open source*, favorecendo especialmente pequenas e médias empresas que não podem onerar a adoção de práticas ágeis de desenvolvimento de software com custo adicional de ferramentas.

### 5.1 Motivação global pelo FireScrum

O interesse e aceitação do FireScrum pode ser observado através do número de acessos ao site utilizando soluções como o Google Analytics (<http://www.google.com/analytics>). Como indicador do grau de abrangência da ferramenta, pode-se atestado pela distribuição geográfica dos acessos ao site, conforme apresentado na Figura 5.



**Figura 5 Informações de acesso providas pelo Google Analytics.**

O mapa apresentado na Figura 5 apresenta o número e distribuição do acesso ao site desde sua publicação, no início de sua concepção em 2008. Embora comparado a outras ferramentas já consolidadas nesta área como ScrumWorks, o número de acesso possa ser considerado baixo, o mesmo se torna relevante considerando que o FireScrum é uma iniciativa desenvolvida por trabalho voluntário de estudantes da Engenharia de Software. A relevância dos dados intensifica-se se observando a abrangência geográfica de acessos ao site, provenientes de 97 localidades (vide área verde do mapa).

### 5.2 Funcionalidades do FireScrum versus ferramentas similares

Para se definir a arquitetura funcional do FireScrum, uma análise de algumas das ferramentas mais conhecidas na comunidade foi realizada, para se identificar lacunas e diferenciais potenciais. 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.

A Tabela 1 apresenta um mapeamento de funcionalidades relevantes entres as ferramentas analisadas e as funcionalidades implementadas pelo FireScrum.

**Tabela 1. Análise comparativa de funcionalidades**

FUNCIONALIDADES	VERSIONONE	TARGETPROCESS	SCRUMWORKS	FIRESCRUM	AGILO 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	-

Analisando a Tabela 1, é possível observar que o FireScrum atende a maioria das funcionalidades fornecidas pelas demais ferramentas conjuntamente. Desktop Agent e o Planning Poker, focado no suporte ao planejamento de *sprints* incluindo recursos multimídia para intensificar a interação do time.

## 6. Considerações Finais

Este artigo apresentou o FireScrum, uma aplicação open-source que reúne um conjunto de aplicações para suportar o planejamento e gerenciamento de projetos com Scrum, sendo útil especialmente para equipes distribuídas.

Como trabalhos futuros, estão planejadas otimizações da ferramenta a partir dos resultados dos testes pela comunidade, além do desenvolvimento de funcionalidades

para apoio às reuniões de Sprint Review e Sprint Retrospective fechando assim todo o ciclo Scrum.

## Referências

- Adobe Systems Incorporated (2008) “BlazeDS-Overview”. Disponível em: <http://opensource.adobe.com/wiki/display/blazeds/Overview>. Acesso em: 31/03/2009.
- Adobe Systems Incorporated (2008) “About Cairngorm”. Disponível em: <http://opensource.adobe.com/wiki/display/cairngorm/About>. Acesso em: 05/04/2009.
- Beck, K at all. (2001) “Manifesto for Agile Software Development”. Disponível em: <http://www.agilemanifesto.org>. Acesso em: 2/04/2008.
- Cristal M., Wildt D., Prikladnicki, R. (2008) “Usage of Scrum Practices Within a global Company”, "Usage of SCRUM Practices within a Global Company", 2008 IEEE International Conference on Global Software Engineering, pp. 222-226.
- Damian, D., Moitra, D. (2006). “Guest Editors' Introduction: Global Software Development: How far Have We Come?”, IEEE Software, 23(5), pp.17-19.
- Highsmith, J. (2004) “Agile Project Management – Creating Innovative Products”, AddisonWesley.
- Schwaber, K. et al. (1997) “Scrum Development Process”, OOPSLA Business Object Design and Implementation Workshop, Eds. London: Springer.
- Schwaber, K. and Beedle, M. (2002) “Agile Software Development With Scrum”. NJ: Prentence Hall.
- Schwaber, K. (2004) “Agile Project Management With Scrum”, Redmond: Microsoft Press.
- Sutherland, J. (2004) “Agile Development: Lessons Learned from the First Scrum”, Cutter Agile Project Management Advisory Service: Executive Update, vol. 5, pp. 1-4.
- Sutherland J., Viktorov A., Blount J., Puntikov N. (2007) "Distributed Scrum: Agile Project Management with Outsourced Development Teams," HICSS'40, Hawaii International Conference on Software Systems, Big Island, Hawaii, 2007.
- Versionone (2008) “The State of Agile Development Survey Results”, [http://www.versionone.com/pdf/3rdAnnualStateOfAgile\\_FullDataReport.pdf](http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf). Acesso em: 29/03/2009.
- Takeuchi, H., Nonaka, I. (1986) “The New New Product Development Game”. Harvard Business Review, 86116, Janeiro-Fevereiro, pp137-146.