



**V SINGEP**

**Simposio Internacional de Gest3o de Projetos, Inova3o e Sustentabilidade**  
**International Symposium on Project Management, Innovation and Sustainability**

ISSN: 2317 - 8302

## **Aplicando SCRUM em Projetos de Desenvolvimento de Softwares em uma Institui3o Financeira**

**ANDRE BUENO LIMA**

Universidade Cat3lica de Bras3lia  
andre@openti.biz

Traditional methods of project management have presented many difficulties in projects with high complexity and low predictability. The SCRUM framework is one of the agile methods developed to work with complex and adaptive problems, delivering products with the highest possible value. The conciseness of this method is the removal of obstacles and periodical deliveries, through empirical process control. In the face of wide difficulties in the traditional waterfall models, with the adoption of SCRUM framework, the software development team of financial institution studied in this research obtained an increase of 75% of project performance over the traditional method adopted so far.



**V SINGEP**

Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade

International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317 - 8302

## **APLICANDO SCRUM EM PROJETOS DE DESENVOLVIMENTO DE SOFTWARES EM UMA INSTITUIÇÃO FINANCEIRA**

### **Resumo**

Os métodos tradicionais de gestão de projetos apresentaram muitas dificuldades em projetos com alta complexidade e baixa previsibilidade. O framework SCRUM é um dos métodos ágeis desenvolvido para trabalhar com problemas complexos e adaptativos, entregando produtos com o maior valor possível. A concisão deste método está na remoção de obstáculos para a efetivação de entregas periódicas, através do controle de processos empíricos. Diante das grandes dificuldades encontradas nos modelos tradicionais, com a adoção do framework SCRUM, a equipe de desenvolvimento de softwares da instituição financeira estudada nesta pesquisa obteve um aumento de 75% do desempenho do projeto em relação ao método tradicional adotado até então.

**Palavras-chave:** SCRUM, Banco, Método Ágil, Agilidade

### **Abstract**

Traditional methods of project management have presented many difficulties in projects with high complexity and low predictability. The SCRUM framework is one of the agile methods developed to work with complex and adaptive problems, delivering products with the highest possible value. The conciseness of this method is the removal of obstacles and periodical deliveries, through empirical process control. In the face of wide difficulties in the traditional waterfall models, with the adoption of SCRUM framework, the software development team of financial institution studied in this research obtained an increase of 75% of project performance over the traditional method adopted so far.

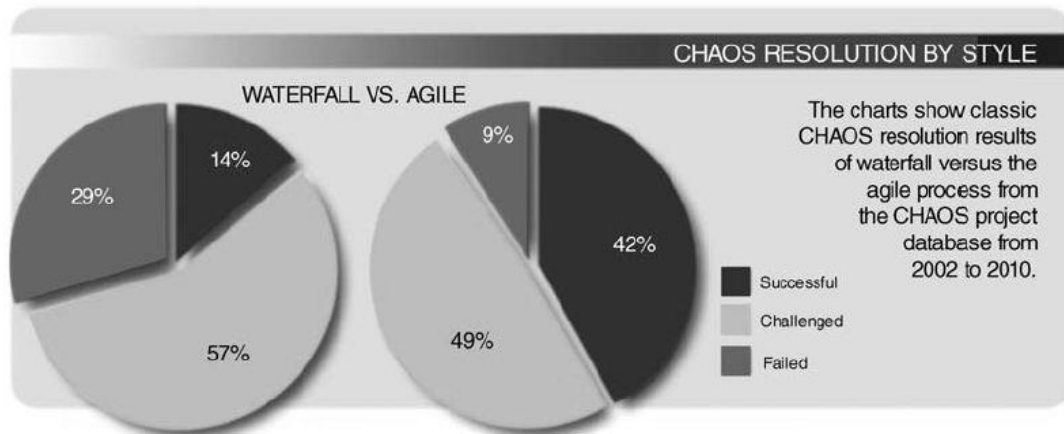
**Keywords:** SCRUM, Bank, Agile Method



## 1 Introdução

O desenvolvimento de softwares vem evoluindo em tecnologias, metodologias e processos nos últimos tempos, a fim de que um software atenda as expectativas daquele que o solicitou, com qualidade funcional e documental. Muito já foi discutido sobre a elaboração de processos advindos do desenvolvimento em cascata (Waterfall), com a ajuda de práticas originárias do PMBOK<sup>1</sup>. Porém muitas dificuldades são encontradas enquanto o mercado absorve este modelo de desenvolvimento, como:

- Métodos pesados (ou peso pesados) que tornavam o desenvolvimento de software moroso e complexo;
- Alto índice de fracasso dos projetos da área de TI;
- Distanciamento dos interessados, ou seja, as metodologias de desenvolvimento mantinham os interessados distantes do processo de desenvolvimento de software;
- Paradigma comando e controle (gerência e equipe). Essa abordagem torna a equipe apenas uma executora de tarefas.



