

## GESTÃO ÁGIL DE PROJETOS - SCRUM

Gabriel Wasil Jaciuk Raio<sup>(\*)</sup>

### RESUMO

Este artigo tem como objetivo apresentar uma visão geral do *SCRUM*, com seus conceitos, papéis e certificações. Em estudos recentes foram constatados que a maioria dos projetos de *software* falham, e os motivos são vários como o não cumprimento dos prazos e custos e o principal motivo é a insatisfação do cliente por receber o que não havia pedido e deixar de receber o que foi solicitado. Visando melhoria nesses tipos de projetos, metodologias ágeis como o *SCRUM* foram desenvolvida para poder guiar todo o ciclo dos projetos e principalmente de *softwares*. Este trabalho tem como objetivo apresentar uma breve explicação sobre o *framework SCRUM*, seus papéis, conceitos e até as suas formas de certificações profissionais. A metodologia utilizada foi qualitativa, o levantamento de dados foi através de livros e pesquisas na internet.

**PALAVRAS-CHAVE:** *Scrum*; Gerência de Projetos, Metodologias Ágeis.

### ABSTRACT

*This article aims to present an overview of SCRUM, with its concepts, roles and certifications. In recent studies found that most software projects fail, and the reasons are many as the fulfillment of the terms and costs and the main reason is the dissatisfaction of the customer to receive the request and that there was no longer receive the requested . For improvement in these types of projects, agile methodologies such as SCRUM were developed in order to guide the entire project cycle and especially software. This paper aims to present a brief explanation of the Scrum framework, their roles, concepts and even their forms of professional certifications. The methodology used was qualitative data collection was through books and internet searches.*

*Keywords: SCRUM, Project Management, Agile Methodology.*

(\*) WASIL, Gabriel. Tecnólogo em Sistemas da Informação, Analista de Sistemas, Gerente de Projetos da INFOX e aluno Pós-Graduando no Curso MBA em Gerência de Projetos na Faculdade de Administração e Negócios de Sergipe – FANESE, estando sob a orientação do Prof. Me. José Guilherme da Cunha Castro Filho  
E-mail: [gabrieljaciuk@gmail.com](mailto:gabrieljaciuk@gmail.com)

## LISTA DE FIGURAS

Figura 1 - O Destino dos Projetos

Figura 2 - Manifesto Ágil

Figura 3 - Histórico dos Métodos Ágeis

Figura 4 - Ciclo de vida *SCRUM*

Figura 5 - Papéis do *SCRUM*

SUMÁRIO	
INTRODUÇÃO .....	04
SCRUM.....	07
Como funciona? .....	10
Papéis e responsabilidades.....	11
Certificações.....	13
CONCLUSÃO .....	14
REFERÊNCIAS .....	15

## INTRODUÇÃO

As recentes definições sobre metodologias ou métodos ágeis evoluíram a partir da metade de 1990 como parte de uma reação contra métodos considerados "pesados", caracterizados por uma pesada regulamentação, regimentação e micro gerenciamento usado no modelo em cascata que é composto basicamente por atividades sequenciais de levantamento de requisitos, análise, projeto, implementação, teste, implantação e manutenção. Este modelo é derivado de outras engenharias tradicionais (Civil, Elétrica, Naval) e foi o primeiro a ser usado pela Engenharia de Software, na década de 70 (COCKBURN; HIGHSMITH, 2001).

Dados de 1995 segundo a *Standish Group - Chaos Report*, usando como exemplos 8380 projetos, mostram que apenas 16,2% dos projetos foram entregues respeitando os prazos e os custos e com todas as funcionalidades especificadas. Aproximadamente 31% dos projetos foram cancelados antes de estarem completos e 52,7% foram entregues, porém com prazos maiores, custos maiores ou com menos funcionalidades do que especificado no início do projeto (SOARES, 2009).

