

# **Scrum framework e sua usabilidade com a ferramenta de princípios ágeis, Trello**

**Letícia de Souza Martins** – leticia.martins2509@gmail.com

Universidade de Araraquara (Uniará)

## **Trabalho Orientado pelo Prof.º Marcelo Torres**

**Resumo:** Abordagem em um *framework* de Metodologias Ágeis, denominada como Scrum, onde seu uso está se evoluindo por todo mundo pela sua eficiência em organizar uma equipe e com isso conseguir garantir suas entregas de forma que o prazo acompanhe o estimado. É estabelecido um comum acordo com a equipe sobre prazos, meios de comunicação e tarefa que cada membro ficará responsável a realizar, com isso as informações sendo repassadas para todos em reuniões diárias para que o desenvolvimento do software não tenha desperdício de tempo e sua qualidade garantida. Para que tudo aconteça de forma organizada existem várias ferramentas que auxiliam no gerenciamento e na organização de tarefas da equipe, dentre elas, Trello, tema de estudo deste trabalho.

**Palavras Chave:** Metodologias ágeis, Scrum, Trello.

**Abstract:** *Approach in a framework of Agile Methodologies, denominated as Scrum, where its use is evolving throughout the world by its efficiency in organizing a team and with this to be able to guarantee its deliveries so that the term follows the estimated time. A common agreement is established with the team about deadlines, means of communication and tasks that each member will be responsible to carry out, with this information being passed to everyone in daily meetings so that the software development time is not wasted and its quality is guaranteed. For everything to happen in a organized way there are several tools that help in the management and organization of tasks of the team among them, Trello, theme of study of this work.*

**Keywords:** *Agile methodologies, Scrum, Trello.*

## Introdução

Projetos de desenvolvimentos de software tem crescido a cada dia e com esse crescimento a preocupação na organização e qualidade de sua entrega ao cliente. Há alguns anos o projeto era entregue para que o cliente desse uma opinião do lado funcional, e o trabalho para refazer geralmente é dobrado, pois tinha que chegar no dia da entrega para perceber que precisava de um ou mais detalhes práticos e/ou necessários e o cliente raramente ficava satisfeito. Por isso a necessidade em substituir uma única entrega completa, por sequências de entregas de partes funcionais do projeto para o cliente.

Com referência nesse cenário, alguns fatores críticos ou exigências para o sucesso se destacam: a agilidade, a capacidade de adaptação, o poder de inovar de forma rápida e eficiente, o potencial de aprimoramento contínuo sob grandes restrições de recursos. Em resposta a essas exigências, fortalecem-se os sistemas de gerenciamento de projetos, como forma de gerir os empreendimentos temporários, únicos e multifuncionais, que caracterizam o processo de implementação de estratégias, inovação, adaptação e aprimoramento.

A Metodologia Ágil garante a qualidade e eficiência desde o processo de desenvolvimento até a entrega ao cliente através de princípios e práticas adotadas que vem revolucionando a área de projetos de desenvolvimento de *software*.

Nesse artigo será focado em um *framework* Scrum, com metodologias de gerenciamento de projetos que investe em vários processos e técnicas, onde deixa claro a eficiência de gerenciamento de desenvolvimentos de projetos de maneira que possa saber onde está o possível erro de uma equipe e que modo melhorar. Ferramentas são necessárias para garantir a organização que o *Scrum* tanto preza, será abordado a ferramenta Trello, que com sua simplicidade e clareza de manusear as tarefas mostra o quanto é eficiente.

## **Metodologia científica**

Para a elaboração desse artigo, foi levantado a necessidade de uma ferramenta ágil para suprir o que a metodologia Scrum propõe no dia a dia.

Foi realizado estudo de caso em uma ferramenta gratuita que facilita nessa dinâmica com os membros da equipe, priorizando a transparência, organização e agilidade que fazem parte da essência do Scrum.

Scrum é framework de metodologias ágeis, que junto de suas regras e artefatos priorizam uma entrega funcional a curto prazo. Sua equipe é considerada pequena e auto-organizáveis, buscando sempre trabalhar em conjunto para um desenvolvimento e entrega do produto com alta qualidade.

Trello é uma ferramenta gratuita com valores ágeis, que visa a transparência e organização. Com sua simplicidade em manuseá-la supri a necessidade de ter uma equipe presente no mesmo local, dando recursos suficientes para se trabalhar a distância sem perder a dinâmica entre os membros da equipe.

### **1. Metodologias Ágeis**

Desde que surgiu a necessidade de garantir a produtividade e qualidade do *software* em desenvolvimento, várias metodologias foram avaliadas, porém uma de muito sucesso, as metodologias ágeis ou leves.

Segundo Agile (2011) e citado por Andrade (2012, p. 3), o surgimento dessa metodologia surgiu por volta dos anos 90, quando a parte gerencial de projetos não estava mais suportando o peso do desenvolvimento custoso e lento da época, que no começo era chamado de metodologias 'leves', o que em 2001 se tornou 'metodologias ágeis', o seu resultado surgiu a partir de 17 especialistas de software compartilhando a mesma técnica e criando a partir de suas experiências equivalentes, o manifesto ágil.