**Figura 1. Agile x Waterfall.**

Fonte: Schwaber, K; Sutherland, J. 2012

Isto posto, formula-se a seguinte pergunta da pesquisa: Como o Banco Alfa Instituição Financeira pode suprimir as dificuldades do desenvolvimento de softwares impostas pelo modelo de gerenciamento de projetos tradicional?

Desta forma, esta pesquisa visa observar a implementação e os resultados advindos de um dos métodos ágeis mais utilizados no mercado, o SCRUM, em uma das unidades estratégicas do Banco Alfa Instituição Financeira.

## 2 Referencial Teórico

Entre os mais diversos obstáculos em projetos de TI destacamos: a comunicação ineficaz entre o cliente e os desenvolvedores de softwares; a quantidade de horas alocadas em retrabalho decorrente da ausência de compreensão das necessidades e expectativas envolvidas no projeto; escopo mal formatado; alta complexidade da manutenção de softwares desenvolvidos por diferentes programadores; e, no caso do Banco Alfa Instituição Financeira,

<sup>1</sup> O PMBOK fornece aos gerentes de projetos com as práticas fundamentais necessários para alcançar os resultados organizacionais e excelência na prática de gestão de projetos.



uma grande quantidade de intervenientes que possuem visões diferentes para o mesmo produto.

O Manifesto Ágil declara os valores fundamentais que o desenvolvimento ágil de software deve-se fundamentar:

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

Uma das principais características dos métodos ágeis é a aceitação de mudanças. É comum que requisitos de software venham a sofrer mudanças devido a alteração de cenários mercadológicos e econômicos. Os processos adotados nos métodos ágeis permitem alta adaptabilidade à mudanças de cenários. Outros pontos fortes dos modelos ágeis são:

- Novas lições e novos conhecimentos são adquiridos com os insucessos do projeto, gerando novos modelos adaptativos aos obstáculos encontrados, que os evitarão no futuro;
- A divisão de grandes projetos em partes menores e funcionais;
- A entrega continuada de software funcional garante a satisfação do cliente.

## 2.1 SCRUM

Segundo o SCRUM Guide (2013), o SCRUM é um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos e entregar, produtivamente e criativamente, produtos com o mais alto valor possível. Este framework é baseado em times SCRUM, que possuem papéis, artefatos e eventos que servirão à propósitos específicos e essenciais para o sucesso de um projeto ágil. Ele possui processos iterativos empíricos e incrementais para o desenvolvimento de qualquer produto e gerenciamento de qualquer projeto, onde a filosofia de trabalho presente envolve a atitude das pessoas envolvidas no projeto. A concisão deste método está na remoção de obstáculos para a efetivação de entregas periódicas, através de três pilares:

- **Transparência:** Toda informações e alterações que impactam os resultados dos processos devem estar disponíveis para todos os responsáveis pelos resultados.
- **Inspecção:** O monitoramento deve estar presente sobre os artefatos SCRUM para a detecção de variações das atividades e tarefas inerentes ao projeto.
- **Adaptação:** Através da inspecção pode-se verificar se um processo extrapolou os limites aceitáveis, evidenciando a necessidade de ajustes necessários à entrega adequada do resultado esperado.

O Time SCRUM é definido por pessoas que exercem os seguintes papéis: SCRUM Master, Product Owner e Time de Desenvolvimento. Os Times SCRUM devem ser auto organizáveis o suficiente para atuarem com eficiência em sua produtividade e flexibilidade.

O SCRUM Master, antes de atuar como um gestor, ele atua como líder servidor para o time, removendo impedimentos que poderão surgir durante o desenvolvimento das tarefas pelo time. Ele treina a equipe na adoção de SCRUM e outras práticas necessárias durante o projeto, e não gerencia o time, pois este deve ser auto gerenciável.

O Product Owner atua como gestor do Product Backlog (requisitos), além de tomar as medidas necessárias para evidenciar o valor do produto, e do trabalho realizado pelo Time de Desenvolvimento. Uma de suas funções é garantir que os objetivos a serem atingidos com o produto estejam claros à todos os envolvidos. Assim como o SCRUM Master, o Product Owner é uma pessoa e não um grupo.



O Time de Desenvolvimento é composto pelos responsáveis por desenvolver funcionalidades potencialmente entregáveis. Apesar de um time ser composto por pessoas com especialidades distintas, o ideal é que estas saibam pegar um requisito e transformá-lo em um produto utilizável, mesmo que tenham que atuar em áreas diferentes das que estão acostumados. Times pequenos favorecem que as interações ocorram aderentes aos objetivos do projetos, minimizando a perda do foco da visão do produto final. Times grandes aumentam a complexidade de um processo empírico, dificultando o gerenciamento de todo o Time SCRUM. Sabendo disto, o SCRUM Guide recomenda que um Time de Desenvolvimento deva possuir no mínimo 3 integrantes e no máximo 9 integrantes.