Dentre os projetos que não foram finalizados de acordo com os prazos e custos especificados, a média de atrasos foi de 222%, e a média de custo foi de 189% a mais do que o previsto. Considerando todos os projetos que foram entregues além do prazo e com custo maior, na média, apenas 61% das funcionalidades originais foram incluídas. Mesmo os projetos cuja entrega é feita respeitando os limites de prazo e custos possuem qualidade suspeita, uma vez que provavelmente foram feitos com muita pressão sobre os desenvolvedores, o que pode quadruplicar o número de erros de *software*, segundo a mesma pesquisa. As principais razões destas falhas estavam relacionadas com o processo em cascata (COCKBURN; HIGHSMITH, 2001).

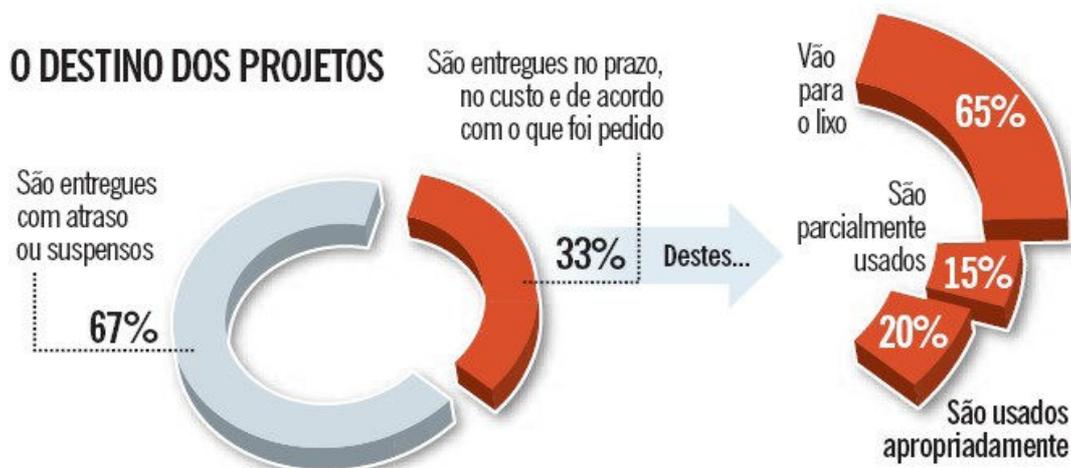


Figura 1 - O Destino dos Projetos

Fonte: Chaos Report 2009

A idéia das metodologias ágeis é o enfoque nas pessoas e não em processos ou algoritmos (ISOTTON NETO, 2008).

Apesar do uso crescente das metodologias ágeis, ainda falta uma base maior de projetos para verificar suas vantagens. Mesmo assim, os resultados iniciais em termos de qualidade, confiança, entregas e custos são promissores (ISOTTON NETO, 2008).

Um projeto para ser considerado ágil deve aceitar a mudança e não tentar prever o futuro, pois a mudança não deveria ser considerada como um problema, até por que ocorrerá de qualquer forma. A forma como devemos trata-las é que devemos gerenciar, avaliar e responder de forma precisa e ágil.

Em 2001, membros proeminentes na área de *software* se reuniram em *Utah* nos EUA e embora cada membro tivesse suas próprias práticas e teorias sobre como fazer um projeto de software ter sucesso, cada qual com as suas particularidades, 22 todos concordavam que, em suas experiências prévias, um pequeno conjunto de princípios sempre parecia ter sido respeitado quando os projetos foram considerados de sucesso e então publicaram o Manifesto ágil, documento que reúne os princípios e práticas desta metodologia de desenvolvimento. Mais tarde, algumas pessoas formaram a *Agile Alliance*, uma organização não lucrativa que promove o desenvolvimento ágil (AGILE, 2009).

Segundo Fowler (2005) os conceitos chave do Manifesto Ágil são:



(Manifesto for Agile Software Development - <http://agilemanifesto.org/>)

Figura 2 - Manifesto Ágil

Fonte: Elaborado Givanildo Nascimento com base no manifesto ágil.

