

Utilização do framework Scrum no gerenciamento de projetos de software

Josemara Oliveira Gonçalves (UTFPR – Campus Londrina) josemara2006@gmail.com

José Luis Dalto (UTFPR – Campus Londrina) josedalto@utfpr.edu.br

Resumo:

A área de Tecnologia da Informação está em constante transformação e os projetos de desenvolvimento de software requerem controle e assertividade nas entregas, pois influenciam diretamente nos resultados das áreas de negócios envolvidas. Novas tecnologias de desenvolvimento, metodologias e frameworks para o gerenciamento de projetos são constantemente lançados ou adaptados, demandando estudo e análise das melhores ferramentas que tenham capacidade de responder ao alto grau de desempenho que estes projetos necessitam. Este estudo descreve a implantação do framework Scrum pela equipe de desenvolvimento de software em uma instituição de ensino privada. A revisão da literatura foi efetuada através de artigos e publicações com referência no assunto. São apresentados os conceitos do Scrum e a importância da metodologia ágil no gerenciamento de projetos de softwares, e como ocorreu a implantação do Scrum no controle e gerenciamento dos projetos vinculados ao time de desenvolvimento de software, com objetivo de demonstrar que este framework pode ser adaptado de acordo com as necessidades de cada projeto e empresa. São descritos os benefícios e dificuldades enfrentados na implantação do framework, os ganhos na comunicação e produção de documentos dos projetos que foram gerenciados aplicando os ritos e artefatos do Scrum. O artigo é concluído mencionando os próximos passos para a consolidação do framework como a metodologia oficial de desenvolvimento dos projetos futuros na empresa objeto de estudo.

Palavras chave: Scrum, Ágil, Tecnologia da Informação, Gerenciamento de Projetos.

Using the framework Scrum in the management of software projects

Abstract

Information Technology has been passing by constant transformation once software projects development require control and accuracy in deliveries and it directly influence the results of the business. New development technologies, methodologies and frameworks for project management have been constantly launched and adapted, requiring best tools analysis with capacity to respond to high performance that these projects need. This paper presents framework Scrum for software development to private educational institution. The survey was made by articles and references at the subject. It shows the agile methodology importance in control and management of projects linked to software development team, demonstrating that this framework can be adapted according to the needs of each project and company. Furthermore, the benefits and difficulties faced in framework implementation and the gains in communication and documentation are summarized in this paper too. Finally, this document ends showing next steps for framework consolidation as official methodology for future software development in company.

Key-words: Scrum, Agile, Information technology, Projects Management.

1. Introdução

A área de tecnologia da informação vive em constante evolução necessitando que em seus projetos exista respostas e entregas cada vez mais rápidas, sendo necessária a adoção de metodologias e ferramentas que auxiliem no controle das atividades que compõem os projetos.

Entende-se por metodologia ou práticas de desenvolvimento o conjunto de atividades recomendadas para o desenvolvimento de softwares, normalmente divididas em fases que representam as subdivisões do processo para ordená-lo e melhor gerenciá-lo (SOMMERVILLE, 2011).

De acordo com GONÇALVES, MÉXAS e DRUMOND (2016), há diversas metodologias disponíveis para auxiliar o gerenciamento dos projetos de software que são divididas em:

- a) Tradicional: com embasamento na análise do projeto mantendo tudo em documentos. Este tipo de metodologia as mudanças de escopo e requisitos não são bem-vindas. O *Rational Unified Process* (RUP) é o principal framework dessa categoria.
- b) Ágil: totalmente adaptável a mudanças nos requisitos, porém, fraco na parte contratual e de documentos. *Extreme Programming* (XP), *Feature Driven Development* (FDD), *Crystal*, *Test Driven Development* (TDD), *Dynamic Systems Development Method* (DSDM), *Adaptive Software Development* (ASD), *Pragmatic Programming* (PP) e *Scrum* são consideradas metodologias ágeis.

O Scrum é um framework com base na filosofia ágil e utiliza práticas iterativas e incrementais com objetivo de entregar um serviço/produto com valor agregado ao cliente (SCHWABER, SUTHERLAND, 2013).

Os principais objetivos das atividades propostas no Scrum são transparência, motivação, aperfeiçoamento contínuo, auto-organização da equipe, redução de riscos e falhas do projeto e a entrega de valor ao cliente. Atende todos os tamanhos e níveis de complexidade de projetos.