### 2.1.1 Eventos SCRUM

Os eventos no SCRUM são definidos a fim de implementar uma rotina que dispense a alocação de reuniões não definidas. Outro conceito utilizado nos eventos que colaboram na manutenção desta rotina é o time-boxed, que diz que os eventos devem possuir sua duração máxima fixada, que não deve ser reduzida ou aumentada. Cada um dos eventos foram especificados para permitir a transparência e a inspeção contínua dos mesmos.

#### 2.1.1.1 Sprint

A Sprint é um time-boxed de no máximo um mês, onde uma versão ou incremento utilizável do produto é desenvolvido.

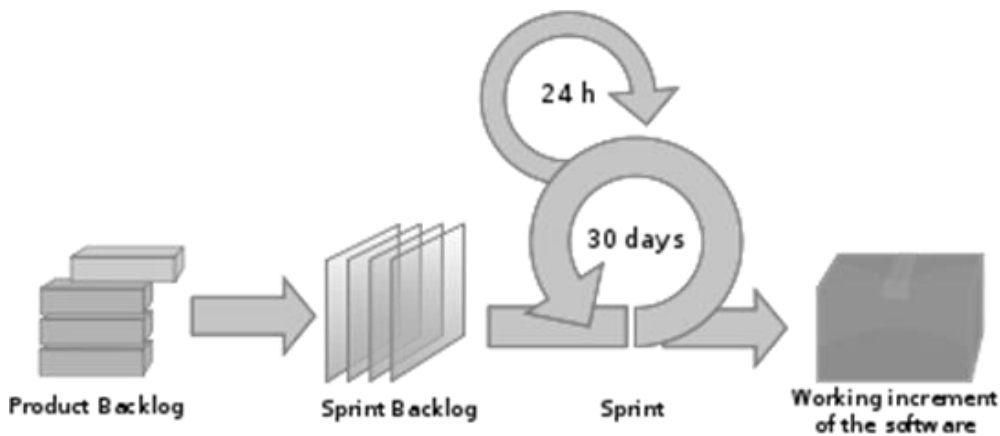


Figura 2. Fluxo da Sprint.

Fonte: T2Ti. 2013

#### 2.1.1.2 Sprint Planning Meeting

A reunião de planejamento da Sprint é quando a iteração é planejada. Essa reunião é facilitada pelo SCRUM Master. O PO apresenta os itens do Product Backlog de maior valor e o time irá estimar o tamanho/complexidade desses itens. Então é decidido quais destes serão feitos na Sprint, constituindo assim a Sprint Backlog.

O SCRUM Guide recomenda um time-boxed máximo de 8 horas para um Sprint de 30 dias. Esta reunião responde aos seguintes questionamentos:

- Qual o objetivo da Sprint?
- Qual será o resultado entregue ao final desta Sprint?
- Quais os trabalhos necessários (tarefas) para a concretização do resultado esperado desta Sprint?





- O que é considerado um incremento pronto?

### 2.1.1.3 Daily SCRUM Meeting

Na reunião diária do SCRUM o time se encontra para uma reunião em pé de no máximo 15 minutos, onde é discutido:

- O que fizemos ontem?
- O que iremos realizar de hoje para amanhã?
- Existe algum impedimento?

O Scrum Master deve garantir que as reuniões sejam realizadas. O objetivo desta reunião não é reportar, mas verificar o progresso do time em relação ao objetivo da Sprint, além de identificar impedimentos à plena execução das atividades.

### 2.1.1.4 Sprint Review Meeting

A reunião de revisão da Sprint é realizada no final da Sprint, onde será apresentada ao PO, e demais interessados, o que foi realizado e concluído durante a Sprint. Aqui responde-se a questionamentos que poderão surgir nos mais diversos níveis, além de planejar passos futuros. Este evento é uma time-boxed de no máximo 4 horas para uma Sprint de um mês, e durante sua execução pode-se adaptar o Product Backlog se necessário.

### 2.1.1.5 Sprint Retrospective Meeting

A reunião de retrospectiva do Sprint é realizada antes do próximo planejamento de Sprint, e ao final da Sprint Review Meeting. Ela tem o objetivo de revisar os métodos adotados, de modo a torná-los mais eficazes e gratificantes. Assim, todo o time SCRUM deve inspecionar a si mesmo e elaborar planos de melhorias para as próximas Sprints.

## 2.1.2 Artefatos SCRUM

Os Artefatos SCRUM são os trabalhos ou valores utilizados para impressão da transparência, inspeção e adaptação no framework SCRUM.

### 2.1.2.1 Product Backlog

Este artefato reuni a lista de requisitos de forma priorizada realizada pelo PO. Uma das ferramentas utilizadas para elaborá-lo é a história do usuário (user story). A priorização dos itens são definidos por uma nota numérica de acordo com o PO, e este deve garantir que todos os itens estejam disponíveis ao Time SCRUM. Uma técnica muito comum utilizada para a priorização dos itens é a MoSCoW.

O Product Backlog não necessita de que todos os requisitos estejam presentes. Somente os requisitos inicialmente conhecidos e suficientes para que a primeira Sprint seja possível. Os demais requisitos evoluirão de acordo com o andamento do projeto.