Esses conceitos aproximam-se melhor com a forma que pequenas companhias de Tecnologia da Informação trabalham e respondem a mudanças (SOARES, 2009).

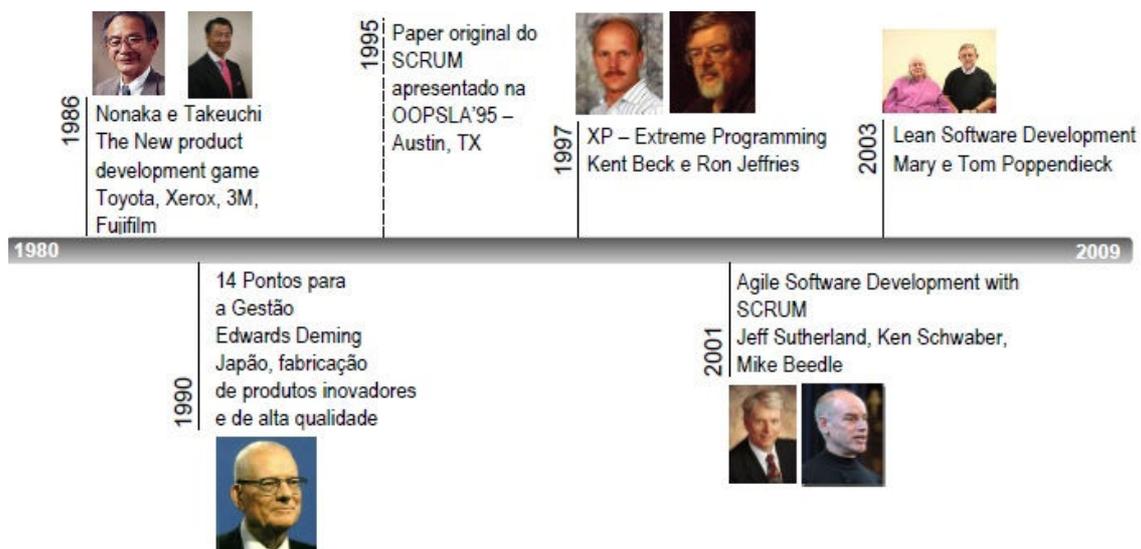


Figura 3 - Histórico dos Métodos Ágeis

Fonte: Elaborado por Givanildo Nascimento com base na história do manifesto Ágil.

## **SCRUM**

O *Scrum* é um processo de desenvolvimento iterativo e incremental para gerenciamento de projetos e desenvolvimento ágil de *software*. Apesar de a palavra não ser um acrônimo, algumas empresas que implementam o processo a soletram com letras maiúsculas como *SCRUM*. Isto pode ser devido aos primeiros artigos de *Ken Schwaber*, que capitalizava *SCRUM* no título. *Scrum* não é um processo prescribente, ou seja, ele não descreve o que fazer em cada situação. Ele é usado para trabalhos complexos nos quais é impossível prever tudo o que irá ocorrer. Apesar de *Scrum* ter sido destinado para gerenciamento de projetos de software, ele pode ser utilizado em equipes de manutenção de software ou como uma abordagem geral de gerenciamento de projetos/programas. Inicialmente, o *Scrum* foi concebido como um estilo de gerenciamento de projetos em empresas de fabricação de automóveis e produtos de consumo, por *Takeuchi* e *Nonaka* no artigo "*The New Product Development Game*" (*Harvard Business Review*, Janeiro-Fevereiro 1986). Eles notaram que projetos usando equipes pequenas e multidisciplinares (*cross-functional*) produziram os melhores resultados, e associaram estas equipes altamente eficazes à formação *Scrum* do *Rugby* (utilizada para reinício do jogo em certos casos). *Jeff Sutherland*, *John Scumniotales*, e *Jeff McKenna* conceberam, documentaram e implementaram o *Scrum*, incorporando estilos de gerenciamento, *Ken Schwaber* formalizou a definição de *Scrum* e ajudou a implantá-lo em desenvolvimento de software em todo o mundo.