Este estudo busca apresentar o conceito do *framework* Scrum e sua aplicabilidade em uma empresa descrevendo a forma que foi organizado visando demonstrar que pode ser adaptado de acordo com as necessidades de cada organização.

2. Revisão Bibliográfica

2.1. Conceitos do Scrum

O Scrum é um *framework* estrutural utilizado desde 1990 para gerenciar o desenvolvimento de produtos e projetos, pode ser utilizado para projetos complexos e adaptativos que visa a entrega de produtos com maior valor agregado. Pode ser utilizado em diversos projetos de diferentes segmentos que visem a desburocratização, interação entre equipe e patrocinadores e agilidade na entrega do produto (CARVALHO; MELLO, 2012).

De acordo com Strauhs (2013), o Scrum foi concebido por Jeff Sutherland e Schwaber influenciados por três grandes temas: a teoria de sistemas adaptativos e complexos; as melhores práticas da indústria japonesa, seguindo os princípios do *Lean Development* (desenvolvimento enxuto) implementado em empresas como a Toyota e Honda; e as estratégias de gestão do conhecimento desenvolvidas por Takeuchi e Nonaka.

O nome Scrum surgiu da comparação entre desenvolvedores e jogadores de *Rugby*. Scrum é a denominação da rápida reunião que ocorre quando os jogadores de *Rugby* vão iniciar um lance. Esta analogia foi usada porque no *Rugby* cada time age em conjunto, como uma unidade integrada. Nele, cada membro desempenha um papel específico e todos se ajudam em busca de

um objetivo comum como deve ocorrer com os times de desenvolvimento que adotam o método Scrum (CARVALHO; MELLO, 2012).

O Scrum é leve e simples de entender. Não é um processo ou uma técnica para construir projetos e produtos, é um *framework* que pode ser empregado vários processos ou técnicas. Cada componente dentro do *framework* serve a um propósito específico e é essencial para o uso e sucesso do Scrum (SCHWABER; SUTHERLAND, 2013).

De acordo com Schwaber e Sutherland (2013), o Scrum emprega uma abordagem iterativa e incremental para aperfeiçoar a previsibilidade e o controle de riscos. É movido por três pilares que apoiam a implementação e controle de processo:

- a) Transparência: os aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Esta transparência requer aspectos definidos por um padrão comum para que os observadores compartilhem um mesmo entendimento do que está sendo visto;
- b) Inspeção: todo projeto Scrum deve ser frequentemente inspecionado visando a correta aplicação dos artefatos do framework e amadurecimento da equipe Scrum. Esta inspeção é importante para detectar as dificuldades do time envolvido possibilitando a melhoria contínua, no entanto não deve ser muito frequente a ponto de atrapalhar a execução diária das tarefas;
- c) Adaptação: quando praticado desde início os pilares de transparência e inspeção proporcionam a detecção de pontos a serem ajustados. A fase de adaptação será importante para que as melhorias ocorram de forma a minimizar os impactos no projeto visando a prevenção de problemas futuros.

De acordo com Sutherland (2014), o Scrum é fortemente ligado às práticas japonesas, onde é considerado como um modo de fazer, de ser, um modo de vida.

Para Brod (2015), praticar Scrum é como tocar violão: com poucos acordes e um mínimo de talento é possível obter bons resultados. Mas para chegar a ser solista em uma orquestra há um grande caminho a ser trilhado.

O Scrum é simples e objetivo, porém, exige projetos bem documentados e possui um conjunto de formalidades em sua prática. Na execução de projetos é imprescindível que se estabeleça um nível de confiança mútua entre cliente e equipe, de forma que o cliente possa mudar os requisitos do projeto caso identifique esta necessidade para garantir a entrega de valores, de forma que, desenvolvedor ou cliente não fiquem em desvantagem ao ter que realizar adequações durante o andamento do projeto. (BROD, 2015)

Um dos conceitos-chave do Scrum é que os membros da equipe decidem, como irão executar o trabalho. A responsabilidade da direção da empresa é estabelecer os objetivos estratégicos restando a equipe decidir como atingi-los (SUTHERLAND, 2014).

No desenvolvimento de projetos ágeis é essencial que na equipe todos tenham as habilidades necessárias para concluir o trabalho. Em clássicas estruturas organizacionais, normalmente encontra-se a equipe de planejamentos, seguida pela equipe de desenvolvimento, equipe de testes, de produção e de entrega. E cada um precisa concluir sua parte do projeto antes que seja avançado para o próximo passo o Scrum visa agilizar a etapa final de forma a unir essas diferentes etapas (BROD, 2015).

