



**C.E.S.A.R.EDU**

Unidade de Educação do Centro de Estudos e Sistemas Avançados do Recife

## **Projeto de Dissertação de Mestrado**

# **FireScrum: Ferramenta de apoio à gestão de projetos utilizando Scrum**

**Eric de Oliveira Cavalcanti**

**Orientador:  
Jones Albuquerque, CESAR.Edu**

**Abril, 2008**

**Mestrado Profissional em Engenharia de Software  
Centro de Estudos e Sistemas Avançados do Recife (CESAR)**



## Sumário

<b>1. INTRODUÇÃO E MOTIVAÇÃO .....</b>	<b><u>45</u></b>
<b>2. OBJETIVOS .....</b>	<b><u>57</u></b>
<b>3. METODOLOGIA .....</b>	<b><u>68</u></b>
3.1 ATIVIDADES PREVISTAS .....	<u>68</u>
3.2 CRONOGRAMA DE ATIVIDADES .....	<u>68</u>
<b>4. RESULTADOS PARCIAIS .....</b>	<b><u>79</u></b>
4.1 FUNDAMENTAÇÃO TEÓRICA / ESTADO-DA-ARTE .....	<u>79</u>
4.1.1 <i>Papéis e Responsabilidades</i> .....	<u>79</u>
4.1.2 <i>Ciclo de Vida</i> .....	<u>84</u>
4.2 TRABALHOS RELACIONADOS .....	<u>104</u>
<b>5. CONTRIBUIÇÕES ESPERADAS .....</b>	<b><u>104</u></b>
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b><u>124</u></b>



## Declaração

Eu declaro que este projeto de dissertação foi elaborado por mim, e que o trabalho contido aqui é de minha autoria, exceto quando explicitamente afirmado no texto, e que este trabalho não foi submetido para nenhum outro grau ou qualificação profissional.

---

Eric de Oliveira Cavalcanti

---



## 1. Introdução e Motivação

---

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 [1].

Acreditando que o processo utilizado é um dos motivos para a ocorrência desses problemas, um segmento crescente da Engenharia de Software vem defendendo a adoção de processos mais simplificados conhecidos como métodos ágeis, que visam à desburocratização das atividades associadas ao desenvolvimento [3].

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 [6]. 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 à identificação e correção de deficiências e/ou impedimentos no desenvolvimento [9].

O Scrum sugere a utilização de artefatos simples, muitas vezes construídos através de planilhas e quadro de tarefas baseados em post-its, com o objetivo principal de não criar um overhead de trabalho nas informações necessárias para o acompanhamento e fornecer uma visibilidade do status das iterações e do projeto como um todo.

No entanto, podemos considerar a dificuldade de se trabalhar com Scrum usando um ambiente de times remotos, onde a presença física e reuniões face a face que são uma das características dos métodos ágeis não são possíveis, e muito menos a visualização dos quadros de tarefas com *post-its*. Por utilizar planilhas, muitas vezes o processo de consolidação de métricas e histórico passa a ser custoso.

O FireScrum é uma ferramenta *open source* proposta para servir de suporte ao Scrum, proporcionando seu uso por equipes remotas, evitando a perda de histórico, propiciando o levantamento de métricas sem demandar esforço excessivo do time e principalmente facilitando sua adoção sem burocratizar o processo. Esta Dissertação apresenta o FireScrum, sua definição, arquitetura e um protótipo como estudo-de-caso.



## 2. Objetivos

---

A dissertação tem como objetivo apresentar o Scrum como um framework para gerenciamento de projetos ágeis e propor uma ferramenta para suportá-lo, suprimindo necessidades como apoio a equipes remotas, obtenção de métricas e históricos.

O conceito de aplicações RIA (Rich Internet Application) será introduzido, assim como, será apresentada a arquitetura, tecnologias, frameworks e padrões utilizados no desenvolvimento da ferramenta.



### 3. Metodologia

---

A seguir serão apresentadas as atividades e o cronograma para a conclusão da dissertação.

