



VII SINGEP

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

ISSN: 2317-8302

A UTILIZAÇ3O DO FRAMEWORK SCRUM EM UM PROJETO DE INDICADORES DE NEG3CIOS

ALINE PITONDO MONTEIRO

Centro Estadual de Educa3o Tecnol3gica Paula Souza - CEETPS

MARCELO T OKANO

Centro Estadual de Educa3o Tecnol3gica Paula Souza - CEETPS



A UTILIZAÇÃO DO FRAMEWORK SCRUM EM UM PROJETO DE INDICADORES DE NEGÓCIO.

Resumo

O presente trabalho tem por objetivo analisar, por meio de um estudo de caso, como a utilização do framework ágil Scrum, empregado em projetos de desenvolvimento de tecnologias de sistemas, foi utilizado com sucesso durante os processos de gestão e execução de um projeto de melhoria de um indicador chave de negócios em uma empresa de telecomunicações, trabalhando exclusivamente por meio da melhoria de processos e da criação de ferramentas paliativas externas aos sistemas de gestão da empresa. A metodologia usada foi um estudo de caso baseado em uma observação direta intensiva de um projeto para redução de chamadas no SAC (Serviço de Atendimento ao Consumidor) por motivos financeiros, que a autora participou como gerente e executora entre os meses janeiro de 2016 e dezembro de 2017. O resultado apresentado ao final do estudo de caso demonstra que o framework Scrum pode ser utilizado com sucesso em projetos de negócios, não ficando restrito ao desenvolvimento de softwares e outros elementos de tecnologia da informação.

Palavras-chave: *Scrum*. Ágil. Processos. Projeto. Indicadores.

Abstract

The purpose of this paper is to analyse, through a case study, how the Scrum framework, used in systems technology development projects, was successfully used during the management and execution of a project to improve a key business indicator in a telecommunication's company, working exclusively by improving processes and creating palliative tools external to the company's management systems. The methodology used was a case study based on an intensive direct observation of a project which the objective was reduce calls for financial reason in the Customer Attendance Service, which the author participated as manager and executor and that took place between January 2016 and December 2017. The result presented at the end of the case study demonstrates that the Scrum framework can be used successfully in business projects, not restricted to the development of software and other elements of information technology.

Keywords: Scrum. Agile. Processes. Project. Indicator.



1 Introdução

O gerenciamento ágil de projeto está em voga nos dias atuais, sendo muito utilizado para o desenvolvimento de *software*. Ele preza por “satisfazer o cliente, através da entrega adiantada e contínua de *software* de valor” (Manifesto Ágil, 2001).

Por trabalhar de forma interativa, com times pequenos e multifuncionais, esse tipo de gerenciamento tende a dar resultados rápidos, com possibilidade de mudanças na mesma velocidade.

Como sua origem é no desenvolvimento de *software*, nesse nicho de mercado *frameworks* e métodos de trabalho são bem conhecidos e cada vez mais aplicados, principalmente no mercado de internet e "apps", em que a evolução e adaptação ao usuário precisam ser rápidas e constantes.

Hoje existem associações e uma série de certificações que atestam que esta metodologia ou maneira de gerir projetos está consolidada no mercado e, com isso, *frameworks* de gestão de projetos ágeis começam a ganhar espaço em outras áreas que não possuem seu foco em desenvolvimento.

O objetivo deste trabalho é demonstrar a aplicação de técnicas e cerimônias da gestão ágil, mais especificamente, o *Scrum*, em um projeto que tinha como objetivo a melhora de um indicador operacional por meio de melhoria de processos de negócios e gestão da qualidade dos dados, ativos de processos organizacionais e regulamentares da empresa.

2 Introdução

Todo o *Scrum* tem suas definições, regras, eventos e artefatos definidos no “*The Scrum Guide*”, que também possui uma versão em português e é disponibilizado no site "Scrum.org".

Além do guia, serão utilizados livros e artigos de apoio que possam complementar o texto, incluindo, quando necessário, explicações sobre técnicas e ferramentas não definidas no guia, porém, utilizadas no mercado por usuários do *Scrum*.

2.1 O *Scrum*

O *Scrum* foi criado para desenvolver, entregar e manter produtos complexos, sendo esses produtos normalmente *softwares*.

A primeira apresentação formal sobre *Scrum* foi em 1995, na Conferência OOPSLA, por Ken Schwaber e Jeff Sutherland. Após essa apresentação os mesmos autores escreveram e hoje mantém o “*The Scrum Guide*” (Guia do *Scrum*) e, apesar de hoje existirem vários livros, artigos e apresentações sobre o assunto, todos eles se referenciam nesse guia, que, em sua versão traduzida para o português, possui somente 20 páginas.

Hoje existem duas grandes comunidades que se dedicam à disseminação e prática do *Scrum*, a "*Scrum.org*" e a "*Scrum Alliance*", ambas provêm certificações diversas, sendo as mais conhecidas no mercado as de *Scrum Master* (PSM e CSM) e as de *Product Owner* (PSPO e CSPO).