Para Sabbagh (2013), uma das grandes qualidades do Scrum é a junção das práticas de mercado já conhecidas e consagradas de uma forma organizada e que funciona sem acrescentar novos conceitos e destaca como benéficos:

- a) Entregas frequentes que representem valor ao cliente;
- b) Redução dos riscos do projeto;
- c) Maior qualidade no produto gerado;
- d) Mudanças utilizadas como vantagem competitiva;
- e) Visibilidade do progresso do projeto;
- f) Redução do desperdício;
- g) Aumento de produtividade.

Em um projeto com Scrum, as mudanças são consideradas oportunidades e são parte do ambiente dinâmico e representam a descoberta dos detalhes do produto, que é um processo incremental que ocorre durante todo o projeto. O desperdício é reduzido e evitado por meio da busca da simplicidade, por este motivo é essencial que o time Scrum produza e utilize apenas o que é necessário e suficiente para atender o propósito do projeto (SABBAGH, 2013).

2.2. Papéis no Scrum

No Scrum existem três papéis:

Product Owner (PO): é o responsável principal pelo projeto, deve dominar o negócio e ter consciência da sua representatividade e responsabilidade. O PO é quem dá o aceite final às entregas nas revisões do *Sprint*.

Para Sutherland (2014), é essencial que *Product Owner*:

- a) Esteja bem-informado em relação ao setor;
- b) Deve ter o poder de tomar decisões;
- c) Precisa estar disponível para explicar o que precisa ser feito e o porquê;
- d) Ele precisa ser responsável pelo valor, ou seja, deve estar diretamente ligado ao produto e/ou serviço produzido no projeto.

De acordo com Schwaber e Sutherland (2013), o *Product Owner* é o único responsável por gerenciar o *Backlog* do Produto que inclui:

- a) Definir claramente os itens;
- b) Definir a priorização dos itens visando objetivo final do projeto;
- c) Estar sempre disponível para esclarecer eventuais dúvidas do time Scrum;
- d) Garantir o valor do trabalho realizado pelo Time de Desenvolvimento;
- e) Garantir que o *Backlog* do Produto seja visível, transparente e claro para todos.

Scrum Master (SM): é um membro da equipe que durante um *Sprint* garante o bom andamento do projeto, assegurando que os ritos do Scrum sejam cumpridos, seus artefatos sejam usados de maneira correta e qualquer obstáculo ao trabalho da equipe seja removido. O *Scrum Master* é a pessoa responsável por garantir que o Scrum seja entendido e aplicado, cabe a ele ajudar todos os envolvidos no projeto, esclarecer as dúvidas do Scrum, treinar a equipe, facilitar os eventos do Scrum e remover os impedimentos que possam prejudicar ou atrasar as entregas (SCHWABER; SUTHERLAND, 2013).

Time de Desenvolvimento: é o grupo de pessoas responsáveis pela construção do projeto. Como no Scrum não há hierarquia entre os membros é necessário que a equipe seja auto gerenciável

na distribuição de tarefas e execução das atividades definidas no *Sprint Backlog* (SABBAGH, 2013).

Os times de desenvolvimento devem apresentar características de auto-organização e multifuncionais. Os integrantes do Time de Desenvolvimento podem ter habilidades individuais especializadas, mas a responsabilidade pertence ao time de desenvolvimento como um todo. É recomendado que as equipes possuam no mínimo 3 e no máximo 9 integrantes facilitando a manutenção da agilidade e garantindo as entregas dentro prazo estabelecido no *Sprint*. Acima de 9 pessoas o gerenciamento torna-se complexo (SCHWABER; SUTHERLAND, 2013).

2.3. Artefatos do Scrum

No Scrum são utilizados apenas os artefatos necessários e suficientes para o entendimento e desenvolvimento do projeto proporcionando a agilidade durante as fases. Ferramentas de gerenciamento de projetos são úteis até o ponto em que começam a gerar desperdício, quando em excesso desloca o foco daquela que é realmente necessária. A documentação utilizada no suporte à gestão do projeto e ao desenvolvimento do produto devem cumprir seu papel sem adicionar burocracia ao trabalho (SABBAGH, 2013).

Product Backlog: O Scrum se inicia com a definição do *Product Backlog* que representa a lista de desejos do projeto apresentadas no formato de *User Stories* que irão compor o produto e uma primeira estimativa do esforço para seu desenvolvimento. Essa lista deve ser priorizada e dividida em *Sprints Backlogs*, cada um contemplando as tarefas que serão realizadas dentro de um período fixo (LEITE; LUCRÉDIO, 2014).