Segundo Sutherland e citado pelo site oficial da Microsoft (2016), o manifesto ágil estabeleceu um conjunto de valores a serem seguidos para seu sucesso:

- Indivíduos e interações entre eles
- Fornecer um software em funcionamento
- Colaboração ao pedido do cliente

- Responder positivamente a mudanças

Por mais que, ferramentas e processos, documentação geral, negociação entre contratos e ter um plano para o desenvolvimento de um software sejam valorizados, os itens citados acima são mais importantes no desenvolvimento ágil. Esses valores são resumidos a partir de 12 princípios básicos do manifesto ágil (Site oficial Manifesto Ágil):

1. A prioridade é agradar o cliente, através de entregas adiantadas e contínuas de um software de valor.
2. Como processos ágeis se adequam a mudanças, aceitando tais mudanças no requisito mesmo que esteja no fim do desenvolvimento, para prevalecer o cliente em seu mercado de competições.
3. Entregar software em funcionamento em prazos curtos que variam de semanas a meses.
4. Todos os membros da equipe trabalhar em conjunto e diariamente durante o projeto.
5. Construir projetos com membros da equipe motivados, disponibilizando todo ambiente técnico necessário.
6. Manter transparência e conversas claras para que todos estejam com informações coerentes.
7. Software funcional é o primeiro passo para o progresso.
8. Para conservar um ambiente sustentável, o time de desenvolvedores, os usuários e patrocinadores devem manter passos constantes.
9. Conciliação entre excelência técnica e bom design é possível obter agilidade.
10. Simplicidade: visando aumentar a quantidade de trabalho que não precisou ser feito.
11. Times auto organizáveis garantem na melhor arquitetura, requisitos e designs.
12. Aos poucos, o time vai se aperfeiçoando em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Através dos princípios citados acima, pode-se dizer que esta metodologia aborda formas adaptativas onde o cliente, com o cenário funcional em curtos prazos que duram

apenas algumas semanas, e com isso o cliente tem a opção de fazer alterações de requisitos de acordo com sua prioridade e/ou necessidade. Enquanto metodologias tradicionais tem como prioridade a geração de documentação sobre o projeto e no cumprimento rígido dos processos até a entrega e na proposta ágil, o foco é priorizar a relação entre os envolvidos e no desenvolvimento em si.

Essa agilidade citada várias vezes, refere-se em criar oportunidades de resolver problemas e alterações em meio do desenvolvimento a ponto que o projeto não perca a qualidade e o seu ponto de controle de prazos. Para melhor resultado, é necessário que a equipe entre em uma comunicação onde todos estão aptos a se adaptar a tarefas e aperfeiçoa-las, além de enfatizar uma estratégia para que a entrega seja totalmente efetiva e o software esteja funcionando perfeitamente no ambiente combinado com o cliente. (Cruz, 2015)

Pode-se citar várias metodologias ágeis: Scrum, XP (*Extreme programming*), Kanban, Lean, DSM, Crystal, entre outras. Todos eles têm seu nicho de aplicação e particularidades, mas todos são interativos e incrementais, seguindo os princípios ágeis. (Brod, 2015)

## **2. Scrum**

Criado por Jeff Sutherland em 1993 e formalizado em 1995 junto com Ken Schwaber, o Scrum veio da necessidade em organizar e entregar um projeto de forma rápida, com qualidade e funcional.

O termo Scrum vem do jogo de rúgbi, onde o time se junta por um único objetivo, levar a bola no campo adversário. Todo seu objetivo as claras, um alinhamento previsível e cuidadoso, tudo se unindo entre os membros. Uma tática perfeita para o que uma equipe deseja fazer. (Sutherland, 2014)

O Scrum não é uma técnica ou um processo para construir um produto, ao contrário, é um framework que visa os princípios do manifesto ágil e pode-se empregar várias técnicas e processos na construção do produto. Nesse framework é possível que pessoas desenvolvem e gerenciem projetos complexos dividindo em etapas e entregas a partir de regras e artefatos do Scrum. (Sutherland e Schwaber, 2013)

Sua essência assimila a paradas regulares ao começar o projeto, para analisar se o que está sendo feito está seguindo na direção certa, se é o que o cliente deseja ou até

mesmo analisar melhorias para que se desenvolva rapidamente, melhorar os resultados e verificar quais obstáculos que impedem o avanço da entrega do produto. Esse ciclo é chamado de inspeção e adaptação, que é uma ideia simples mas exige honestidade, reflexão, introspecção e disciplina. (Sutherland, 2014)

Para que a ideia acima seja cumprida, o Scrum visa de regras e artefatos para a transparência e organização seguindo a partir de um time Scrum.

## 2.1 O Time Scrum

A prioridade do time *Scrum*, é que ele seja auto organizável escolhendo sempre a melhor tomada de decisão no desenvolver de suas tarefas.

Esse time garante uma entrega incremental 'pronto', de acordo com os requisitos daquela história, de uma forma que o projeto até ali desenvolvido esteja de uma forma funcional. Para ser eficaz deve conter de 5 (cinco) a 9 (nove) membros, dentre eles: *Product Owner*, o time de desenvolvimento e o *Scrum Master*. A seguir são representados os devidos papéis de cada membro (Machado, 2009):

- *Product Owner*: É o dono pelo projeto, sendo quem está pagando por ele ou o responsável pelo mesmo, tem as informações para que caminho o projeto deve chegar, suas prioridades e necessidades. Além de controlar e gerenciar as entregas, também é de sua responsabilidade deixar totalmente claro e visível seu objetivo ao time onde se origina na lista do *Product Backlog*.
- *Scrum Master*: Responsável por gerenciar e ser transparente entre os membros sobre tudo que está ocorrendo durante o desenvolvimento do projeto. É quem marca as reuniões que visa o Scrum e garante que a comunicação do time esteja acessível. O *Scrum Master* também assegura que todos os ritos do Scrum estejam sendo cumprido e seus artefatos sendo usados corretamente. É seu papel também, remover qualquer empecilho que esteja bloqueando o andamento da equipe.
- Equipe: É que realiza a construção do projeto, sendo eles: de desenvolvimento e de teste, que tem a autoridade de tomar decisões necessárias e organizá-las para que cada tarefa seja executada no tempo estimado de cada *Sprint Backlog*. Não há hierarquias entre eles, cujo objetivo é ser auto gerenciável na distribuição de tarefas.