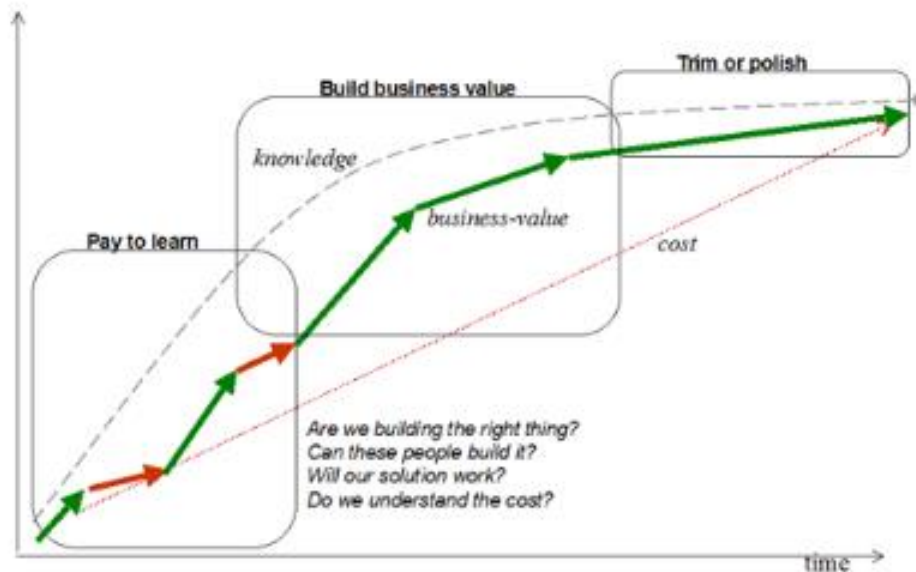


Figura 2: Curva de Aquisição do Conhecimento
Autor: Alistair Cockburn

O *Scrum* é considerado um *framework*, com papéis, eventos, artefatos e regras com propósitos específicos e que são considerados essenciais para implantação, uso e sucesso da técnica. Por essas razões muitos sites, profissionais e acadêmicos se referem ao *Scrum* como uma metodologia devido as possibilidades limitadas de *tailoring*. No próprio guia oficial há esta conclusão:

“O *Scrum* é livre e oferecido neste guia. Papéis, eventos, artefatos e regras do *Scrum* são imutáveis e embora seja possível implementar somente partes do *Scrum*, o resultado não é *Scrum*. *Scrum* existe somente na sua totalidade e funciona bem como um container para outras técnicas, metodologias e práticas” (SCHWABER; SUTHERLAND, 2017)

O *Scrum* tem em sua essência um pequeno time de pessoas (de 3 a 9 pessoas), que são altamente flexíveis e adaptativas, além de um *Scrum Master* e um *Product Owner*.

Fundamentado nas teorias empíricas de controle de processo, isto é, “o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido” (*The Scrum Guide*), o *Scrum* utiliza uma abordagem iterativa e incremental, buscando sempre aperfeiçoar a possibilidade de previsão e o controle de riscos. Para isso, o *framework* se apoia em 3 pilares: Transparência, Inspeção e Adaptação.

- **Transparência:**

A transparência requer que aspectos significativos tenham uma definição padrão comum e estejam sempre visíveis aos responsáveis.

- **Inspeção:**

A inspeção dos artefatos do *Scrum* é feita frequentemente pelos próprios usuários para detectar variações indesejadas durante a *Sprint*, porém as inspeções mais eficientes se feitas por inspetores especializados.



- Adaptação

Se um ou mais aspectos de um dos processos desviou para fora dos limites aceitáveis e o resultado do produto não será aceitável, o processo ou material sendo produzido deve ser ajustado o mais rápido possível.

Para esses pilares serem alcançados e seguidos, o *Scrum* têm quatro eventos formais para inspeção e adaptação: Planejamento da *Sprint*, Reunião Diária, Revisão da *Sprint* e Retrospectiva da *Sprint*.

Importante ressaltar que o gerenciamento ágil, independente se pelo *Scrum* ou outro *framework* da família, há uma característica de melhor funcionar em ambientes mais complicados e complexos, pois devido sua característica fortemente voltada para a aceitação e implementação de mudanças, os requisitos podem ser alterados sem grande burocracia. O livro do PMI para metodologias ágeis, o *Ágile Practice Guide*, ilustra bem as melhores condições de aplicação do ágil (ver figura 3).

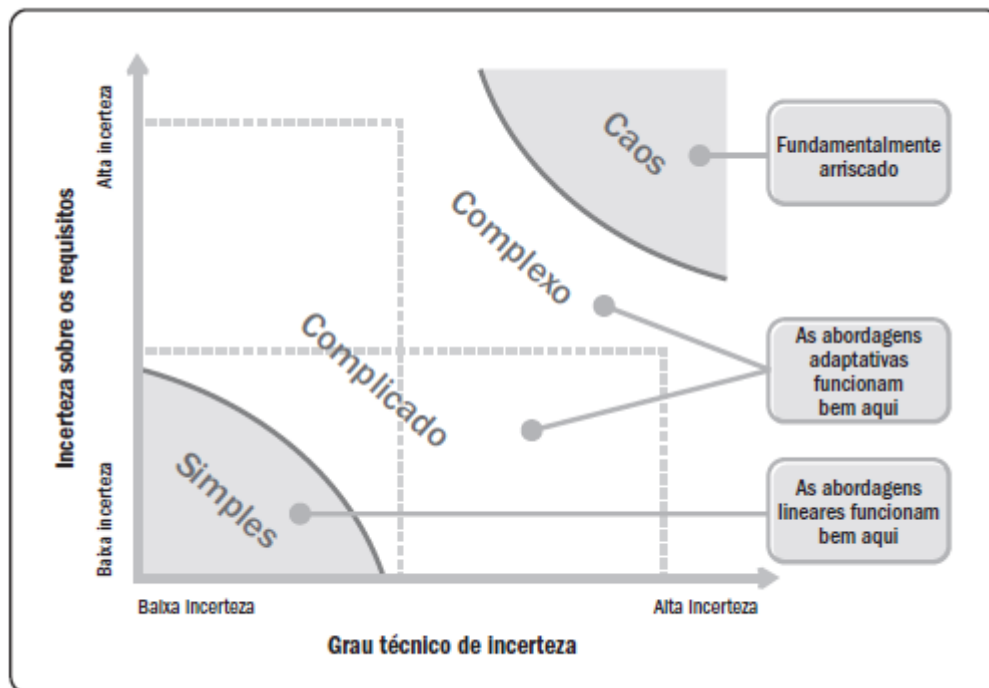


Figura 3: Modelo de Incerteza e Complexidade Inspirado no Modelo de Simplicidade de Stacey.