A função primária do *Scrum* é ser utilizado para o gerenciamento de projetos de desenvolvimento de *software*. Ele tem sido usado com sucesso para isso, assim como *Extreme Programming* (XP) e outras metodologias de desenvolvimento. Porém, teoricamente pode ser aplicado em qualquer contexto no qual um grupo de pessoas necessitem trabalhar juntas para atingir um objetivo comum, como iniciar uma escola pequena, projetos de pesquisa científica, ou até mesmo o planejamento de um casamento.

Segundo (Franco, 2007) o *Scrum* não define uma técnica específica para desenvolvimento de software durante a etapa de implementação, ele se concentra em descrever como os membros da equipe devem trabalhar para produzir um sistema flexível, num ambiente de mudanças constantes. A ideia central do *Scrum* é que o desenvolvimento de sistemas envolve diversas variáveis (ambientais e técnicas) e elas possuem grande probabilidade de

mudar durante a execução do projeto (por exemplo: requisitos, prazos, recursos, tecnologias, etc).

O *SCRUM* possui um processo bem definido com uma fase de planejamento e de encerramento. Entre estas fases, há uma fase chamada de *Sprint*, com duração de 2 a 4 semanas, que ocorre várias vezes durante o projeto. O que caracteriza as metodologias ágeis são as iterações, O *Sprint* é como uma caixa preta onde ocorre o desenvolvimento do produto. Antes do início de cada iteração é feita uma Reunião de Planejamento (*Sprint Planning Meeting*) entre os desenvolvedores e o responsável pelo negócio para priorização do trabalho, escolher e estimar tempo para cada tarefa que poderá ser desenvolvida pelo time durante a *Sprint*.

Após a finalização da reunião de planejamento é executado o *sprint*, durante este período o time controla o andamento do projeto através de reuniões diárias rápidas com duração máxima de 15 minutos e onde se é observado através de gráficos ou *post-it* o andamento do projeto.

Ao finalizar uma *Sprint* é feita uma análise de tudo que foi planejado para ser entregue e o que se foi realmente implementado através de uma demonstração do produto, e também é feito uma validação da entrega. Este processo é chamado de *Sprint Review*.

*Scrum* não é uma metodologia que irá te ajudar a desenvolver melhores produtos, não lhe dará a resposta de como desenvolver *software* de qualidade mais rapidamente. *Scrum* é uma ferramenta, um *framework*, que você pode usar para identificar o que você precisa fazer para desenvolver software de qualidade rapidamente, *scrum* não necessita que a equipes estejam co-localizada, porém lhe permite medir a produtividade de equipes co-localizadas.

*Scrum* é um *framework Agile* que permite entregar um “valor de negócio” mais elevado num período de tempo mais curto, permite entregar rapidamente software funcionando e de qualidade a cada duas a quatro semanas (*Sprints*). O cliente defini as prioridades. O time se auto-organiza e determina a melhor forma de entregar as funcionalidades de maior priorização. No fim de cada *Sprint* o time apresenta para o cliente as funcionalidades acordadas.

Vale ressaltar que as práticas do *SCRUM* podem ser aplicadas em qualquer contexto onde pessoas precisem trabalhar juntas para atingir um objetivo comum. *SCRUM* é recomendado para projetos de outras áreas além de

software principalmente para projetos de pesquisa e inovação (CHAPIEWSKI, 2009).

Segundo o *Scrum Guide* de 2010 (KEN SCHWABER, JEFF SUTHERLAND) Scrum, que é fundamentado na teoria de controle de processos empíricos, emprega uma abordagem iterativa e incremental para otimizar a previsibilidade e controlar riscos. Três pilares sustentam cada implementação de controle de processos empíricos.