A qualquer momento a evolução do trabalho em relação ao Product Backlog pode ser monitorado. Para isto, pode-se somar todo o trabalho restante em pelo menos a cada Sprint Review Meeting, onde o Product Owner comparará o trabalho realizado até então e verificará a evolução em direção ao objetivo previsto. Um dos gráficos preferenciais neste tipo de inspeção é o gráfico de burndown.

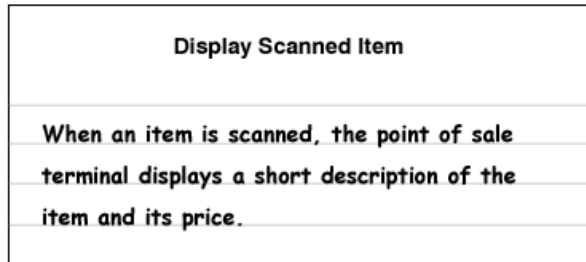


Figura 3. User story.

Fonte: DeveloperWorks. 2013

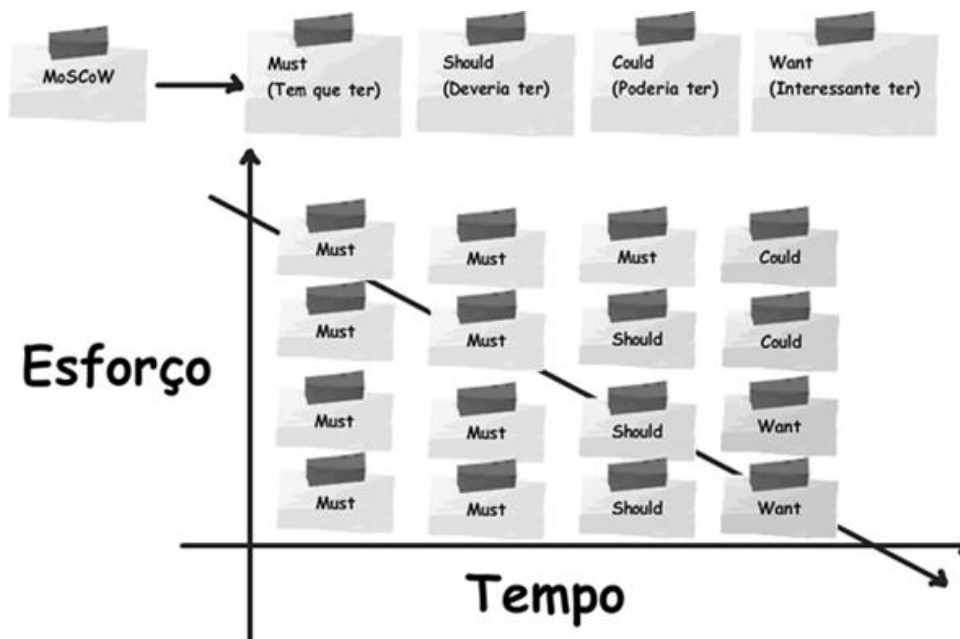


Figura 4. Fluxo da Gráfico de MoSCoW.

Fonte: carlostristacci.com.br. 2013

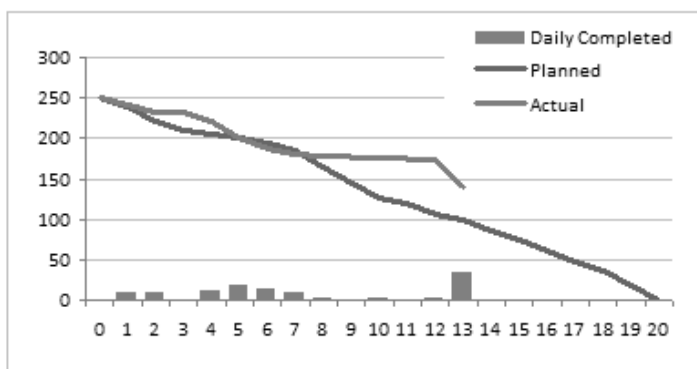


Figura 5. Gráfico Burndown.

Fonte: Chandoo.org. 2013

### 2.1.2.2 Sprint Backlog

Este artefato é composto pelos itens do Product Backlog que foram selecionados para a Sprint a ser imediatamente executada, concomitante ao plano de execução do(s) incremento(s) propostos para a mesma. Aqui, cada um dos itens selecionados podem ser



decompostos em uma ou mais tarefas, cada uma com sua própria estimativa de complexidade, pontos e/ou carga de trabalho. A manutenção do Sprint Backlog é exclusividade do Time de Desenvolvimento. O monitoramento também pode ocorrer a qualquer tempo da Sprint, porém o Time de Desenvolvimento deve verificar o total de trabalho restante a cada Daily SCRUM Meeting.