#### 3.1 Atividades Previstas

O desenvolvimento do projeto será dividido em atividades, compreendendo:

- **Revisão Bibliográfica Básica**  
Pesquisar nas diversas fontes da literatura material relacionado ao tema proposto para verificar o estado da arte e adquirir embasamento para o desenvolvimento do projeto. Esta atividade foi iniciada como parte da disciplina de Fábrica de Software.
- **Análise de ferramentas**  
Analisar as ferramentas de suporte ao Scrum existentes e identificar possíveis diferenciais a serem implementados. Esta atividade foi iniciada como parte da disciplina de Fábrica de Software.
- **Proposta do Modelo**  
Elaborar uma proposta da ferramenta a ser desenvolvida, com base nas pesquisas, análises e oportunidades. Esta atividade foi iniciada como parte da disciplina de Fábrica de Software.
- **Implementação**  
Implementação do software para validar o modelo proposto
- **Escrita da dissertação**  
Descrever a fundamentação teórica, a tecnologia utilizada e os resultados durante o processo de desenvolvimento.

#### 3.2 Cronograma de Atividades

Atividade	Março	Abril	Mai	Junho	Julho	Agosto
Pesquisar	X	X	X	X	X	
Análise de Ferramentas	X					
Proposta do Modelo	X					
Implementação	X	X	X	X	X	
Escrita da dissertação	X	X	X	X	X	X
Seminário 1			X			
Seminário 2					X	
Defesa da dissertação						X



## 4. Resultados Parciais

---

Durante a disciplina de Fábrica de Software foi realizado um levantamento de requisitos e problemas encontrados na utilização do framework SCRUM a partir de entrevistas com gerentes de projetos e profissionais do C.E.S.A.R que utilizam o SCRUM em seus projetos, bem como foram realizados estudos baseados na literatura, pesquisas nos fóruns especializados em SCRUM e análise de ferramentas já existentes.

### 4.1 Fundamentação Teórica / Estado-da-Arte

O Scrum surge dentro das metodologias ágeis como um framework para gerenciamento de projetos, sendo fundamentado na produção de softwares 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 [2], muitas pessoas responsáveis por gerenciar projetos são educadas a usar uma abordagem determinística que utilizam planos detalhados com gráficos de Gantt e schedules de trabalho. Sendo o Scrum 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 e 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.

#### 4.1.1 Papéis e Responsabilidades

O Scrum implementa um esqueleto iterativo e incremental através de três papéis principais [2]:

- Product Owner - define as funcionalidades do produto, os itens do backlog do produto, o valor de negócio para cada item do backlog (ROI), prioriza os itens do Backlog do produto, 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.
- 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.

- Time - 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 conseqüentemente pelo projeto como um todo [1].