A transparência garante que aspectos do processo que afetam o resultado devem ser visíveis para aqueles que gerenciam os resultados. Esses aspectos não apenas devem ser transparentes, mas também o que está sendo visto deve ser conhecido. Isto é, quando alguém que inspeciona um processo acredita que algo está pronto, isso deve ser equivalente à sua definição de pronto.

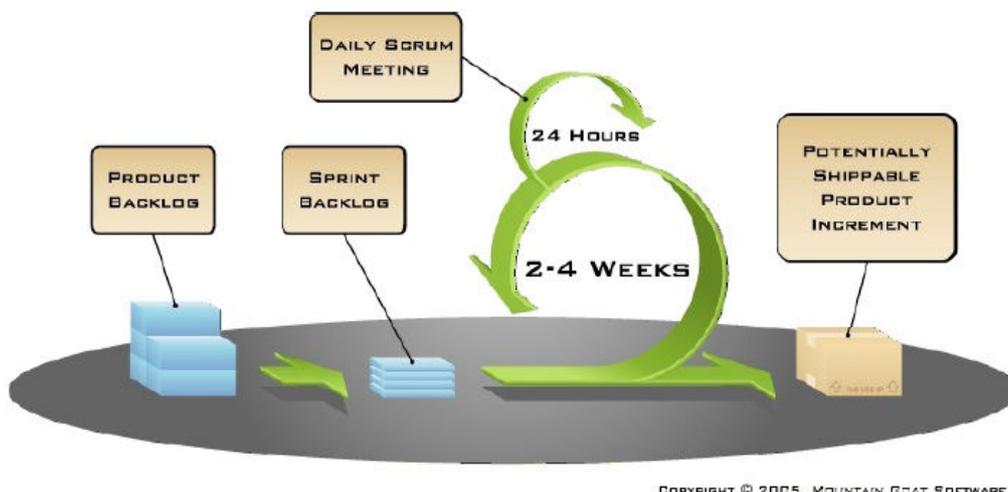
Os diversos aspectos do processo devem ser inspecionados com uma frequência suficiente para que variações inaceitáveis no processo possam ser detectadas. A frequência da inspeção deve levar em consideração que qualquer processo é modificado pelo próprio ato da inspeção. O problema acontece quando a frequência de inspeção necessária excede a tolerância do processo à inspeção. Felizmente, isso não parece ser verdade no desenvolvimento de software. Os outros fatores são a habilidade e a aplicação das pessoas em inspecionar os resultados do trabalho.

Se o inspetor determinar, a partir da inspeção, que um ou mais aspectos do processo estão fora dos limites aceitáveis e que o produto resultante será inaceitável, ele deverá ajustar o processo ou o material sendo processado. Esse ajuste deve ser feito o mais rápido possível para minimizar desvios posteriores.

O conhecimento das suas práticas permite a aplicação das mesmas de forma variada e este é um dos aspectos positivos do *SCRUM*, a adaptabilidade (SCHWABER, 2004).

## COMO FUNCIONA

Existem três pontos para inspeção e adaptação em *Scrum*. A *Daily Scrum* é uma reunião utilizada para inspecionar o progresso em direção à Meta da *Sprint* e para realizar adaptações que otimizem o valor do próximo dia de trabalho. Além disso, as reuniões de Revisão da *Sprint* e de Planejamento da *Sprint* são utilizadas para inspecionar o progresso em direção à Meta da *Release* e para fazer as adaptações que otimizem o valor da próxima *Sprint*. Finalmente, a Retrospectiva da *Sprint* é utilizada para revisar a *Sprint* passada e definir que adaptações tomarão a próxima *Sprint* mais produtiva, recompensadora e gratificante.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Figura 4 - Ciclo de vida SCRUM