## 2.2 Eventos Scrum

Eventos são necessários para minimizar reuniões fora do planejamento do Scrum. Geralmente tem uma duração máxima para que nada seja discutido além do que foi definido, mas pode terminar a partir do momento que tudo esteja realmente esclarecido e definido. Cada evento é uma oportunidade de inspecionar e adaptar alguma coisa, por isso é preciso trazer todas as dúvidas para que possa ser esclarecido e com isso garantir transparência de acordo com a definição da reunião. (Sutherland e Schwaber, 2013)

Os eventos citados acima são: *Sprint Planning Meeting*, *Daily Meeting*, *Sprint Review* e *Sprint Retrospective*. É comum ser utilizado também, para ajudar na qualidade do desenvolvimento com o prazo de entrega, o *Burndown Chart*. A seguir será analisado cada um desses itens (Brod, 2015):

### 2.2.1 *Sprint Planning Meeting*

A princípio é realizada a *Sprint Planning Meeting*, ou seja, uma reunião para o planejamento da *Sprint*. Após a definição do projeto (objetivos e necessidades) pelo *Product Owner*, é feita uma lista de histórias contendo o que será necessário para o desenvolvimento do projeto. Geralmente é definido pontos sendo ressaltado na utilização da aplicação pelo usuário, sistema e empresa. Com essa lista pronta, o *Product Backlog* estará completo, sendo dividido em versões e em cada versão é gerada uma nova *Sprint Backlog*. A equipe define o que fará parte de cada *Sprint* de acordo com o *Product Backlog* sempre analisando as tarefas que irão conseguir entregar diante do prazo estimado que pode variar de 1 a 4 semanas, e as tarefas que sobraem fica para uma próxima *Sprint*. Esse é um dos pontos principais do *Scrum*, somente se comprometer a fazer as tarefas que consegue entregar no prazo estimado. Com a *Sprint* definida é realizado a divisão da responsabilidade de executar cada tarefa, o que não se limita quantas tarefas cada membro pode pegar. Nessa primeira reunião, é definido também o tempo que cada um vai demorar a executá-la, nessa etapa todos da equipe participam votando conforme sua opinião. Muitos times optam pelo jogo conhecido como *Planning*

*Poker*, onde se encontra cartas com pontuações e todos mostram uma ao mesmo tempo (pontuações definidas pela equipe) para simbolizar a dificuldade e tempo de cada *Sprint*, essas pontuações podem ser em cartas ou até mesmo um aplicativo que simbolizam cartas. Se as pontuações ficarem muito distantes de cada membro, e discutido as dificuldades até chegarem em um comum acordo.

### 2.2.2 Daily Meeting

Durante uma *Sprint*, o time se organiza a ponto de cada um realizar suas tarefas com prazos. O *Scrum Master* tem que garantir, que durante a execução, não haja nenhuma interferência externa, pois nada poderá tirar o foco dos objetivos estimados. Para garantir a qualidade nas entregas, são realizadas *Daily Meeting* (Reuniões diárias). Essas reuniões podem ser feitas presencialmente ou através de conferência, o ideal é sempre estar no mesmo lugar e na mesma hora, para que todos se adaptem seus horários, ela tem a duração de 15 a 20 minutos, passando a palavra a todos do time de desenvolvimento falar o que fizeram, o que vão fazer, os impedimentos e as dúvidas. De preferência é feito em frente ao quadro que mostra as tarefas que constam na *Sprint*, conforme o quadro tradicional mostrado na figura 1 ou na frente de um computador que mostra a ferramenta com as tarefas, conforme será abordado no próximo tópico.



Figura 1- Quadro de tarefas de uma Sprint.

Fonte: <https://realhosting.nl/2015/06/scrum/>

Costuma-se fazer na parte da manhã para que se inicie um novo dia de trabalho, como não há hierarquias na equipe esse é o momento de esclarecer todas as dúvidas que surgiram no dia anterior e deixar transparente para a equipe como está o andamento da *Sprint*, mas não é obrigatório ser nesse período. Quando houver impedimentos citados pelos membros da equipe, é papel do *Scrum Master* resolve-los rapidamente.

### **2.2.3 *Sprint Review***

A cada entrega, ou seja, de 1 a 4 semanas é feita a *Sprint Review* (Revisão da *Sprint*), para apresentar o produto desenvolvido na *Sprint* passo a passo com o sistema funcionando. Essa reunião pode contém além do time *Scrum*, o cliente, gerente e até engenheiros de outros projetos.