No Scrum não há um formato ou padrão para o *Product Backlog*. O importante é que ele tenha o formato de itens em uma sequência, com prioridade única e facilmente reordenável. O gerenciamento do *Product Backlog* pode ser realizado através de softwares especializados, planilha e até com notas adesivas em um quadro branco, o importante é que os itens e sua priorização fiquem claros para o time Scrum (SABBAGH, 2013).

User Storie: devem traduzir de forma clara e simples os itens que compõem o projeto. Deve “caber” no trabalho a ser executado em uma *Sprint* pela equipe de desenvolvimento, deve ser facilmente testável (SABBAGH, 2013).

Toda *User Storie* deve atender ao conceito INVEST (*Independent, Negotiable, Valuable, Estimable, Small, Testable*), ou seja, deve ser independente de outras histórias, negociável entre o fornecedor e o cliente, deve apresentar valor de negócio, deve ser pequena e testável (SUTHERLAND, 2014).

As *User Stories* também devem atender aos critérios SMART (*Specific, Measurable, Attainable, Relevant, Time-boxed*), cada tarefa deve ser especificada, mensurável, alcançável, relevante e possuir um tempo limite para ser cumprida. Devem ser escritas mesmo antes de se chegar ao *Product Backlog* (BROD, 2015).

Épico: são *User Stories* que representam itens grandes demais ou sem detalhes suficientes para serem desenvolvidos. Os épicos serão detalhados para compor uma ou mais *user storie* quando forem priorizados. É comum que sejam priorizadas no *Product Backlog*, *user stories* que sejam um épico. É preciso analisar se uma determinada *user storie* pode ser dividida em outras antes de seu detalhamento técnico. Sua divisão deve representar um claro valor de negócio para o cliente ou os usuários finais do produto ou projeto (SCHWABER; SUTHERLAND, 2013).

Sprint Backlog: O *Backlog* da *Sprint* é um conjunto de itens do *Product Backlog* selecionados para a *Sprint*. Representa a previsão do time de desenvolvimento sobre qual funcionalidade estará no próximo incremento e sobre o trabalho necessário para entrega da funcionalidade

(SABBAGH, 2013).

O *Backlog da Sprint* é um plano com detalhes suficientes para que as mudanças no progresso sejam entendidas durante a reunião diária. Sempre que um novo trabalho é necessário, o Time de Desenvolvimento adiciona este ao *Backlog da Sprint*. Conforme o trabalho é realizado ou completado, a estimativa do trabalho restante é atualizada. Quando elementos do plano são considerados desnecessários, eles são removidos. Somente o time de desenvolvimento pode alterar o *Backlog da Sprint* que deve ser visível e refletir o trabalho a ser realizado durante a *Sprint* (SCHWABER; SUTHERLAND, 2013).

As decisões para otimizar o valor e o controle de riscos são feitas com base na percepção existente do estado dos artefatos. Quando os artefatos não são completamente transparentes as decisões podem ser falhas e os valores podem diminuir e os riscos aumentarem (SCHWABER; SUTHERLAND, 2013).

O *Scrum Master* deve trabalhar com o *Product Owner*, Time de Desenvolvimento, e outras partes envolvidas para entender se os artefatos estão plenamente transparentes. O *Scrum Master* pode detectar transparência incompleta pela inspeção dos artefatos, percebendo padrões, ouvindo atentamente o que está sendo dito, e detectando diferenças entre o esperado e o resultado real (SCHWABER; SUTHERLAND, 2013).

Definição de “Pronto”: todos os envolvidos em um projeto que utiliza Scrum devem ter entendimento do conceito de “Pronto”. Os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo, assegurando a transparência (SABBAGH, 2013).

Esta definição é utilizada para assegurar quando o trabalho está completo e orienta o time de desenvolvimento no conhecimento de quantos itens do *Backlog* do Produto podem ser selecionados durante a reunião de planejamento da *Sprint*. Em toda *Sprint* o objetivo é entregar incrementos de funcionalidades potencialmente utilizáveis que atendam aos critérios de “Pronto” definidos em conjunto pelo time Scrum (SCHWABER; SUTHERLAND, 2013).