Fonte: Copyright 2005, Mountain Goat Software

O *framework Scrum* consiste em um conjunto formado por Times de *Scrum* e seus papéis associados, *Time-Boxes* (eventos com duração fixa), Artefatos e Regras. O ciclo do *SCRUM* tem o seu progresso baseado em uma série de iterações bem definidas, chamada *Sprints*, cada uma com duração de duas a quatro semanas, chamada de *Time-box*. Antes de cada *Sprint*, realiza-se uma Reunião de Planejamento - *Sprint Planning Meeting* em que o time de desenvolvedores tem contato com o cliente - *Product Owner* para priorizar o trabalho que precisa ser feito, selecionar e estimar as tarefas que o time pode realizar dentro da *Sprint*. A próxima fase é a Execução da *Sprint* (MOUNTAIN, 2009).

Durante a execução da *Sprint*, o time controla o andamento do desenvolvimento realizando Reuniões Diárias Rápidas - *Daily Meeting*, não mais que 15 minutos de duração, e observando o seu progresso usando um gráfico chamado *Sprint Burndown*. Ao final de cada *Sprint*, é feita uma revisão no produto entregue para verificar se tudo realmente foi implementado (PEREIRA et al, 2009).

Ao final da *Sprint*, deve-se realizar uma Reunião de Revisão - *Sprint Review*, em que o time demonstra o produto gerado na *Sprint* e valida se o objetivo foi atingido. Logo em seguida, realiza-se a Reunião de Retrospectiva - *Sprint Retrospective*, uma reunião de lições aprendidas, com o objetivo de melhorar o processo/time e/ou produto para a próxima *Sprint* (FIGUEIREDO, 2009). SCRUM torna-se ideal para projetos dinâmicos e suscetíveis a mudanças de requisitos, sejam eles novos ou apenas requisitos modificados. No entanto para aplicá-lo, é preciso entender antes os seus papéis, responsabilidades, conceitos e artefatos das fases de seu ciclo (FIGUEIREDO, 2009).

## PAPEIS E RESPONSABILIDADES

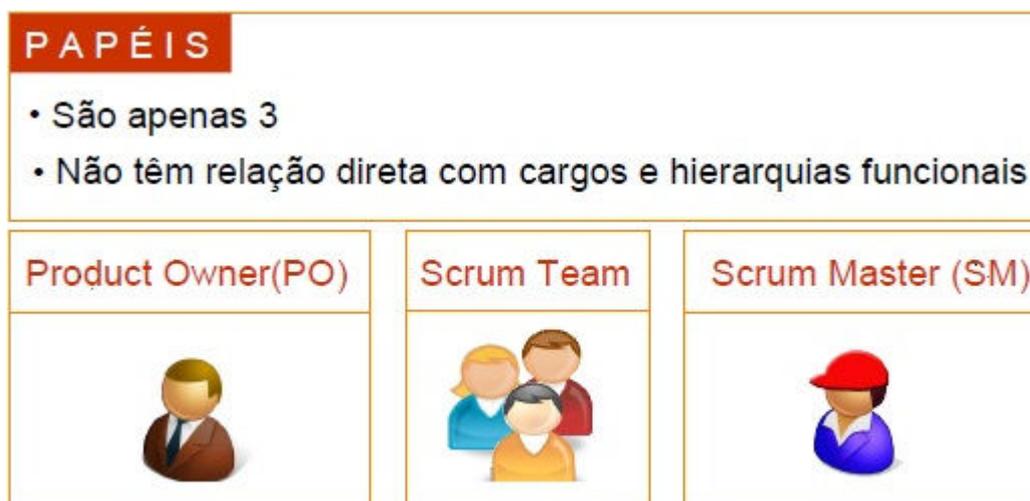


Figura 5 - Papéis do SCRUM

Fonte: Elaborado Givanildo Nascimento com base no manifesto ágil.