Fonte: PMI - *Ágile Practice Guide* (2017)

2.1.1 Time Scrum

Os Times *Scrum* são caracterizados por serem times pequenos, auto-organizáveis e *cross*-funcionais (*squad*¹), para aperfeiçoar a flexibilidade, criatividade e produtividade. Esses times têm o comprometimento em entregar produtos de forma iterativa e incremental, garantindo que uma versão potencialmente funcional do produto do trabalho esteja sempre

¹*Squad* é um time *cross*-funcional, isto é, com membros de diversas áreas e/ou especialidades, que possuem autonomia para definir prioridades e capacidade para desenvolver todos os aspectos de um projeto do início ao fim.



disponível.

Esses times sempre são formados por: 01 *Product Owner*, 01 *Scrum Master* e o Time de Desenvolvimento.

- *Product Owner*:

O *Product Owner* é a única pessoa responsável por gerenciar o *backlog* do produto. É esse profissional que define as prioridades dos itens que o time de desenvolvimento deve trabalhar e conhece bem os requerimentos de negócios que a solução deve atender. É também ele que decide se o incremento pronto será liberado em produção ou não.

- *Scrum Master*:

O *Scrum Master* é o guardião dos eventos e artefatos do *Scrum*, é ele o responsável por certificar que todos no time compreendem a teoria, as regras, as práticas e os valores do *Scrum*. Além disso, ele é responsável também por fazer com que os *stakeholders* que estão fora do time *Scrum* entendam quais interações junto ao time *Scrum* são úteis e quais não são.

Esse profissional trabalha em conjunto com o *Product Owner*, com o Time de Desenvolvimento e com a Organização, garantindo a disseminação da prática *Scrum* e colaborando para que haja uma maximização dos valores entregues pelo time de desenvolvimento, eliminando barreiras e impedimentos que possam afetar de alguma maneira essas entregas.

- Time de Desenvolvimento:

São pequenos times, recomenda-se entre de 03 a 09 pessoas, responsáveis por entregar um incremento “Pronto”.

Os profissionais do time de desenvolvimento não possuem títulos e o time é multifuncional o suficiente para entregar um incremento potencialmente liberável no final de cada *Sprint*.

O time de desenvolvimento é auto-organizado e estruturado e autorizado pela empresa para gerenciar e organizar o próprio trabalho, ninguém, nem mesmo o *Scrum Master* diz ao time de desenvolvimento como transformar o *Backlog* do Produto em incrementos de funcionalidades potencialmente liberável.

2.1.2 Eventos *Scrum*

Podemos dizer que os eventos utilizados no *Scrum* são a alma do *framework*. Eles são prescritos para reduzir a necessidade de reuniões não definidas no *Scrum*. Todos eles têm tempo pré-definido (*time boxed*) de duração que não pode ser ultrapassado, porém, exceto pela *Sprint*, todos os outros eventos podem terminar assim que o objetivo é alcançado.

- *Sprint*

O ponto mais marcante dentro do *Scrum* são as *Sprints*. *Sprints* (que em português quer dizer iteração) são períodos (*time-boxed*) de um mês ou menos durante o qual um incremento é criado e durante todo o projeto elas procuram manter o mesmo prazo, que é definido no início do projeto. De acordo com o guia, essa estratégia é importante porque



se o horizonte da *Sprint* for muito longo seu escopo pode mudar. Por consequência, a complexidade de desenvolvimento pode aumentar e o risco da falta de aderência do incremento ao negócio pode crescer. Além disso, o prazo curto limita o risco do custo ao período definido.

Neste ponto vale uma observação, o guia oficial do *Scrum* afirma que não existe *Sprint Zero*, porém, ela é muito utilizada pelos usuários do *Scrum* a fim de criar uma visão e compreensão geral do produto, que é importante para se tangibilizar o esforço de trabalho necessário. Nesta *Sprint*, com duração usual entre uma e duas semanas, o time de desenvolvimento, com uma visão mais completa das necessidades do produto, define o tempo necessário para a criação de incrementos a partir dos esforços de trabalho mapeados para, assim, definir o tempo padrão das demais *Sprints*.

No período de uma *Sprint*, não são realizadas mudanças que possam comprometer o objetivo da *Sprint*. As metas de qualidade não reduzem e, a qualquer momento dentro da *Sprint*, o escopo pode ser renegociado e melhor entendido por meio de uma interação entre o *Product Owner* e o time de desenvolvimento.

Uma *Sprint* pode ser cancelada somente pelo *Product Owner* e isto acontece principalmente quando o objetivo da *Sprint* se torna obsoleto, seja por razões de mercado, de tecnologia ou organizacionais, porém, como uma *Sprint* é curta, raramente cancelamentos têm sentido.

- Planejamento da *Sprint* (*Sprint Planning*)

O planejamento do trabalho em uma *Sprint* é feito com todo o time *Scrum* em um evento *time-boxed*, com no máximo oito horas de duração (para *Sprints* de um mês). O *Scrum Master* garante que todos compreendam o seu propósito e atinjam seus objetivos dentro do prazo proposto.

No planejamento da *Sprint* se responde as seguintes questões:

1. O que pode ser entregue como resultado do incremento da próxima *Sprint*?
2. Qual é o trabalho necessário para entregar o incremento será realizado?