2.4. Eventos do Scrum

Eventos são usados no Scrum para criar uma rotina e minimizar a necessidade de reuniões não programadas. Cada evento Scrum foi criado com objetivo de inspecionar e adaptar os processos que envolvem o projeto e permitir a transparência. Todos os eventos oficiais do Scrum têm uma duração máxima definida. Uma vez que a *Sprint* começa, sua duração é fixada e não pode ser reduzida ou aumentada. Os eventos restantes podem terminar sempre que o propósito do evento é alcançado, garantindo que uma quantidade adequada de tempo seja gasta sem permitir perdas no processo (SABBAGH, 2013).

Sprint: É o principal evento do Scrum, um *time-boxed* de um mês ou menos, durante o qual uma versão utilizável do produto é criada. *Sprints* tem durações definidas de acordo com o ritmo de desenvolvimento. Uma nova *Sprint* inicia imediatamente após a conclusão da *Sprint* anterior (CARVALHO; MELLO, 2009).

As *Sprints* são compostas por uma reunião de planejamento, reuniões diárias, o desenvolvimento, uma revisão e a retrospectiva da *Sprint*. Durante a *Sprint* “não são feitas mudanças que possam pôr em perigo o objetivo definidos para este período de trabalho do projeto” (SCHWABER; SUTHERLAND, 2013).

Reunião de planejamento do *Sprint*: ocorre antes do início da *Sprint*, com objetivo planejar o trabalho que será desenvolvido de acordo com as histórias priorizadas no *Product Backlog* que serão selecionadas e divididas em tarefas para desenvolvimento. É obrigatória a presença do

PO e da equipe de desenvolvimento (TAVARES, 2015).

Nessa reunião serão avaliadas quais as capacidades da equipe e é definido o que cada pessoa fará, quais tecnologias serão adotadas e quais os itens do *Product Backlog* serão atendidos. O resultado da reunião será o objetivo do *Sprint* e o *Sprint Backlog* (SCHWABER; SUTHERLAND, 2013).

Reunião diária: também chamada de *Stand Up Meeting*, já que é de praxe que todos os membros a realizem de pé. Diariamente o time de desenvolvimento se reúne para que cada membro da equipe relate as atividades desenvolvidas no dia anterior, e o que será feito no dia de hoje e relatar quais obstáculos que podem impedir o andamento das tarefas. Cabe ao *Scrum Master* avaliar os obstáculos relatados e quando possível auxiliar na resolução. Obrigatório ter uma duração máxima de 15 minutos, aproximadamente 90 segundos por participante (CARVALHO; MELLO, 2009).

Cada membro da equipe deve responder as seguintes perguntas:

- a) O que você fez ontem?
- b) O que você fará hoje?
- c) Quais obstáculos que impedem (ou podem impedir) seu trabalho?

Esta reunião deve ser presencial e não pode ser substituída por uma lista de discussões ou outra forma de encontro, também não são destinadas à solução de problemas. Como não existe hierarquia entre os membros de um time Scrum, cada pessoa deve procurar resolver os problemas entre seus pares, acessando-os diretamente (BROD, 2015).

Reunião de revisão do *Sprint*: realizada no final da *Sprint*, neste encontro é apresentado ao PO uma versão do produto desenvolvido durante a *Sprint* (BROD, 2015).

Reunião de retrospectiva do *Sprint*: é o último evento realizado no ciclo do Scrum. Ocorre no final da *Sprint* e neste encontro deve ser feito uma análise do que deu certo e errado durante a *Sprint* e o que é possível melhorar. É realizada com a presença do time de desenvolvimento e *Scrum Master*. Esta reunião deve ter no máximo 3 horas de duração, as oportunidades de melhoria identificadas devem ser incorporadas a um plano de ação a ser executado (TAVARES, 2015).

Scrum é corretamente aplicado quando:

- a) Ao final de cada *Sprint* foi entregue algo funcional que agrega valor ao projeto;
- b) *Product Backlog* claramente definido e priorizado de acordo com valor do negócio para o cliente;
- c) O *Product Owner* participa ativamente na construção do projeto;
- d) O time de desenvolvimento participa da definição de todas as etapas do projeto;
- e) O time conhece sua velocidade de desenvolvimento;
- f) Não há nível hierárquico entre os membros da equipe.

3. Caso de Uso

Neste estudo é relatado a implantação do framework Scrum pela equipe de T.I. de uma instituição de ensino privada localizada em Londrina-PR. A equipe é composta por 40 colaboradores responsáveis pelo desenvolvimento do sistema de informação da instituição. O sistema é dividido em três módulos:

- a) Acadêmico: responsável pela estrutura das funcionalidades e processos acadêmicos da instituição, como, cadastro de cursos e grade curricular, cadastro de alunos, controle de notas entre outros.
- b) Financeiro: responsável pelo controle das funcionalidades financeiras que dizem respeito a instituição, como, preços por curso/disciplina, controle de mensalidades entre outras funcionalidades necessárias para manutenção financeira da instituição.
- c) Web - AVA (Ambiente Virtual de Aprendizagem): responsável pelo desenvolvimento das aplicações destinadas aos alunos e professores.