## PRODUCT OWNER

É a peça chave para a definição do produto, traduzindo literalmente ele é o “dono do projeto” é quem informa quais as funcionalidades e característica o produto deve conter, no caso o *Product Owner* é o cliente do projeto, ele é o

responsável por definir a prioridade dos itens do *Product Backlog*, o *Product Owner* deve estar sempre acessível para equipe, ele têm a responsabilidade de não trazer novas funcionalidades durante o Sprint e de fazer a validação do produto ao final de cada Sprint, O *Product Owner* também têm que estar presente em todas as reuniões para poder definir as prioridades e explicar a necessidade de cada tarefa ao *Scrum Team*, caso o cliente recuse a participar da reunião, o *Scrum Master* têm três opções para seguir:

- 1- Tentar convencer ao *Product Owner* da sua importância na reunião
- 2- Replanejar a reunião, de acordo com a disponibilidade do *Product Owner*
- 3- Elegir uma pessoa da equipe para assumir a responsabilidade do *Product Owner*, onde ele terá liberdade para definir a prioridade dos itens.

## **SCRUM TEAM**

É a equipe de desenvolvimento do produto onde vai executar as tarefas de acordo com as prioridades definidas pelo *Product Owner*. Segundo (Franco, 2007), é a equipe de projeto que possui autoridade de decidir sobre as ações necessárias e de se organizar para poder atingir os objetivos preestabelecidos.

A equipe *Scrum* é envolvida, por exemplo, na estimativa de esforço, na criação e revisão da lista de funcionalidades do produto, sugerindo obstáculos que precisam ser removidos do projeto. Idealmente a equipe deve ser pequena, cerca de 5 a 9 pessoas, porém pode ser maior, todos trabalham juntos afim de cumprirem a entrega de todas as funcionalidades que assumiram no *Sprint*, não há uma definição de papéis formais como programador, analista ou arquiteto, o que não impede que haja essas funções.

## **SCRUM MASTER**

É o responsável por aplicar e garantir que o projeto está sendo conduzido de acordo com as práticas, valores e regras definidas no *SCRUM*, e verificar se o progresso do projeto está de acordo com o que foi especificado pelo *Product Owner*, o *Scrum Master* possui o papel de gerente de projeto, que pode ser exercido por qualquer membro da equipe, ele é responsável por interagir com o *Product Owner* e com o *Scrum Team*, ele também é responsável por remover quaisquer barreiras identificadas e proteger contra interferências externa que

impeça que a *Scrum Team* trabalhe da forma mais produtiva possível durante o andamento do projeto.

## CERTIFICAÇÕES

*Scrum Alliance* é uma comunidade emergente de profissionais que são *Certified Scrum Masters* que estão trabalhando juntos para compartilhar as suas experiências e aprimorar o seu entendimento do processo *Scrum*. A manifestação primária de isto é um *Scrum Gathering*, que é tido varias vezes ao ano, onde pessoas se juntam e fazem progressos no processo *Scrum*. O processo de certificação em *Scrum* é um pouco diferente das certificações que vemos no mercado atualmente. Para esclarecer um pouco melhor este processo. No primeiro nível de certificação, que seriam os treinamentos de CSM - *Certified Scrum Master* e CSPO - *Certified Scrum Product Owner*, a pessoa apenas recebe o treinamento de um *Trainer* aprovado pela *Scrum Alliance*. Este treinamento atesta que o participante possui conhecimento e é capaz de aplicar o *Scrum*, mas não comprova experiência ou habilidade.

A certificação CSD - *Certified Scrum Developer* é uma certificação avançada, lançada pela *ScrumAlliance* em 2009 e que chegou no Brasil em outubro de 2010 através da parceria Massimus e Kleer, a primeira organização da América Latina a oferecer o curso. Apenas organizações REP (*Registered Education Provider*) podem oferecer e dar a certificação CSD. O objetivo do programa é treinar os membros de um time de desenvolvimento em práticas avançadas de engenharia ágil e outras habilidades, juntamente com os fundamentos do *Scrum*, de forma que os desenvolvedores, CSMs e CSPOs tenham todas as habilidades necessárias para desenvolver *software*.