As entradas dessa reunião são o *Backlog* do Produto, o mais recente incremento do produto, a capacidade projetada do Time de Desenvolvimento durante a *Sprint* e o desempenho passado do Time de Desenvolvimento. Somente o Time de Desenvolvimento pode definir o que será completo ao longo da *Sprint*, porém, é o time *Scrum* quem define qual será a meta da *Sprint* e quais os requisitos necessários para que essa meta seja considerada como pronta.

Um fator primordial nesta reunião é que ao fim do planejamento o Time de Desenvolvimento seja capaz de explicar ao *Product Owner* e ao *Scrum Master* como pretende trabalhar para completar o objetivo da *Sprint* e entregar o incremento pronto.

- Meta da *Sprint* (*Sprint Goal*)

Toda *Sprint* tem um objetivo/meta que deve ser atendido ao seu final, que, como dito anteriormente, é definido durante o planejamento da *Sprint*. Para atingir esse objetivo um dos meios é a utilização do *Backlog* do Produto, que fornece uma direção ao time de desenvolvimento do porquê construir o incremento, além de permitir que o Time de



Desenvolvimento trabalhe em conjunto, em vez de utilizar iniciativas individuais ou separadas.

A Meta da *Sprint* pode ser negociada junto ao *Product Owner* caso o Time de Desenvolvimento perceba que o esforço de trabalho necessário é diferente do esperado e percebido durante o Planejamento da *Sprint*.

- Reunião Diária (*Daily Scrum*)

Ponto marcante dentro do *framework Scrum* são as Reuniões Diárias de não mais do que 15 minutos (*time-boxed*) para o Time de Desenvolvimento. Como o próprio nome diz, ela é realizada todos os dias e sempre no mesmo horário e local para reduzir a complexidade.

Nesta reunião é planejado o trabalho para as próximas 24 horas e, também, é uma janela de inspeção do trabalho desde a última reunião, aumentando a probabilidade da Meta da *Sprint* ser atingida. O Guia do *Scrum* sugere que sejam respondidas três perguntas neste evento:

1. O que eu fiz ontem que ajudou o Time de Desenvolvimento a atingir a meta da *Sprint*?
2. O que eu farei hoje para ajudar o Time de Desenvolvimento atingir a meta da *Sprint*?
3. Eu vejo algum obstáculo que impeça a mim ou o Time de Desenvolvimento no atingimento da meta da *Sprint*?

Após este evento, é frequente que os membros do Time de Desenvolvimento se encontrem imediatamente após para discussões mais detalhadas ou para adaptar ou para replanejar o restante da *Sprint*.

O *Scrum Master* é responsável pelo acontecimento dessa reunião dentro dos parâmetros definidos, porém, sua presença não é obrigatória.

- Revisão da *Sprint* (*Sprint Review*)

A Revisão da *Sprint* tem uma duração máxima de 4 horas para uma *Sprint* de um mês. Ela é realizada sempre ao final da *Sprint* para inspeção do incremento e adaptação do backlog se necessário.

Nessa reunião há a presença de todo o *Time Scrum* e das principais partes interessadas com a finalidade de revisitar o *Backlog* do Produto para otimizar o valor a ser entregue nas próximas *Sprints* com base no que foi feito na *Sprint* que está sendo finalizada.

É uma reunião informal que prevê o recebimento de *feedback* e promoção de colaboração entre o *Time Scrum* e os principais *Stakeholders*. Além disso, é nessa reunião também que *Product Owner*, se necessário, projeta prováveis entregas e datas alvo baseado no progresso até a data.

- Retrospectiva da *Sprint* (*Sprint Retrospective*)

O objetivo desse evento é que o *Time Scrum* tenha uma oportunidade formal de inspecionar a si próprio e definir, se necessário, planos de ações de melhorias que devem ser aplicadas na próxima *Sprint*. Seu propósito é:



1. Inspecionar como a última *Sprint* foi em relação às pessoas, aos relacionamentos, aos processos e às ferramentas
2. Identificar e ordenar os principais itens que foram bem e as potenciais melhorias
3. Criar um plano para concretizar melhorias no modo que o Time *Scrum* faz seu trabalho

Essa reunião tem a duração de até 3 horas para *Sprints* de um mês e ocorre entre a Revisão da *Sprint* e o planejamento da próxima *Sprint*.

- Refinamento do *Backlog* (*Grooming*)

Apesar do Guia do *Scrum* não citar o Refinamento do *Backlog* como um evento dentro do *Scrum* e sim como um processo contínuo dentro da administração do *Backlog* do Produto, é uma prática de mercado que essa reunião aconteça em alguns momentos durante o projeto. Alguns *Agile Coaches* indicam que ela seja realizada anterior à Revisão da *Sprint* de cada *Sprint*, mas não há uma unanimidade sobre o tema, assim como não há um acordo sobre a necessidade de todo o Time *Scrum* participar, apesar de ser essa uma boa prática.

De acordo com Cesar Brod (2015), o objetivo dessa reunião é aprimorar o *Product Backlog*, dando oportunidade aos membros do time de fazer perguntas que normalmente surgem durante o Planejamento da *Sprint*. Além disso, o Refinamento do *Backlog* também envolve itens como a descoberta de novos itens, a alteração ou remoção de itens existentes, a divisão de histórias de usuário, a priorização de itens, definição de critérios de aceitação, entre outros.

Mesmo não estando dentro dos rituais oficiais, este evento costuma ser *time-boxed* de acordo com a duração da *Sprint*, não ultrapassando 10% da capacidade do Time de Desenvolvimento.

2.1.3 Artefatos *Scrum*

É representado pelo trabalho ou o valor para fornecimento de transparência e oportunidades de inspeção e adaptação. Esses artefatos são “projetados para maximizar a transparência das informações chave de modo que todos tenham o mesmo entendimento” (Guia do *Scrum*, 2017).