### **2.2.4 *Sprint Retrospective***

Após essa reunião de *Review*, já é feita a reunião *Sprint Retrospective*, ou seja, uma reunião que faz uma retrospectiva de tudo o que aconteceu na *Sprint*, indicando pontos positivos e negativos a serem melhorados, e lições aprendidas também, para apresentar se a equipe houve dificuldade em alguém parte da execução, sendo ela no desenvolvimento, comunicação ou transparência do produto. Essa reunião geralmente é feita com a equipe e o *Scrum Master*. É natural que as primeiras *Sprints* tenham vários erros cometidos pela equipe, mas é justamente com esse tipo de reunião que garante o acerto na próxima *Sprint*.

### **2.2.5 *Burndown Chart***

Ao final de cada *Sprint*, pode-se contar também com o *Burndown Chart*, que é artefato importante do *Scrum*, atuando no controle de qualidade. De forma simples, é estimado quantos dias tem uma *Sprint*, com um total de 100 pontos que serão

consumidos ao longo dos dias, sendo 20 pontos por dia a média de consumo correto, caso um dia se consumiu mais que 20, a ideia é recuperar no próximo dia. A partir desses dados é montado um gráfico para melhor visualização, que podem resultar em análises como: Exigiram mais esforços que os estimados, ocorreram algum fato que não precisasse consumir todos os pontos exigidos, etc.

### **3. Trello**

Há no mercado uma diversidade de ferramentas para o gerenciamento do produto a ser entregue, algumas visam em prazos, outras em gastos. No Scrum, como são definidos períodos curtos de entrega do produto ao cliente, necessita de uma ferramenta de simples funcionamento, onde a necessidade está apenas na organização e comunicação dos envolvidos. Porém A ferramenta Trello, disponibiliza de forma intuitiva e clara essa organização, utilizada em diversas empresas como Google, Adobe e Paypal, otimizando suas tarefas em uma organização transparente. (Trello, 2016)

O Trello é importante na parte de organização, para que nada seja perdido do quadro tradicional utilizado no Scrum, mas o essencial é quando uma equipe trabalha remotamente, pois sem esse tipo de ferramenta, não há comunicação entre as equipes e também as tarefas.

#### **3.1 Visão rápida sobre o Trello**

Essa ferramenta é gratuita e o processo de criação de uma conta é extremamente simples, com certeza não levará mais do que alguns segundos. Aos interessados, há ainda uma versão GOLD, que oferece algumas customizações visuais adicionais e um limite maior para upload de imagens e arquivos. A figura 2 ilustra sua estrutura e componentes:

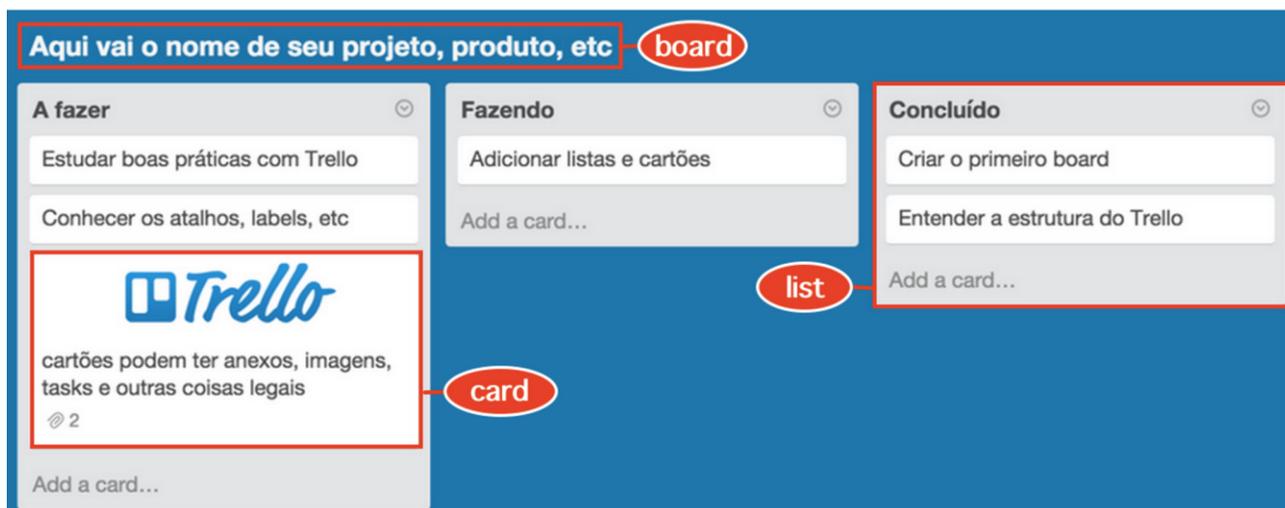


Figura 2 – Quadro de tarefas da Ferramenta Trello.

Fonte: <http://blog.caelum.com.br/encontrando-melhores-formas-trabalhar-com-o-trello/>

Conforme a figura 2, a parte 'board' é um quadro que é geralmente denominada 'Sprint 1' ou algum nome dado a *Sprint* com sua particularidade, 'list' são os status das suas tarefas e 'card' é uma tarefa do *Sprint*. Geralmente todas as tarefas começam na primeira lista e é passada para a segunda quando alguém membro da equipe começar a executá-la. Assim que essa tarefa já estiver totalmente feita e testada é passada para a última lista. Como o intuito do Trello é ser bem simples, para passar as tarefas de uma lista para outra é necessário somente arrastar e já está na lista desejada. Na metodologia Scrum, assim que um membro terminar de executar todas as tarefas ele poderá ajudar seu colega de trabalho a terminar a dele, nunca pegando uma tarefa de uma próxima *Sprint*.