No segundo nível de certificação, a pessoa precisa responder a uma série de questionamentos sobre a utilização de *Scrum* em um projeto em que participou. Estes questionamentos são avaliados por uma comissão da *Scrum Alliance*, que pode fazer novos questionamentos para esclarecer algo que não esteja claro. Esta comissão verifica se no projeto avaliado os conceitos de *Scrum* foram seguidos corretamente, conferindo (ou não) o título de CSP - *Certified*

*Scrum Practitioner*. Nesse nível, a certificação também atesta experiência e habilidade com a implantação e utilização de *scrum*.

No terceiro nível da certificação, a pessoa passa a ter um tutor, que vai treiná-la para a realização de treinamentos de novos *Certified Scrum Masters* e *Certified Scrum Product Owners*, ou então atuar como *coach* de *Scrum* em organizações. A impressão que tenho, é que este nível da certificação é algo parecido com um "treinamento Jedi", mestre e aprendiz. Quando o tutor decidir, o candidato passa por uma avaliação, onde uma comissão pode aprová-lo como *Certified Scrum Trainer* ou *Certified Scrum Coach*. Nesse nível, a certificação atesta que a pessoa é capaz de transmitir conhecimento para outras pessoas que estejam interessadas em aprender *Scrum*.

## **CONCLUSÃO**

Este artigo apresentamos de forma simples e clara o *framework SCRUM* como um dos métodos ágil mais utilizados no mundo e que aos poucos está recebendo adeptos no Brasil. Para projetos que necessitam de um resultado ou algo palpáveis com mais agilidade o *SCRUM* é uma das práticas mais indicadas para atingir este objetivo. Ao contrario do que muitos pensam demonstramos neste artigo que o *SCRUM* tem processos bem definidos e um volume de documentos e regras que devem ser seguidas para seu sucesso total. As certificações disponíveis no mercado também ajudam aos profissionais a aplicarem em suas equipes o *SCRUM*. A intenção do *SCRUM* é causar a redução de falhas em projetos, principalmente de software por conta do grande nível de incertezas que este tipo de projeto tem e as técnicas de integração constantes, por meio da comunicação entre a equipe nos diversos processos como as reuniões de planejamento e acompanhamento facilitam e minimizam as falhas dos projetos de *softwares*.

## REFERÊNCIAS

Wikipedia SCRUM. <http://pt.wikipedia.org/wiki/Scrum> Acessado em: 26 de Junho de 2011.

IMPROVE IT. Scrum. Disponível em: <<http://www.improveit.com.br/scrum>>. Acessado em: 12 de Julho de 2011

<http://cid-963c3dfeeb673b90.office.live.com/browse.aspx/Monografias/SCRUM> - Francisco Neto. Acessado em 15 de Julho de 2011.

AGILE MANIFESTO. Manifesto for Agile Software Development. Agile Alliance, 2001. Disponível em: <<http://www.agilemanifesto.org/iso/ptbr/>>. Acesso em: 19 de Julho de 2011.

AGILE. **Agile Alliance**. Disponível em: < <http://www.agilealliance.org/>>. Acesso em: 20 Julho 2011.

SOARES, S.S. Metodologias Ágeis Extreme Programming e SCRUM para o Desenvolvimento de Software. Disponível em: <[com.br/index.php/reinfo/article/view/146/38](http://com.br/index.php/reinfo/article/view/146/38)>. Acesso em: 26 junho de 2011.

BRUNO, L.B. Estudo Comparativo da Compatibilidade Entre as Melhores Praticas do PMI® e SCRUM Monografia <http://pt.scribd.com/doc/23370443/Estudo-comparativo-da-compatibilidade-entre-as-melhores-praticas-do-PMI-e-Scrum>. Acessado em: 26 de Julho de 2011.