- *Backlog* do Produto (*Product Backlog*)

É uma lista única ordenada de tudo que se sabe ser necessário para o produto e a única origem de seus requisitos. Enquanto o produto existir, o *Backlog* do Produto também irá existir, pois nele está lista todas as características, funções, requisitos, melhorias e correções que formam as mudanças que devem ser feitas no produto nas futuras versões. Os itens que estão nesta lista possuem os atributos de descrição, ordem, estimativa e valor, bem como descrições de testes que comprovarão sua completude quando “Prontos”. Mesmo que haja mais de um Time *Scrum* trabalhando em um mesmo produto, esta lista será única com os itens de ordem mais alta (topo da lista) mais claros e detalhados que os itens de ordem mais baixa.



O Guia do Scrum não cita as Histórias de Usuário (*User Stories*) como padrão do framework e deixa a maneira de escrever o *backlog* livre para ser adaptado e escrito conforme a necessidade do Time *Scrum*, porém, é comum entre os usuários do Scrum a utilização das mesmas para representar os itens do Backlog do Produto.

Mike Cohn (2011) exemplifica em seu livro como as Histórias de Usuário devem ser curtas e simples e sempre de um recurso narrado do ponto de vista da pessoa que deseja o novo recurso, geralmente um usuário ou cliente do sistema. Essas histórias usualmente seguem um modelo simples: *Eu, como um <tipo de usuário>, quero <algum objetivo> para que <algum motivo>*.

O *Product Owner* é responsável pelo *Backlog* do Produto, incluindo seu conteúdo, disponibilidade e ordenação, porém, é o Time de Desenvolvimento que é responsável por todas as estimativas de esforço de trabalho. O *Product Owner* ajuda o Time de Desenvolvimento deixando claro o entendimento, as necessidades e as expectativas de cada item do Backlog do Produto.

- Backlog da *Sprint* (*Sprint Backlog*)

É o conjunto de itens selecionados do *Backlog* do Produto para a *Sprint*, juntamente com o plano para entregar o incremento do produto e atingir o objetivo da *Sprint*.

O *Backlog* da *Sprint* torna visível todo o trabalho que o Time de Desenvolvimento identifica como necessário para atingir o objetivo da *Sprint* e sempre que um novo trabalho é necessário este é adicionado ao *Backlog* da *Sprint*. Para garantir melhoria contínua, é incluído no mínimo um item de prioridade alta sobre melhoria do processo identificado na última Reunião de Retrospectiva.

- Incremento (*Done*)

São os itens do *Product Backlog* que foram completados durante a *Sprint* juntamente com o valor dos incrementos de todas as *Sprints* anteriores.

O incremento deve estar em condição de ser utilizado independentemente de o *Product Owner* decidir liberá-lo em produção ou não.

- Definição de “Pronto” (*Definition of Done*)

Todos dentro do Time *Scrum* deve ter um entendimento comum do que significa o trabalho estar completo, “Pronto”. Esta é a “Definição de Pronto” para o Time *Scrum* e é usado para assegurar quando o trabalho está completo no incremento de funcionalidades do produto e potencialmente liberável.

Essa definição pode ser definida por todos os Times *Scrum* que estão trabalhando em um mesmo produto em conjunto, ou caso isso não ocorra e a empresa possua convenções, padrões ou diretrizes de desenvolvimento da organização, todos os Times *Scrum* devem segui-la como um mínimo.

2.2 Manifesto Ágil

O Manifesto Ágil é a origem de todas as metodologias ágeis que hoje existem, inclusive o *Scrum*. Este manifesto foi publicado em 2001 por 17 autores e determina em linhas gerais o que se espera de uma metodologia ágil para desenvolvimento de *software*.



Logo na página inicial é deixado claro o que é valorizado dentro desta maneira de gerir:

- Indivíduos e interações 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.

E esses valores se tornam doze princípios que acabaram por guiar o *framework* do *Scrum* e de todos os outros *frameworks*, como anteriormente dito, a saber: *Extreme Programming*, *DSDM*, *Adaptive Software Development*, *Crystal*, *Feature-Driven Development*, *Pragmatic Programming* e outras.

Anterior ao Manifesto, *frameworks* ágeis eram conhecidos como processos “*Light*” ou “*Lightweight*”, porém, isso não refletia a verdade, que, apesar dos *frameworks* serem curtos, eram difíceis de aplicar e perdurar na empresa, devido à maneira disruptiva de cada um desses movimentos tratarem métodos mais tradicionais.

3 Metodologia da Pesquisa

O trabalho foi realizado por meio de um estudo de caso, baseado em uma observação direta intensiva, de um projeto (denominado de “x”) para redução de chamadas no SAC (Serviço de Atendimento ao Consumidor) por motivos financeiros em uma empresa de telecomunicação (denominado de “y”), que a autora participou como gerente e executora e que aconteceu entre os meses janeiro de 2016 e dezembro de 2017.

Estudo de caso ou método monográfico é um procedimento de pesquisa criado por Le Play (1830), que utilizou o método para estudar famílias operárias na Europa. Esse tipo de pesquisa consiste no estudo de um caso de forma profunda considera que o mesmo pode ser representante de casos semelhantes ou ter alto valor didático.

O que caracteriza um estudo de caso é a investigação de um fenômeno em condições contextuais da vida real, com poucos pontos de dados e mais variáveis de interesse.

De acordo com Yin (2001), existem, no mínimo, cinco aplicações diferentes em pesquisas de avaliação de estudos de casos. No decorrer do trabalho a abordagem escolhida utilizada será de ilustrar alguns tópicos dentro da avaliação de modo descritivo, que ainda não é sistematizado, a fim de responder a questão de pesquisa discutida no decorrer do trabalho.