Os projetos desenvolvidos pela equipe utilizam o modelo tradicional também chamado de cascata. Neste modelo as atividades são agrupadas em tarefas que são executadas sequencialmente, de forma que uma tarefa só poderá ter início quando a anterior for finalizada. O projeto é dividido nas seguintes fases:

- a) Análise e definição dos requisitos: nesta etapa estabelece-se os requisitos do projeto que deseja desenvolver e são elaborados documentos como a especificação funcional e técnica, modelos de testes de desenvolvimento, modelos de testes a serem aplicadas com a área de negócio;
- b) Implementação: nesta fase são desenvolvidas as funcionalidades de acordo com a especificação elaborada na documentação da fase anterior;
- c) Testes: na fase de testes é realizada afim de identificar possíveis falhas ou requisitos que não tenham sido implementados. Os testes são realizados pela equipe interna de desenvolvimento e responsáveis do projeto;
- d) Entrega: após testes e aprovação das partes interessadas o produto é disponibilizado para uso.

O modelo em cascata não permite o feedback da área de negócio durante as fases de desenvolvimento, não suporta modificações, sendo que uma vez que forem alterados os requisitos é necessário voltar para fase inicial, o que acarretará em atraso. Gasta-se muito tempo e recursos na documentação do projeto.

Com muitos projetos que englobam várias áreas da instituição, o custo de deslocamento de recursos entre as equipes é alto, ocasionando muitos atrasos nas entregas. Visando otimizar o desenvolvimento a gerência decidiu alterar a metodologia de desenvolvimento optando por utilizar o *framework* Scrum que promete agilidade no formato de desenvolvimento e comunicação (entre equipe e cliente), facilitando também a análise de resultados e melhor acompanhamento dos indicadores como o total de horas planejadas e total de horas gastas no projeto.

Em 2016 a equipe iniciou a implantação do Scrum em seus projetos. Em junho os colaboradores participaram de um treinamento para introdução dos conceitos do Scrum. Após estudo inicial foram definidos alguns projetos pilotos para que fossem desenvolvidos utilizando o Scrum.

Os artefatos e ritos do Scrum ficaram organizados da seguinte forma:

Time de desenvolvimento: Os projetos desenvolvidos utilizando o Scrum as equipes são montadas de acordo necessidade do projeto e grau de conhecimento no assunto, garantindo a assim a pluralidade tornando-as multidisciplinar.

Product Backlog: A fase de concepção antes era realizada através de reuniões com os responsáveis do projeto e na sequência elaborado o documento de especificação funcional e técnico, com a utilização do Scrum este formato foi adequado com maior proximidade com

cliente e equipe que realização em conjunto a definição e priorização dos itens que compõem o *Product Backlog*.

A após definição das *User Stories* o *Product Backlog* é definido na reunião do *Sprint Backlog* de acordo com os objetivos do projeto. Após priorização são definidas as *Sprints* de acordo com tamanho do projeto, podendo ter duração de no mínimo uma e no máximo quatro semanas.

Sprint: A equipe divide as atividades de cada *Sprint* de acordo com a área de atuação de cada integrante.

As iterações são definidas de acordo com as atividades requeridas para cada entrega. É desejável que cada atividade tenha duração 8 horas.

Sprint Review: ao final do período de cada *Sprint* a equipe se reúne para formalizar as entregas realizar os testes e formalizar as entregas das atividades desenvolvidas juntamente com *Product Owner* e demais partes interessadas.