### 2.1.2.3 Incremental

A totalidade de todos os itens do Product Backlog completados e entregues durante a Sprint, em conjunto com os valores entregues nas Sprints anteriores, resulta no incremento do produto desenvolvido. Este incremento deve obedecer o conceito de pronto definido no Planning Sprint Meeting. Portanto o incremento ele deve ser utilizável, independente do Product Owner liberá-lo para produção ou utilização.

## 3 Metodologia de Pesquisa

Um método é um conjunto de processos pelos quais se torna possível conhecer uma determinada realidade, produzir determinado objeto ou desenvolver certos procedimentos ou comportamentos (OLIVEIRA, 1999). O método de pesquisa utilizado nesta pesquisa visa à exploração teórica, bibliográfica e qualitativa através de um estudo de caso.

Pela abordagem de Martins (2000, p.26) é denominado um estudo fenomenológico-hermenêutico aquele que utiliza análises teóricas e análise de documentos e textos, cujas propostas são críticas e buscam a conscientização dos envolvidos na pesquisa.

A pesquisa qualitativa foi utilizada para avaliar a evolução empírica de um projeto SCRUM, em um ambiente de desenvolvimento de software fortemente influenciado pelo métodos tradicionais de gestão de projetos (Waterfall). Segundo CESAR, A. M. R. V. C (2013), "pesquisas de natureza qualitativa envolvem uma grande variedade de materiais empíricos, que podem ser estudos de caso, experiências pessoais, histórias de vida, relatos de introspecções, produções e artefatos culturais, interações, enfim, materiais que descrevam a rotina e os significados da vida humana em grupos".

## 4 Estudo de caso

O objeto de estudo desta pesquisa é a implementação de um modelo SCRUM para a gestão de projetos de desenvolvimento de softwares em uma das unidades estratégicas do Banco Alfa Instituição Financeira. Devido ao sigilo da informação, alguns detalhes do projeto estudado foram omitidos nesta pesquisa.

A gestão de projetos adotada por esta equipe ainda era balizado exclusivamente por modelos tradicionais de gestão de projetos, fortemente influenciados por réplicas de modelos adotados em outras esferas estratégicas da empresa. Diante das necessidades urgentes e emergentes da unidade estratégica referenciada, não havia muitas possibilidades reais da equipe em adequar-se à modelos de desenvolvimento de softwares mais apropriados. Na maioria das ocasiões cada projeto era conduzido por apenas um funcionário, sob a gestão centralizada do gerente de projetos, através da priorização pelo comitê estratégico da respectiva unidade. Este modelo era apoiado pela crença de que assim seria possível ampliar a gama de projetos atendidos simultaneamente na unidade estratégica, diante de uma mão de obra escassa.





Porém, com o aumento regular da necessidade de apoio tecnológico da unidade estratégica, os entrevistados desta pesquisa observaram algumas deficiências no modelo de gestão de projeto utilizado até aquele momento:

- Altos custos de manutenção: Sendo cada um dos projetos executado somente por um funcionário, e diante das constantes necessidades emergentes da unidade, a documentação do projeto era a maior prejudicada, gerando um grande débito técnico. Assim, muitas vezes, a curva de execução de uma manutenção/alteração poderia ultrapassar a curva de desenvolvimento do software original;
- Alto tempo no ciclo de entregas de softwares: Era muito comum ouvir queixas dos clientes internos de que a entrega demorava quando comparada às oportunidades que eles poderiam perder. Uma exploração mais detalhada desta revelou que os requisitos originais, por mais que fossem trabalhadas, muitas vezes mudavam em questão de dias (entre 3 a 5 dias). Apesar destas mudanças decorrerem de um refinamento natural dos requisitos levantados pelos clientes, estes somente percebiam que itens fundamentais lhe escaparam dias após ao início da execução do projeto.
- Perda do capital de conhecimento: Muitos dos conhecimentos tecnológicos e de negócios eram perdidos quando algum funcionário desta equipe era alocado em alguma outra oportunidade de trabalho no Banco Alfa Instituição Financeira, ou quando este desvinculava-se da empresa.

A fim de suprimir estas deficiências, a equipe procurou decidir realizar um piloto para adotar a implementação de um projeto através do framework SCRUM, e avaliar os resultados obtidos ao final da entrega da primeira versão do software.

O projeto escolhido para o piloto possuía um ciclo de desenvolvimento curto, como a equipe mesmo avaliou de início. O objetivo final era elaborar um software que permitisse a visualização hierárquica da alocação dos recursos humanos da rede de atuação da unidade estratégica. Foi definido que a Equipe SCRUM seria composta por: 1 PO (Product Owner), 1 SM (SCRUM Master) e 3 desenvolvedores.

Antes do início do projeto, os envolvidos estimaram que todo o projeto levaria uma média de 72 horas para ser concluído (8 horas de levantamento de requisitos + 64 horas de desenvolvimento) pelo modelo tradicional utilizado até então.