Foi um estudo de caso único, que verificou a utilização de um *framework* (*Scrum*), que é utilizado no desenvolvimento ágil de *software*, em um projeto de negócios, com foco em melhoria de processos e ferramentas paliativas, fora do sistema corporativo da empresa, para diminuição de erros e falhas no sistema de faturamento da mesma organização (consultar figura 01).

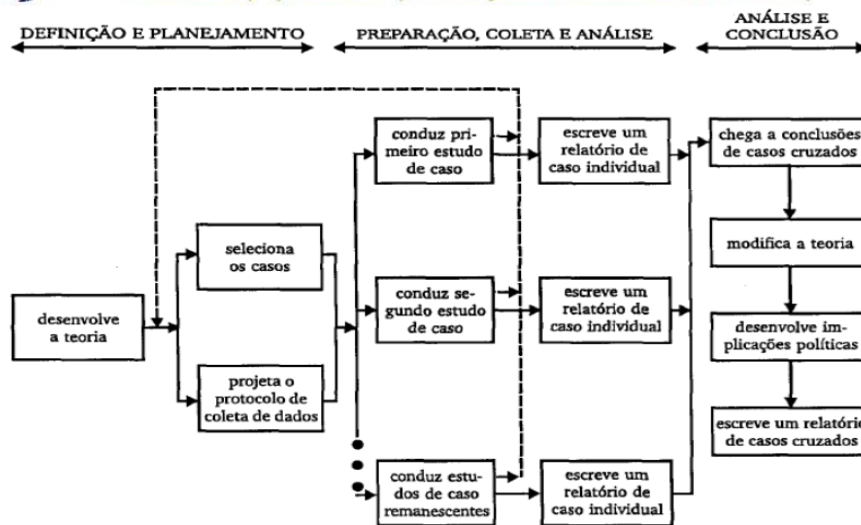


Figura 01 – Método de Estudo de Caso
Fonte: Yin (2001)

4 Estudo de Caso

Devido à natureza do projeto, em que resultados eram refletidos em até 90 dias devido ao ciclo de faturamento e a necessidade de resultados rápidos, foi definido, pela consultoria contratada para o planejamento do projeto, que o melhor meio de gestão seria o ágil, por meio do *framework Scrum*, mesmo este sendo normalmente utilizado no desenvolvimento de *software* e não em melhorias de processos e criação de ferramentas.

4.1 Cenário Geral

No início de 2016, a empresa “Y” estava com níveis muito acentuados de chamadas no SAC por todos os motivos que estava causando um alto custo operacional. Além do custo, havia uma suspeita que o alto volume de ligações impactava diretamente na queda do número de clientes em sua base de clientes.

Diante desse cenário, a empresa decidiu contratar uma consultoria de negócios para criar indicadores que pudessem prever a saída de clientes. O resultado do trabalho dessa consultoria foi a confirmação que quanto mais chamadas o cliente realizasse ao SAC da empresa, considerando somente ligações que foram atendidas por humanos, maior a possibilidade de cancelamento intencional (*churn* voluntário) de produtos e serviços da empresa.

Com esse ponto de partida, os executivos decidiram criar “*War-Rooms*”, com equipes multidisciplinares para cada opção principal da URA disponível ao cliente (financeiro/contas, problemas técnicos, informações sobre produtos, qualidade de atendimento e cancelamento), com foco na queda do volume de chamadas atendidas por humanos mensalmente.

O projeto tinha uma meta agressiva de queda de ligações com atendimento humano em um volume de 50% no total e um percentual importante da remuneração variável vinculada a esse indicador. Havia também uma expectativa que com as ações capitaneadas por cada *war room* colaborassem para a queda de reclamações na ANATEL, que começava a ser medida e separada de acordo com os temas de cada *war room*.



Nesse estudo de caso, vamos entender os métodos utilizados e resultados alcançados da *War Room* de Contas, responsável por reduzir as chamadas do SAC por motivo financeiro.

4.2 Identificação e classificação dos problemas

Apesar de haver no processo padrão a necessidade de identificação das chamadas por parte do operador de atendimento, foi percebido, durante a investigação, que os motivos registrados, em sua maioria, não refletiam as reais razões do atendimento ou estavam classificadas com motivos genéricos, o que impossibilitava uma investigação por meio de uma base retirada do sistema de registro de atendimento.

Com conhecimento dessa situação, a *War Room* de Contas, por meio de analistas cedidos pelo atendimento, ouviu uma amostragem aleatória de 5 ligações por dia, dos últimos 90 dias, totalizando 450 auditorias de chamadas.

Nessas audições eram identificados os reais motivos do atendimento, podendo ser mais que um em uma mesma ligação e essas informações eram tabuladas entre: regras de negócios, regras regulamentares, erros de processos, falhas sistêmicas, atualização de cadastro ou informações gerais (por exemplo: informações sobre data de vencimento, valor de fatura etc). Essa análise gerou o resultado demonstrado na figura 4:

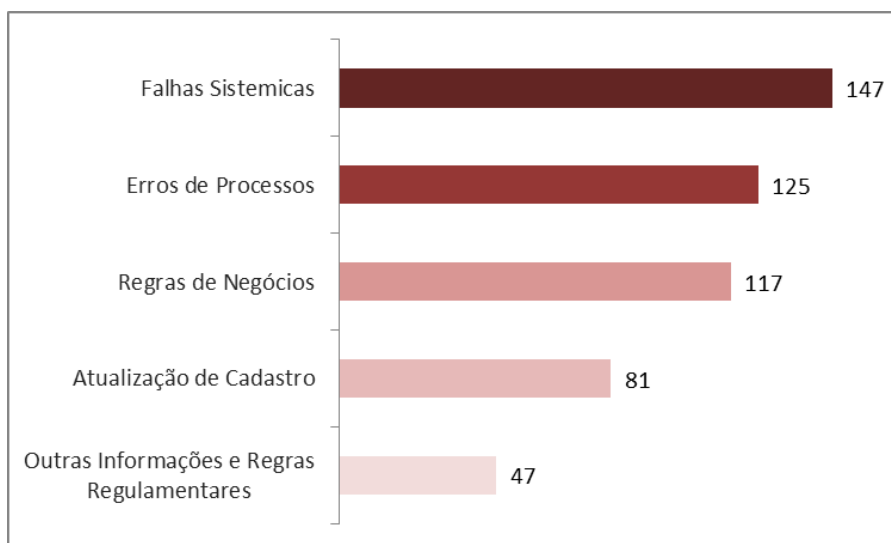


Figura 4 – Motivos de Chamadas
Fonte: autora

4.3 Planejamento da Execução do Projeto

Com essas informações decompostas em informações menores e mais detalhadas (por exemplo: falha na entrega da fatura e vencimento inexistente), deu-se início o mapeamento de premissas e requisitos para solucionar cada problema por meio de sessões de brainstorming de 5 minutos para cada problema, a partir dos problemas com maior impacto. Problemas tabulados como regras regulamentares e outras informações foram desconsiderados.



Após a geração de ideias, cada proposta foi ordenada de acordo com o seu valor no projeto, isto é, de acordo com o potencial de ganho na queda de chamada (com base no gráfico de Pareto de motivos detalhados), reescritas em *post-its* coloridos conforme figura 5 e colados em uma matriz (figura 6), que determinava a ordem de implantação ou desenvolvimento da melhoria, considerando dois fatores: esforço de trabalho e viabilidade de implantação, com uma visão holística da empresa, independente se a necessidade de solução era sistêmica, processual ou disruptiva (com necessidade de inovação ou criação de ferramentas e/ou produtos e serviços).



Figura 5 – Cores post-it para impacto de chamadas
Fonte: autora

Priorização de Implantação

		6	8	9
Viabilidade	Menos viável >	3	4	7
	< Mais viável	1	2	5
		<Baixo Esforço	Esforço de Trabalho	Alto Esforço >

Figura 6 – Matriz de Priorização de Implantação
Fonte: autora

4.4 Execução e Monitoramento do Projeto

Durante a execução do projeto, o grupo se reunia presencialmente durante duas tardes por semana.

Durante a primeira tarde, às quartas-feiras, o objetivo era entender com o time de trabalho se existia impedimentos que pudessem impactar nos prazos definidos anteriormente. Além disso, a tarde era usada também para acompanhar os principais indicadores do projeto, “Chamadas por motivos financeiros” e “Rechamada em 30 dias” e os específicos de cada ação implantada. Isso era feito para determinar se havia necessidade de algum ajuste adicional no processo, sistema ou forma de trabalho.



Na segunda tarde, às sextas-feiras, às 4 horas eram utilizadas para planejar as atividades da próxima semana, mapeando partes interessadas, necessidades sistêmicas e/ou desenvolvimento de novos processos e prazo de implantação das mesmas, gerando metas de datas e resultados.

Nessa segunda tarde, os *post-its* eram “atualizados”, quando do início de uma nova atividade, os *post-it* eram retirados da matriz de tempo e esforço explicada acima, e colocados em outro painel, utilizando o *Kanban* com quatro colunas (Desenvolvendo, Piloto/Em teste, Implantar, Feito), que também sofriam atualizações em relação às suas colunas conforme o avanço da atividade.

Além dessas tardes, nos demais dias da semana, o time de trabalho se reunia presencialmente ou por meio de conferência telefônica por um período de 10 a 15 minutos para atualização do time e solicitação de pedido de ajuda em casos em que era identificado risco para a não entrega do que foi planejado na última semana.

4.5 Resultados do Projeto

Com as implantações propostas pela *War Room* de Conta, a meta de redução de volume de ligação foi atingida em sua totalidade. Foi utilizado como parâmetro de medição a média de chamada entre clientes Tier 1, Tier 2 e Tier 3 e a queda de janeiro/2016 até dezembro/2016 foi de 4,5 pontos percentuais.

Para uma ordem de grandeza, em janeiro, os motivos financeiros representavam 10,8% do total de chamadas recebidas pela empresa Y em um mês. Esse percentual representava 1,3 milhão de chamadas por mês. Em dezembro, no fim do projeto, o percentual de ligações pelo mesmo motivo representava 6,4% das ligações ao SAC recebidas por motivo financeiro, que representava aproximadamente 400 mil chamadas. Para completar o entendimento, em fevereiro de 2017, já no acompanhamento de qualidade do projeto, o percentual médio de chamadas entre as categorias de clientes foi de 5,2% do total de ligações do mês, o menor índice medido no projeto, representando um volume de 280 mil ligações. Toda a evolução pode ser acompanhada na figura 7.

Como comentado no início do caso, também havia uma expectativa de que queda das reclamações de ANATEL por razões financeiras, isso realmente ocorreu. Em janeiro de 2016, 0,44% da base de clientes da empresa Y reclamaram na ANATEL. Em dezembro do mesmo ano, esse índice passou a ser 0,25%, uma queda de 43,2%.