Para acompanhamento da lista de tarefas a equipe adotou o quadro de atividades, e diariamente durante as reuniões são atualizadas as fases de cada atividade. Exemplo de quadro de tarefas utilizado pela equipe:

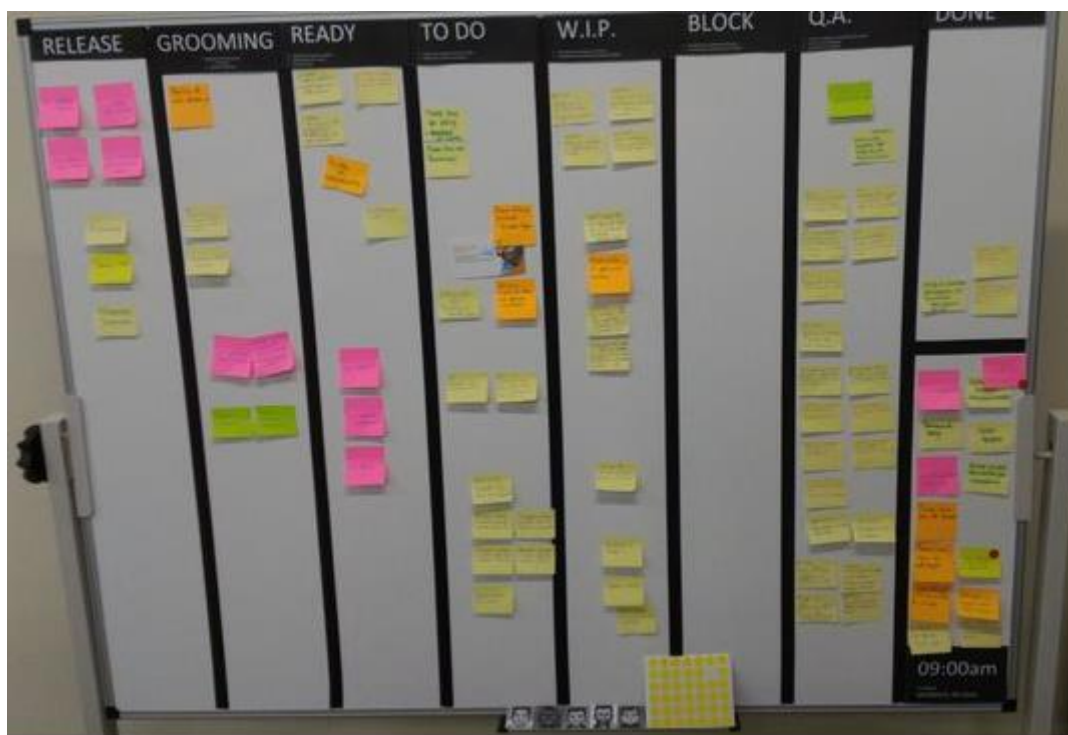


Figura 2 – Quadro de Tarefas

No quadro as tarefas são separadas de acordo com a fase que se encontram:

- a) *Release*: tarefas definidas no *Product Backlog* não desenvolvidas;
- b) *Grooming*: tarefas em fase de definição do escopo;
- c) *Ready*: tarefas já especificadas aguardando desenvolvimento;
- d) *To Do*: tarefas já priorizadas que estão aguardando desenvolvimento;
- e) *Work In Progress* (W.I.P.): tarefas em desenvolvimento;
- f) *Block*: tarefas “bloqueadas” devido algum impedimento que impossibilita o

desenvolvimento;

g) *Quality Assurance* (Q.A.): tarefas em fase de teste;

h) *Done*: tarefas finalizadas e aprovadas pelo *Product Owner*.

Reunião diária: No início da manhã a equipe se reúne para relatar as atividades realizadas no dia anterior, e quais foram programadas para o dia, a reunião tem duração máxima de 15 minutos.

3.1. Lições aprendidas

Após um ano utilizando Scrum nos projetos, os membros da equipe já estão familiarizados com os conceitos do framework. Todos concordam que Scrum facilitou a execução dos projetos. A comunicação e a redução da documentação obrigatória na entrega dos projetos representam os maiores benefícios percebidos pela equipe.

A maior dificuldade enfrentada foi a adequação do Scrum com as políticas internas da empresa. O framework é simples e visa agilidade nas entregas, porém, a fim de ter todos os projetos devidamente documentados a empresa ainda solicita que alguns documentos sejam entregues, porém já foi possível adequá-los afim de manter os benefícios do Scrum e o nível de documentação dos projetos.

A utilização do framework iniciou-se com a equipe de T.I., as demais áreas não estavam preparadas para dar o suporte necessário que Scrum exige das partes envolvidas no projeto. Atualmente a empresa está promovendo treinamento do framework para os colaboradores das áreas de negócio que atuam em conjunto com a equipe de tecnologia da informação.

4. Conclusões e Trabalhos Futuros

O presente estudo teve por objetivo apresentar o conceito do *framework* Scrum e sua aplicabilidade em uma empresa descrevendo a forma que foi organizado visando demonstrar que pode ser adaptado de acordo com as necessidades de cada organização.