A ferramenta também oferece o recurso de se comentar na tarefa criada, quando há algum detalhe ou impedimento, para que todos visualizem e nada fique no esquecimento, basta apenas clicar em cima da tarefa que abre uma aba, conforme figura 3 abaixo.

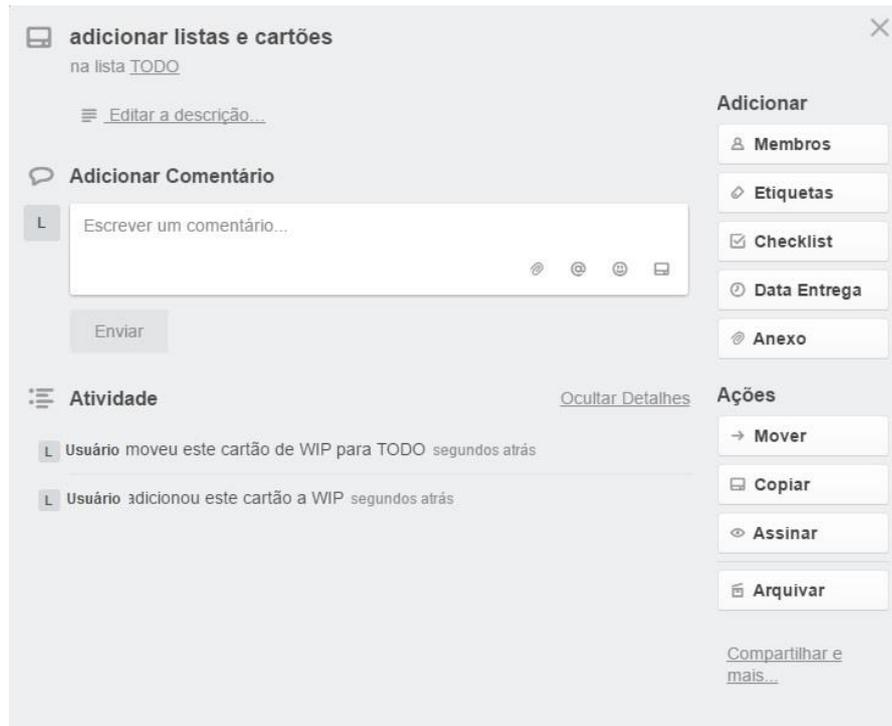


Figura 3 – Detalhamento de uma tarefa específica.

Fonte: Próprio Autor

Também é possível adicionar membros responsáveis pelas tarefas, assim o *Scrum Master* não tem a obrigação de ficar lembrando qual membro da equipe pegou qual tarefa. No campo Atividade é registrado quando um membro move sua tarefa para uma determinada lista. Uma das características que o Trello se sobressai, é o recurso de notificação, sendo possível marcar um membro para ver um comentário ou uma observação feita em determinada tarefa. Ou seja, o Trello oferece recursos suficientes para que tudo sobre a *Sprint* seja armazenada e assim diminuirá os riscos de erros por esquecimento. (Trello, 2016)

#### 4. Estudo de caso

Uma equipe de desenvolvimento junto com seu *Scrum Master* é selecionada para entrar em um projeto com metodologia *Scrum*, com entregas de *Sprint* de 2 semanas. Porém essa equipe trabalha *home-office* (trabalho realizado em ambiente externo) e com isso os membros precisam trabalhar a distância, deve-se utilizar uma ferramenta que dê suporte ao time, para que não seja perdida a essência *Scrum*. Esse projeto é sobre um site *E-Commerce* que comercializa serviços que estão em ofertas, mais conhecido como

site de compra coletiva. O *Product Owner* (PO) deixou claro na contratação do serviço sua real necessidade, e com isso foi marcado o primeiro evento do *Scrum*, a *Sprint Planning Meeting* (reunião de planejamento da *Sprint*). Todos eventos precisarão ser via vídeo conferência, devida a distância de cada um. Priorizando a transparência dos dados e a comunicação, será utilizado a ferramenta Trello, que por sua simplicidade disponibiliza através de seus recursos, tudo que é necessário para manter as regras e artefatos do *Scrum*.

No Primeiro evento, o PO descreve o objetivo do projeto informando também as necessidades que visa no momento e junto com o time, vai quebrando essas necessidades, formando o *Backlog Producter* de acordo com prioridades das entregas que será montado já no ambiente Trello, conforme figura 4.