Diante da cultura organizacional encontrada na unidade, a equipe decidiu que o PO deveria pertencer a equipe de desenvolvimento, e que o mesmo deveria possuir conhecimento prévio do domínio de negócio dos clientes a serem atendidos. Este faria a intermediação destes clientes com a Equipe SCRUM. Assim, o PO elaborou o levantamento de requisitos iniciais e mínimos para a elaboração do Product Backlog, conforme tabela 01. Ainda diante da cultura organizacional fundamentada em métodos tradicionais de gerenciamento de projetos, o PO também elaborou um Termo de Abertura, com o objetivo de conceber tangibilidade à solicitação do desenvolvimento de software pelos clientes internos, porém em sua versão mais simplificada, para que aspectos burocráticos não pudessem prejudicar o ciclo de vida do framework SCRUM.

Após o Sprint Planning Meeting, o Time de Desenvolvimento foi capaz de identificar quais itens deveriam constar na primeira Sprint, desmembrando os itens em tarefas, e dimensioná-las de acordo com a complexidade e carga de trabalho, como podemos visualizar na Tabela 02.

A Sprint ficou definida em 4 dias, onde a carga de trabalho diária disponível para o projeto seria de 6 horas.

No gráfico 01 podemos observar o gráfico burndown da primeira Sprint. Na tabela 03 visualizamos a comparação entre a carga de trabalho estimada e a efetiva.



Tabela 1:  
**Product Backlog.**

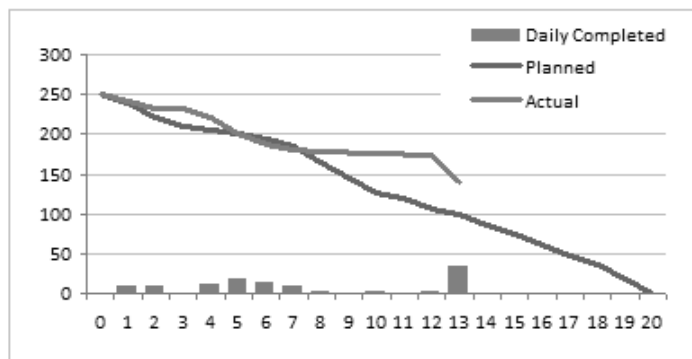
Item nr.	Descrição do item	Prioridade
1	Como usuário, quero visualizar a estrutura organizacional da unidade, através da expansão hierárquica de suas unidades imediatamente subordinadas. Não haverá restrição de acesso.	1
2	Como usuário, quero visualizar a mesma dinâmica de expansão encontrada no modelo elaborado no aplicativo de controle.	1
3	A partir de cada unidade de nossa rede, como usuário eu quero continuar expandindo as unidades vinculada a mesma, de modo que toda a estrutura organizacional possa ser encontrada dinamicamente.	1
4	Como usuário eu quero poder visualizar a lista de funcionários de uma dada unidade, através da abertura de uma dialog.	2

Nota. Fonte: do autor, 2013

Tabela 2:  
**Tarefas da Sprint #1.**

Item nr.	Descrição do item	Complexidade	Estimativa (horas)
1	Versionamento de um novo projeto	Baixa	1
2	Criar beans, persistência, classes de ação	Baixa	3
3	Criar consultas da persistência	Baixa	2
4	Criar camadas/interfaces de apresentação	Baixa	3
5	Homologação e publicação	Baixa	4
Total			13

Nota. Fonte: do autor, 2013



**Figura 6. Burndown Spring #1.**

Fonte: do autor, 2013

Tabela 3:  
**Horas Estimadas x Horas Efetivas Sprint #1.**

Day	Daily Completed	Effective Hours
0	0	0
1	6	1
2	7	4,5
3	0	0
4	0	0
Total	13	5,5
Eficiência	136,36%	

Nota. Fonte: do autor, 2013



A Sprint Review Meeting foi realizado ao final do 4º dia<sup>2</sup>, onde o PO aceitou a entrega sem maiores considerações dignas de nota. Nesta reunião e na Sprint Retrospective Meeting, o time SCRUM chegou as seguintes conclusões:

- As estimas de duração das tarefas elencadas na primeira Sprint foram superestimadas.
- A comunicação entre o Time SCRUM ocorreu muito claramente.

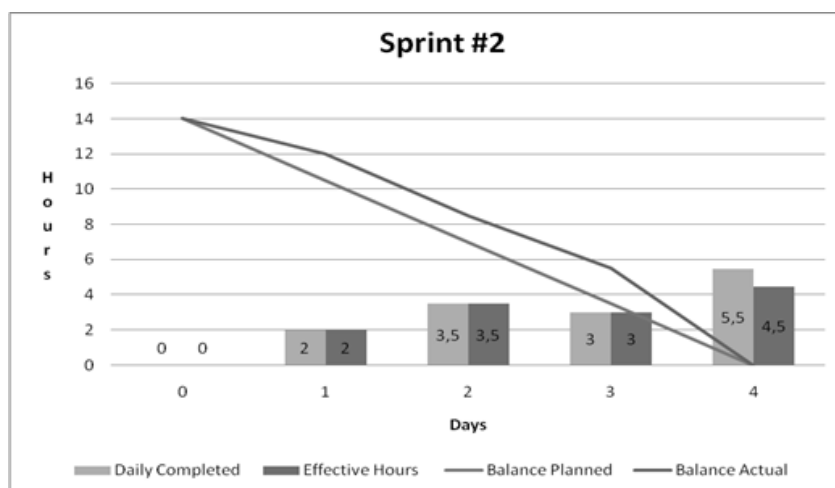
Já na Sprint Planning Meeting da segunda Sprint, somente o item número 4 compôs a Sprint Backlog. Porém, durante o desmembramento de tarefas, identificou-se que uma das bases de dados necessária para a execução do desenvolvimento estava em execução em outro projeto, e esta era fundamental para o sucesso da segunda Sprint. Assim, o Time SCRUM entendeu que esta atividade deveria ser incorporada à segunda Sprint deste projeto.