Após implantação do Scrum, a equipe da empresa descrita no estudo de caso, já pôde notar ganho na otimização do tempo e trabalho realizado. Principal benefício é o entendimento do projeto entre todos os membros da equipe, já houve redução no esforço para produção das extensas documentações requeridas no modelo anterior. Todos da equipe de desenvolvimento e área de negócio possuem conhecimento das entregas realizadas e situação atual do projeto. A instituição já percebe ganho na produtividade e assertividade nos processos entregues. A equipe de tecnologia da informação já está engajada na aplicação do Scrum em seus projetos o passo seguinte é fomentar a participação das áreas de negócios mostrando a importância da participação na concepção e acompanhamento dos projetos do início ao fim.

Com a realização deste estudo foi possível chegar à conclusão que as metodologias ágeis veem crescendo cada vez mais no mercado. A adoção deste tipo de metodologia é impulsionada pela necessidade cada vez maior de se apresentar resultados em menor tempo. Diferentes das demais técnicas e metodologias utilizadas na execução de projetos o Scrum oferece acompanhamento de todos os envolvidos de forma simples e fácil. Na equipe descrita no estudo de caso a utilização do Scrum cumpriu seu propósito, em pouco tempo de utilização já se nota ganhos na comunicação entre os envolvidos e entregas cada vez mais ágeis e assertivas.

Referências

BROD, C. *Scrum – Guia prático para projetos ágeis*. 2. ed. São Paulo: Novatec. 2015.

CARVALHO, B. V.; MELLO, C. H. P. *Revisão, análise e classificação da literatura sobre o método de desenvolvimento de produtos ágil Scrum*. Simpósio de Administração da Produção, Logística e Operações Internacionais – SIMPOI, 12. São Paulo, 2009. Disponível em: <http://www.simpoi.fgvsp.br/arquivo/2009/artigos/E2009_T00109_PCN92031.pdf>. Acesso em 07 set. 2017.

CARVALHO, B. V.; MELLO, C. H. P. *Aplicação do Método Ágil Scrum no Desenvolvimento de Produtos de Softwares em uma Pequena Empresa de Base Tecnológica*. Gestão & Produção, São Carlos, v.19, n.3, p.557-573. 2012. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0104-530X2012000300009&lng=pt&nrm=iso>. Acessos em 07 set. 2017.

GONÇALVES, E. F.; MÉXAS, M. P.; DRUMOND, G. M. *Análise das práticas de gerenciamento de projetos e desenvolvimento de software*. In: Simpósio de Excelência em Gestão e Tecnologia. Rio de Janeiro, 2016. Disponível em: <<https://www.aedb.br/seget/arquivos/artigos16/17924164.pdf>>. Acesso em 08 set. 2017.

LEITE, L. M.; LUCRÉDIO, D. *Desenvolvimento de Software utilizando o Framework Scrum: Um Estudo de Caso*. T.I.S. São Carlos, v.3, n.2, p. 114-121. 2014. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/viewFile/81/75>> Acesso em 07 set. 2017.

SABBAGH, R. *Scrum Gestão ágil para projetos de sucesso*. 1ª ed. São Paulo: Casa do Código, 2013.

SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. New York: Scrum.org, 2013.

SOMMERVILLE, I. *Engenharia de Software*. 9ª ed. São Paulo: Pearson Prentice Hall, 2011.

STRAUHS, E. P. *Limitadores e soluções de contorno na adoção de práticas ágeis do método SCRUM em projetos de desenvolvimento de software*. 2013. 128 f. Dissertação (Mestrado em Informática) - Universidade Católica de Brasília, Brasília, 2013. Disponível em: <<https://bdtd.ucb.br:8443/jspui/handle/123456789/1405>> Acesso em 09 set. 2017.

SUTHERLAND, J. *SCRUM. A arte de fazer o dobro do trabalho na metade do tempo*. 1. ed. São Paulo: Leya, 2014.

SUTHERLAND, J. *The Scrum Papers: Nut Bolts, and Origins of an Agile Framework*. Cambridge: Scrum Inc., 2012.

TAVARES, B. G. *Análise da Gestão de Riscos no desenvolvimento de projetos de software via Scrum*. 2015. 147 f. Dissertação (Mestrado em Engenharia de Produção) - Universidade Federal de Itajubá, Itajubá, 2015. Disponível em: <<https://repositorio.unifei.edu.br/xmlui/handle/123456789/297>> Acesso em: 07 set. 2017.