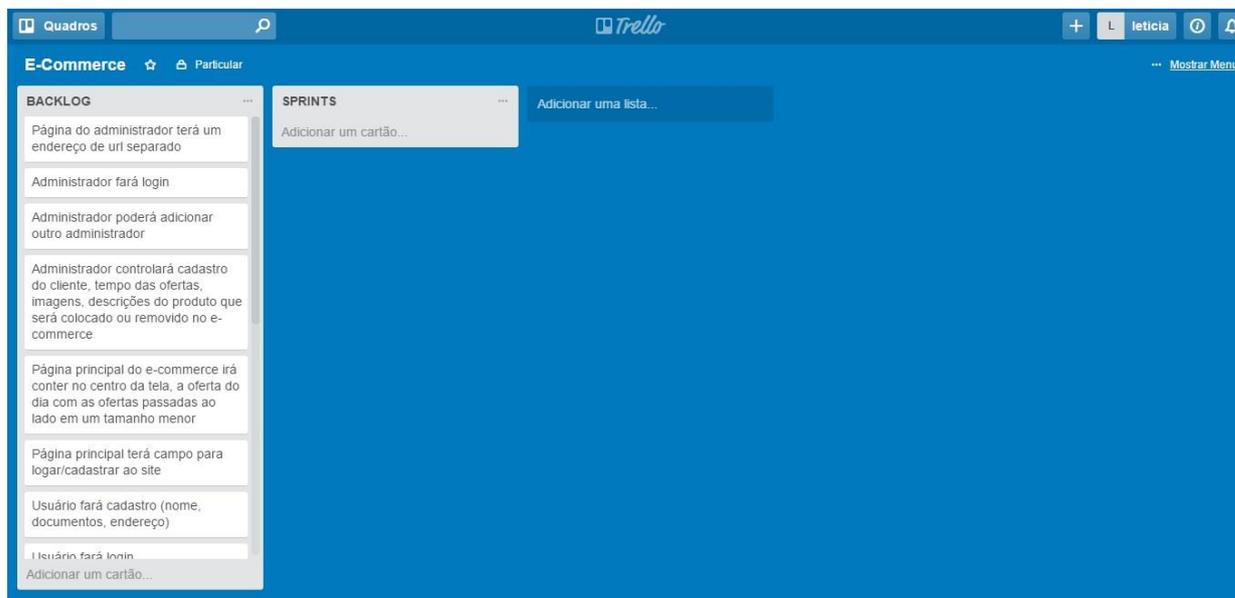


Figura 4: Lista do *Backlog*

Fonte: Próprio autor

Conforme figura 4 acima, foi criado a o projeto “E-Commerce” e nele as listas “backlog” onde o PO descreveu seus requisitos e “Sprints” onde essas estórias já foram sinalizadas em *Sprint* específica. Essas estórias que estão apenas no *backlog* pode ser alterada a cada entrega funcional para o cliente, onde poderá ver outras necessidades e incluir e retirar estórias.

Após a lista “Backlog” pronta, a equipe define estórias que farão parte da primeira *Sprint*, a *Sprint* 1. Essa análise é feita pelo time de desenvolvimento que irá pegar o

primeiro item da lista e analisar sua complexidade incluindo desenvolvimento e teste, através de um jogo chamado *Planning Poker*, onde será votado com cartas ao mesmo tempo (mesmo em vídeo conferência) pela equipe, conforme figura 5.



Figura 5: *Planning Poker*

Fonte: [http://www.museumsandtheweb.com/files/mitroff-fig3\\_0.jpg](http://www.museumsandtheweb.com/files/mitroff-fig3_0.jpg)

No caso acima (figura 5), houve dois votos de 2 pontos e 1 voto com 4 pontos, como não foi em total acordo, será colocado os pós e contras das duas partes, entrando sempre em um senso comum que nesse exemplo segue com 2 pontos e após tê-lo definido é arrastado para a lista “Sprints” e parte para a próxima estória até chegar em 21 a 23 pontos (essa meta geralmente é definida pela equipe dependendo da quantidade de membros da equipe de desenvolvimento). Ao terminar as votações concluindo os pontos necessários, os membros que são auto organizáveis vão ficando responsáveis pelas estórias que mais lhe convém com a capacidade técnica. É criado no Trello um novo quadro com todos os membros da equipe com o nome de *Sprint 1* e colocado somente as estórias escolhidas para essa *Sprint*, a partir dessas estórias, os responsáveis criarão tarefas para se organizar no desenvolvimento e teste da mesma, também colocá-las na ferramenta marcando membros envolvidos e comentários que possa ter surgido na reunião, as listas são divididas em “*Sprint*”, “*Wip*” onde ficarão as tarefas a serem feitas, o “*Todo*” que sinaliza o que está sendo feito no momento e o “*Done*” que são as tarefas finalizadas (Figura 6).