Tabela 4:  
**Tarefas da Sprint #2.**

Item nr.	Descrição do item	Complexidade	Estimativa (horas)
1	ETL da base de funcionários	Média	8
2	Criar consultas, persistência e classes de ação	Baixa	1
3	Criar camadas/interfaces de apresentação	Baixa	2
5	Homologação e publicação	Baixa	3
Total			14

Nota. Fonte: do autor, 2013

No gráfico 02 podemos observar o gráfico burndown da segunda Sprint. Na tabela 05 visualizamos a comparação entre a carga de trabalho estimada e a efetiva.



**Figura 7. Burndown Spring #2.**

Fonte: do autor, 2013

Ao final da segunda Sprint, foi realizado a Sprint Review Meeting ao final do 4º dia, onde o PO e os clientes aceitaram a entrega sem maiores considerações dignas de nota, concluindo

<sup>2</sup> O SCRUM Master e o Product Owner poderiam ter orientado o Time de Desenvolvimento à realocar tarefas ainda na primeira Sprint, porém por decisões gerenciais, o tempo extra conquistado neste evento foi direcionado a outros projetos.



assim o projeto. Nesta reunião e na Sprint Retrospective Meeting, o Time SCRUM chegou às seguintes conclusões:

O Time de Desenvolvimento não foi capaz de alocar as horas de desenvolvimento nos horários pré-definidos devido a constantes interrupções por atendimento de solicitações de outros projetos. Tais interrupções foram responsáveis pelos atrasos na conclusão de determinadas tarefas devido a um alto custo de retomada das mesmas.

- A falta de confiança na capacidade de um dos desenvolvedores por outro desenvolvedor, na execução da tarefa 1 da segunda Sprint (vide tabela 04), interrompeu o fluxo de interação do Time de Desenvolvimento.
- O SM seria fundamental na identificação e no auxílio dos problemas descritos nos itens anteriores, revelando a real importância do SM no Time SCRUM.
- As tarefas de codificação do software foram mais dinâmicas, diferentemente do que ocorria na metodologia tradicional.
- A troca de experiências e pontos de vista entre os desenvolvedores colaborou na mitigação de erros de código, reduzindo o débito técnico.
- A documentação final não foi abrangente, porém foi a mínima necessária para a colaboração mútua entre desenvolvedores e clientes.

**Tabela 5:**  
**Horas Estimadas x Horas Efetivas Sprint #2.**

Day	Daily Completed	Effective Hours
0	0	0
1	2	2
2	3,5	3,5
3	3	3
4	5,5	4,5
Total	14	13
Eficiência	7,69%	

**Nota.** Fonte: do autor, 2013

**Tabela 6:**  
**Alocação de horas por atividades.**

Descrição do item	Horas	Parcial
Levantamento de requisitos	2	2
Sprint Planning Meeting	1,5	6
Review Sprint Meeting	1	3
Sprint Retrospective Meeting	1	3
Desenvolvimento	13,5	27
Total		41

**Nota.** Fonte: do autor, 2013

**Tabela 7:**  
**Método SCRUM x Método Tradicional.**

Método	Horas
Scrum	41
Tradicional	72
Eficiência	75,61%

**Nota.** Fonte: do autor, 2013



## 5 Conclusão

Conforme demonstra o estudo de Schwaber e Sutherland(2012), são cada vez mais evidentes as dificuldades e os fracassos de projetos encontrados no modelo tradicional de gerenciamento de projetos, especialmente os de desenvolvimento de softwares. Contudo a falha não está inerente aos processos recomendados pelo PMBOK, mas ao ambiente em constante mudança imposta às oportunidades de negócios do mercado moderno. Os métodos ágeis foram uma resposta dos engenheiros de softwares às estas oportunidades.