#### 4.1.2 Ciclo de Vida

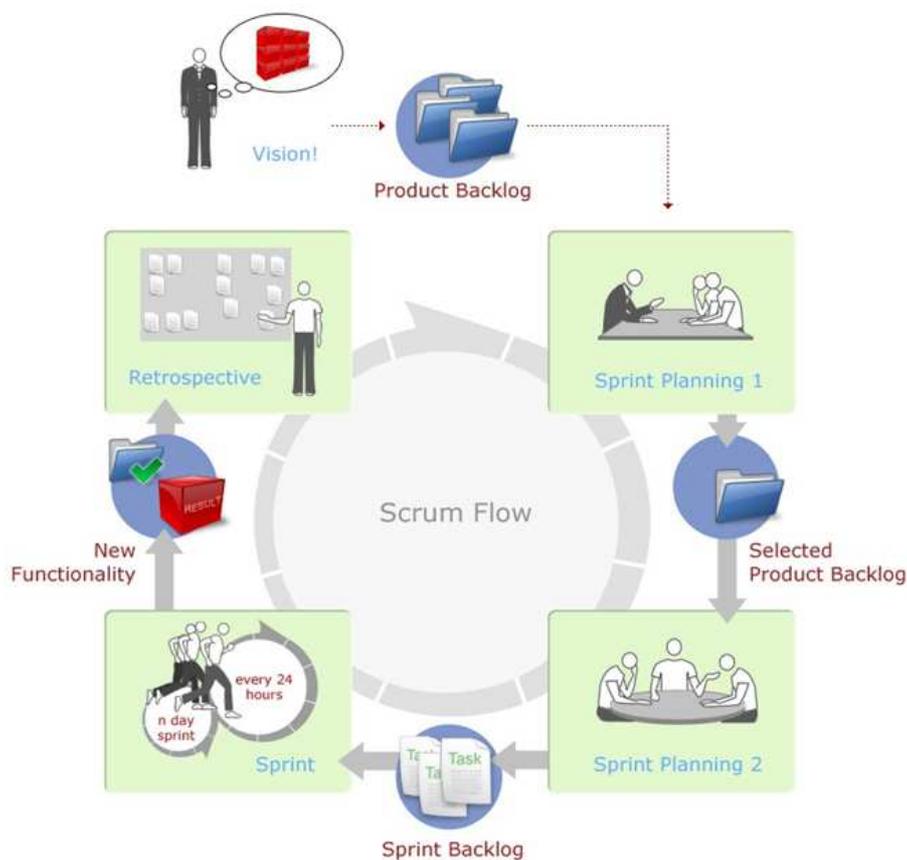


Figura 1. Visão geral do processo do Scrum

No Scrum, um projeto se inicia com uma visão do produto que será desenvolvido [2]. 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 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 1.

Antes do início da Sprint é realizada a reunião de planejamento (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 (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 está última uma breve declaração que ilustre o foco do trabalho durante a Sprint. Na segunda parte da reunião (Sprint Planning 2), 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, 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 1 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 farão, 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. Deve ser realizada diariamente uma reunião de revisão (Scrum Daily Meeting), 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 fizeste ontem? O que vais fazer hoje? Há algum impedimento? 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. Deve ser uma apresentação 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 (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 e o Sprint Burndown.

Segundo Fowler[4], ferramentas que impõe uma menor estrutura devem ser usadas, uma vez que elas se encaixam melhor no que a equipe deseja fazer. Com isso é possível concluir que é necessário a criação de novas ferramentas para suportar o modelo de gestão de projetos do Scrum, que forneça atividades e artefatos necessários para a execução do processo, trazendo uma interface com usuário eficiente, e que se aproxime ao máximo da simplicidade do Scrum quando utilizado com planilhas e *post-its*.

## 4.2 Trabalhos Relacionados

No mercado atual existem algumas ferramentas comerciais para Scrum, onde podemos destacar as seguintes com seus respectivos pontos fortes e fracos:

Ferramenta	Pontos Fortes	Pontos Fracos
Version One	Engloba várias metodologias ágeis: Scrum, Extreme Programming, DSDM, Agile UP.	Muitos recursos, tornando-se complexa
Rally Software ScrumWorks	- Interface Simples	Complexa, Usabilidade Necessita Run-Time Java, Arquitetura fechada, sem possibilidades de customização

## 5. Contribuições Esperadas

A contribuição esperada é a de melhorar a eficácia de projetos de construção de software, baseado em metodologias ágeis apoiados pela ferramenta FireScrum, onde serão também apresentadas justificativas para o modelo



arquitetural proposto com o objetivo de introduzir conceitos como RIA e um estudo de caso de sua aplicação.

**[joa1] Comentário:** Frase longa. E contribuições simplórias...



## 6. Referências Bibliográficas

---

- [1] Sommerville, I. “Engenharia de software”. São Paulo: Addison-Wesley, 2003.
- [2] Schwaber, K. “Agile Project Management with Scrum”, Microsoft, 2004.
- [3] Fowler, M., 2001 “The New Methodology”. <http://www.martinfowler.com/articles/newMethodology.html> (Abril 2008).
- [4] Fowler, M., 2006 “Using an Agile Software Process with Offshore Development” <http://martinfowler.com/articles/agileOffshore.html> (Abril 2008)..
- [5] AgileManifesto, 2001, “Manifesto for Agile Software Development” <http://agilemanifesto.org/> (Abril 2008).
- [6] ControlCaos, 1996, “Controlled chaos: living on the edge”, <http://www.controlchaos.com/old-site/ap.htm>, (Abril 2008).
- [7] Marçal, A., Freitas, B., Soares, F., Belchior, A., Maciel, Teresa (2007), “Uma abordagem do SCRUM a partir das Áreas de Processo de Gerenciamento de Projetos do CMMI”, 2007.
- [8] Marçal, A., Freitas, B., Soares, F., Belchior, A. (2007), “Mapping CMMI Project Management Process Areas to SCRUM Practices”, 31st Annual Software Engineering Workshop, Loyola College, Baltimore, MD, USA, 6-8 March 2007.
- [9] Schwaber, K. and Beedle, M. (2002), “Agile Software Development With Scrum”. NJ: Prentence Hall.

[Joa2] Comentário: Precisa de mais embasamento científico.