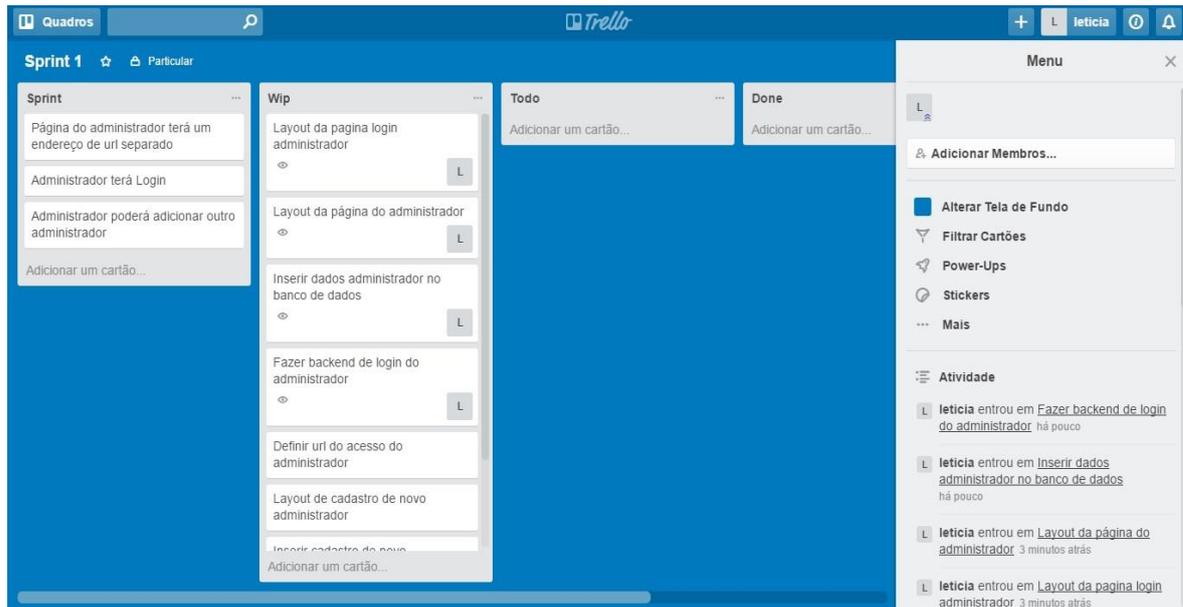


Figura 6: *Sprint 1* dividida em tarefas

Fonte: Próprio autor

Após toda *Sprint* planejada, a equipe encerra a reunião e o time de desenvolvimento começa se organizar para que já ocorra o desenvolvimento. Será realizada a *Daily Meeting* (reunião diária) as 09:00 horas remotamente (pelo fator da distância), horário definido por todos, para que o *Scrum Master* possa acompanhar o desenvolvimento da *Sprint* e remover qualquer impedimento da equipe, é possível o PO participar também se ele quiser. Apenas na *Daily* as tarefas são passadas para “Done”, mas as outras podem ser movidas em qualquer momento do dia e a reunião tem um tempo estimado de 15 minutos de duração, é somente para cada membro falar o que fez no dia anterior, o que vai fazer no dia de hoje, tirar dúvidas e se houver algum impedimento, passar para o *Scrum Master* resolvê-lo. Qualquer comentário feito em uma tarefa específica pode ser colocado pela ferramenta, de maneira simples, para que nada fique apenas na memória dos membros. Após a primeira *Daily* do time, o quadro do Trello ficou conforme a figura 7 para todos os membros da equipe.

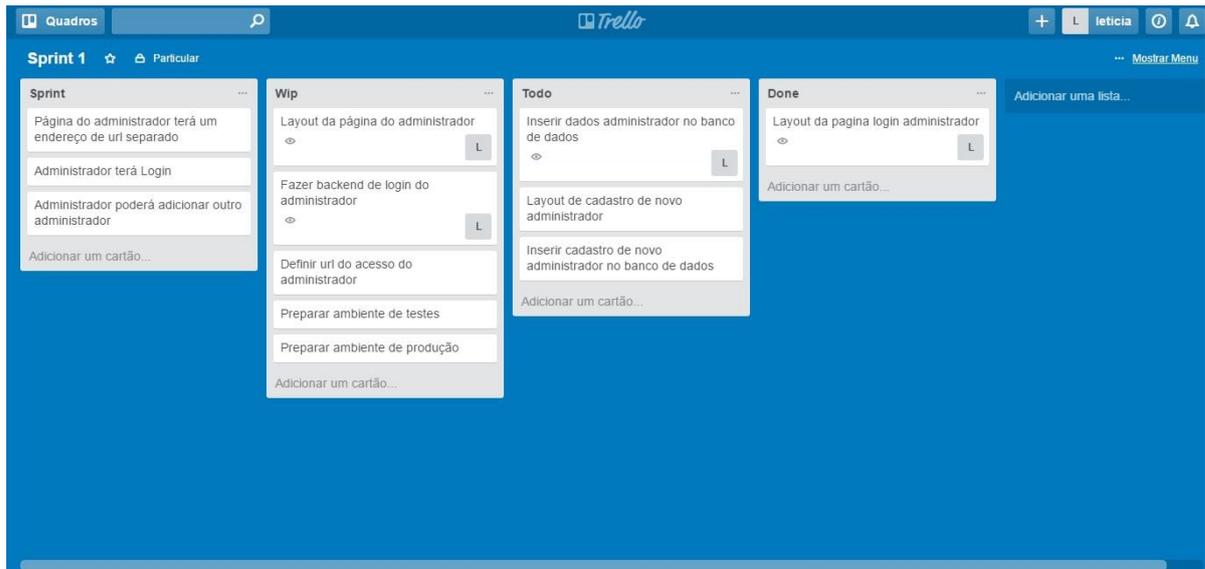


Figura 7: Organização de tarefas após a primeira *Daily*

Fonte: Próprio autor

Caso algum membro acabe suas tarefas antes do final da *Sprint*, ele ajudará algum colega a finalizar sua tarefa, na ferramenta Trello é possível marcar mais que uma pessoa responsável por tarefa e quando precisar destacar algo no comentário ou movida, todos os membros que foi marcado naquela tarefa específica, receberá notificações informando o acontecido.

Após o prazo estimado da *Sprint*, provavelmente todas as tarefas estejam concluídas, é realizado o próximo evento, a *Sprint Review* (Reunião de revisão) a ponto de vista do PO, onde será entregue a parte funcional do *software* de acordo com as histórias da *Sprint*. Nesse evento, o PO poderá ver se as funcionalidades são equivalentes ao solicitado nas histórias postas no *Sprint*.

O último evento da *Sprint* é a *Sprint Retrospective* (reunião de retrospectiva), onde é feito com a equipe de desenvolvimento e o *Scrum Master*. Esse evento é para evidenciar pontos positivos e negativos da equipe nesse período, impondo sempre melhorias nos negativos e conservando os positivos como um todo. O importante de relatar pontos negativos é para a equipe se aperfeiçoe rapidamente nas próximas *Sprints*, tornando-a mais madura e efetiva.