Como pontos positivos na adoção do framework SCRUM, esta pesquisa evidenciou os seguintes argumentos:

- Disseminação do conhecimento: Através do desenvolvimento em grupo, a troca de conhecimentos ocorre dinamicamente de forma mais clara.
- Ganho em escala: A demonstração de diferentes pontos de vista advindos das mais diversas experiências profissionais, proporcionou um plano de execução mais simples e eficiente.
- Refatoração natural: Os processos de refatoração de código ocorreram dinamicamente enquanto o desenvolvimento de código evoluía.
- Mitigação dos erros: Devido a uma maior disseminação do conhecimento e da refaturação natural do código, os erros decorrentes do desenvolvimento de código diminuíram significativamente.
- Documentação inerente: O compartilhamento de informações através da documentação foi necessário, porém natural durante todo o processo de desenvolvimento.
- Quebra da monotonia: Com o ganho de escala e da comunicação dinâmica, os desenvolvedores não sentiram a monotonia encontrada no modelo tradicional.

Como aspecto negativo do SCRUM, pontuou-se:

- Readaptação deficiente: Algumas interrupções de interações em eventos SCRUM foram observadas na segunda Sprint do projeto objeto de estudo desta pesquisa. Apesar de ocorrerem em menor número do que no modelo tradicional, percebeu-se que a retomada destas interações eram mais onerosas que no método tradicional, representando um impacto maior no desempenho do método.

## Referências

CARLOS TRISTACCI. *Priorização MoSCoW*. Disponível em: <http://www.carlostristacci.com.br/blog/priorizacao-moscow/>. Data de acesso: 18/11/2013 às 11:19 hrs.

CESAR, A. M. R. V. C. *Método do Estudo de Caso (Case Studies) ou Método do Caso (Teaching Cases)*: Uma análise dos dois métodos no Ensino e Pesquisa em Administração. Disponível em: [http://www.mackenzie.br/fileadmin/Graduacao/CCSA/remac/jul\\_dez\\_05/06.pdf](http://www.mackenzie.br/fileadmin/Graduacao/CCSA/remac/jul_dez_05/06.pdf). Data de acesso: 18/11/2013 às 16:48 hrs.

CHANDOO.ORG. *Use burndown charts in your project management reports*. Disponível em: <http://chandoo.org/wp/2009/07/21/burn-down-charts/>. Data de acesso: 18/11/2013 às 11:29 hrs.





V SINGEP

Simpósio Internacional de Gestão de Projetos, Inovação e Sustentabilidade  
International Symposium on Project Management, Innovation and Sustainability

ISSN: 2317 - 8302

CRUZ, F. *Scrum e PMBOK: unidos no gerenciamento de projetos*. São Paulo: Brasport Livros e Multimídia, 2013.

DEVELOPERWORKS. *Java Modeling: Holonic Software Development*. Disponível em: <http://www.ibm.com/developerworks/java/library/j-jmod1023/>. Data de acesso: 18/11/2013 às 11:10 hrs.

DIVUS TECNOLOGIA. *Gestão de projetos com SCRUM*. Disponível em: [http://www.google.com.br/url?q=http://www.divus.com.br/sobre/fotos/scrt09-gestao-de-projetos-com-scrum&sa=U&ei=m9eEUtavF4eOkAfhrICQAg&ved=0CCEQFjAB&usg=AFQjCNG26REZR\\_zbT2rMQyhXkBZaDVlwDA](http://www.google.com.br/url?q=http://www.divus.com.br/sobre/fotos/scrt09-gestao-de-projetos-com-scrum&sa=U&ei=m9eEUtavF4eOkAfhrICQAg&ved=0CCEQFjAB&usg=AFQjCNG26REZR_zbT2rMQyhXkBZaDVlwDA). Data de acesso: 14/11/2013 às 12:00 hrs.

IT-ZYNERGY APS. *Planning Poker*. Disponível em: <http://www.it-zynergy.com/scrum-planning-poker>. Data de acesso: 18/11/2013 às 10:00 hrs.

MANIFESTO ÁGIL. *Princípios por trás do manifesto ágil*. Disponível em: <http://manifestoagil.com.br/principios.html>. Data de acesso: 14/11/2013 às 14:30 hrs.

MUNDO PROJECT MANAGEMENT. *Guia PMOK 5ª Edição*. Disponível em: <http://blog.mundopm.com.br/2012/03/07/guia-pmbok-5a-edicao/>. Data de acesso: 18/11/2013 às 14:15 hrs.

PMI. *O que é o PMI*. Disponível em: <https://brasil.pmi.org/brazil/AboutUS/WhatisPMI.aspx>. Data de acesso: 18/11/2013 às 14:00 hrs.

RODRIGOCUSTODIO.NET. *Scrum Diário*. Disponível em: <http://rodrigocustodio.net/2012/02/19/scrum-diario/>. Data de acesso: 18/11/2013 às 10:27 hrs.

SHWABER, K; SUTHERLAND, J. *Guia do Scrum*. Disponível em: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-Portuguese-BR.pdf#zoom=100>. Data de acesso: 14/11/2013 às 11:30 hrs.

SHWABER, K; SUTHERLAND, J. *Software in 30 days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors In the Dust*. New Jersey: John Wiley & Sons, Inc., 2012.

T2TI. *Metodologia de Desenvolvimento*. Disponível em: <http://www.t2ti.com/erp/>. Data de acesso: 18/11/2013 às 09:46 hrs.