Após encerrar esse ciclo, é feita novamente o evento de planejamento da próxima *Sprint* e todos os eventos citados anteriormente até a entrega final do projeto.

## 5. Conclusão

O estudo de caso mostra com uma extensão curta e objetiva, os principais ritos e artefatos da metodologia ágil *Scrum*, que mostra que toda mudança é constante e é necessário que se adapte a ela. *Scrum* prepara de forma organizada tarefas à equipe para que se prevaleça o foco, a rapidez e a qualidade de um *software* funcional no seu prazo de entrega.

A utilização do *Scrum* é perfeita para equipes pequenas cujo projeto é incerto de requisitos. É necessária mudança de cultura na equipe, onde todos os membros estejam sempre focando em conhecimento sobre variados assuntos na área afim de nivelar novos conhecimentos. *Scrum* pode-se levar para qualquer área, sendo ela de *software*, vida pessoal, atividade doméstica, planejamento de festas e entre outros projetos. É possível leva-la a qualquer área e ter a mesma garantia do seu resultado com qualidade e eficiência.

Existem diversas ferramentas que auxiliam esse tipo de metodologia ágil, porém nem todas são gratuitas e fácil de manusear. O Trello dá suporte para os membros que mesmo distantes possam saber de tudo que se passa dentro da *Sprint* com ajuda de comentário, notificação e organização, marcando cada membro da equipe em sua tarefa específica. O Trello dá suporte a acesso remoto, transparência, facilidade e agilidade de utilizá-lo.

A usabilidade do *Scrum* com a ferramenta Trello, não tem limite para o tamanho de um determinado projeto, sendo que corresponde com a mesma qualidade usando-o em projetos simples ao mais complexos, quebrando seus requisitos em pequenas partes para que todos trabalhem o necessário para uma entrega e nada além disso. De diversas maneiras mencionadas acima, pode-se concluir que a metodologia e ferramenta abordada nesse artigo mostra que seu potencial resulta na qualidade da entrega de um produto funcional, sem precisar de burocracia e limitações sobre áreas e complexidade de seus projetos.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Antonio José F. Andrade, J. C. (2012). *Artigo Scrum*. Fonte: EnuComp - Encontro Unificado da Computação:  
<http://www.enucomp.com.br/2012/conteudos/artigos/scrum.pdf>
- Augusto, M. V. (2011). *Desenvolvimento de Software com Apoio de Prática Scrum*. Fonte: Fatec SP: <http://www.fatecsp.br/dti/tcc/tcc0011.pdf>
- Brod, C. (2013). *Livro Scrum - Guia prático para projetos ágeis*.
- Cruz, F. (2015). *Livro Scrum e Agile em Projeto*.
- Desenvolvimento Ágil Scrum*. (s.d.). Fonte: Desenvolvimento Ágil:  
<http://www.desenvolvimentoagil.com.br/scrum/>
- Leitão, M. d. (Junho de 2010). *Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo*. Fonte: Ecomp:  
[http://tcc.ecomp.poli.br/20101/TCC\\_final\\_Michele.pdf](http://tcc.ecomp.poli.br/20101/TCC_final_Michele.pdf)
- Machado, M. (2009). *SCRUM – Método Ágil*. Fonte: Faculdade do Guarujá:  
[http://www.faculadadedoguaruja.edu.br/revista/downloads/edicao12009/Artigo\\_5\\_Prof\\_Marcos.pdf](http://www.faculadadedoguaruja.edu.br/revista/downloads/edicao12009/Artigo_5_Prof_Marcos.pdf)
- Martin, R. C., Beck, K., Cunningham, W., Hunt, A., Thomas, D., Beedle, M., . . . Sutherland, J. (30 de 11 de 2016). *Manifesto Agile*. Fonte: Manifesto Agile:  
[https://msdn.microsoft.com/pt-br/library/dd997578\(v=vs.120\).aspx](https://msdn.microsoft.com/pt-br/library/dd997578(v=vs.120).aspx)
- Oliveira, E. S. (Fevereiro de 2003). *Uso de Metodologias Ágeis no Desenvolvimento de Software*. Fonte: Cpdee Ufmg:  
<http://www.cpdee.ufmg.br/~renato/TesesEDissertacoesOrientadas/Monografia-EbenezerSilvaOliveira.pdf>
- Publicada nova versão do Guia do Scrum*. (2016). Fonte: Infoq:  
[https://www.infoq.com/br/news/2016/11/scrum-guide-2016?utm\\_source=news\\_about\\_scrum&utm\\_medium=link&utm\\_campaign=scrum](https://www.infoq.com/br/news/2016/11/scrum-guide-2016?utm_source=news_about_scrum&utm_medium=link&utm_campaign=scrum)
- Roberto Hernández Sampieri, C. F. (2013). *Livro Metodologia de Pesquisa*.
- schwaber, k., & sutherland, j. (2013). *O guia do Scrum*. Fonte: Scrum Guides:  
<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>
- Scrum com Trello*. (s.d.). Fonte: Mind Master: <http://www.mindmaster.com.br/scrum-com-trello/>
- Soler, A. M. (2015). *Gerenciamento de Projetos em tirinhas*.
- Sutherland, J. (2014). *Scrum, a arte de fazer o dobro com a metade do tempo*. Leya.
- Trello*. (s.d.). Fonte: Trello: <https://trello.com/>