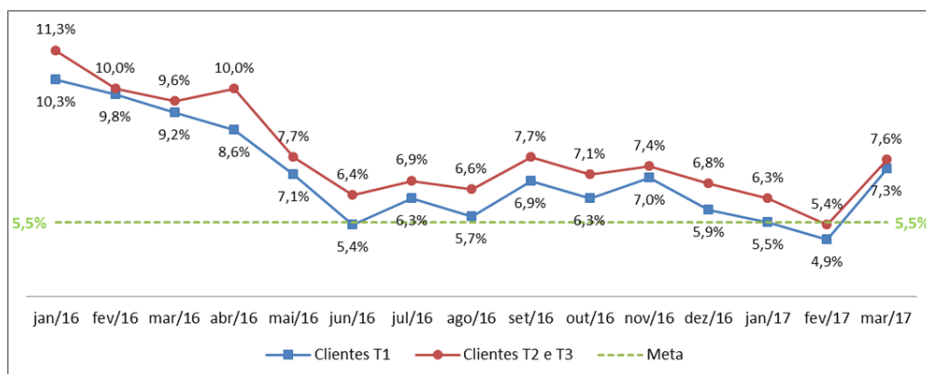


Figura 7 – Evolução do percentual de chamadas por motivos financeiros. Comparação realizada versus ao volume total de ligações em cada mês.



Fonte: autora

5 Conclusões

O uso de eventos do *Scrum* durante o projeto se mostrou eficiente. Apesar de, no caso apresentado, os eventos do *Scrum* não estarem declarados com seus nomes oficiais, nota-se que eles estão presentes nas rotinas de trabalho apresentadas.

Utilizar *Sprints* para entrega de valores do projeto (melhoria de processos e correção de falhas sistêmicas) se mostrou eficiente para acelerar e, principalmente, manter a queda nas ligações devido à rápida possibilidade de respostas. As *Sprints*, apesar de não estarem claramente identificadas são reconhecidas pela rotina de reuniões de retrospectivas seguidas por reuniões de planejamento.

Outro fator essencial ao sucesso do projeto foram as reuniões diárias (*Daily Scrum*), que permitiam que o risco fosse minimizado ou não ocorresse devido a ações rápidas de correção ou intervenção.

Esses pontos criaram uma rotina entre os integrantes do time, o que facilitou o entrosamento entre pessoas de áreas funcionais diferentes dentro de um ambiente de confiança, que perdurou durante todo o projeto e também no acompanhamento dos resultados após a sua finalização.

A rotina consistente reduziu a necessidade de intervenção do gerente de projetos junto aos gerentes funcionais em diversos aspectos, desde a disponibilização dos recursos até o apoio para que as entregas fossem realizadas no prazo planejado.

Importante ressaltar também a necessidade da transparência que o projeto foi integralmente conduzido em todos os momentos, com gestão visual e utilização de ferramentas e dinâmicas ágeis para definição, priorização e replanejamento de entregas, que facilitou com que pessoas de diferentes equipes funcionais compreendessem as interfaces e impactos de todas as entregas e se sentissem como uma equipe única na busca de resultados.

A popularização dessa maneira de condução projetos, cria oportunidades para que projetos hoje geridos exclusivamente por meio de metodologias tradicionais de construção de escopo, cronograma e execução, comecem a adotar essa técnica ágil, se não em sua forma integral, de forma híbrida. Corroborando com essa visão, o PMBOK, guia mais tradicional para gestão de projetos, já traz em sua nova versão (6ª edição) considerações sobre ambientes ágeis e gestão de projetos híbridos, que trabalham simultaneamente com meios mais tradicionais de documentação e meios ágeis de gestão.

Referências Bibliográficas

ÁGILE: Practice Guide. Pensilvânia: Project Management Institute, 2017. 182 p.

ANDRADE, Maria Margarida de. **Introdução a Metodologia do Trabalho Científico: Elaboração de trabalhos na graduação.** 10. ed. São Paulo: Editora Atlas SA, 2010. 158 p. Disponível em: <<https://integrada.minhabiblioteca.com.br/#/books/9788522478392/>>. Acesso em: 23 abr. 2018.

BROD, César. **Scrum Guia Prático para Projetos Ágeis.** 2ª. ed. São Paulo: Novatec, 2015.



200 p.

COCKBURN, Alistair. Alistair Cockburn. Disponível em: <<http://alistair.cockburn.us/>>. Acesso em: 07 maio 2018.

COHN, Mike. **Desenvolvimento de Software com Scrum: Aplicando Métodos Ágeis com Sucesso**. 1ª. ed. São Paulo: Bookman Companhia Editora Ltda, 2011. 200 p. v. 496.

GUIA de Apresentação de Teses: Tabelas, Quadros e Figuras. Disponível em: <http://www.biblioteca.fsp.usp.br/~biblioteca/guia/i_cap_04.htm>. Acesso em: 09 maio 2018.

GUIA PMBOK. 6ª. ed. Pensilvânia: Project Management Institute, 2018. 793 p.

KELLER, Robert T. Cross-Functional Project Groups in Research and New Product Development: Diversity, Communications, Job Stress and Outcomes. **The Academy of Management Journal**, [S.l.], p. 547-555, jun. 2001. Disponível em: <<https://journals.aom.org/doi/full/10.5465/3069369>>. Acesso em: 10 maio 2018.

MANIFESTO for Agile Software Development. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 23 abr. 2018.

MOUNTAIN Goat Software. Disponível em: <<https://www.mountaingoatsoftware.com>>. Acesso em: 06 maio 2018.

PRODANOV, Cleber Cristiano ; FREITAS, Ernani César de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. 2. ed. Novo Hamburgo: Universidade Feevale, 2013. 276 p.

SCHWABER, Ken; SUTHERLAND, Jeff. Guia do Scrum MR. [S.l.: s.n.], 2017. 20 p. Disponível em: <<http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 28 abr. 2018.

SCRUM Alliance. Disponível em: <<https://www.scrumalliance.org/>>. Acesso em: 05 maio 2018.

SCRUM.ORG. Disponível em: <<https://www.scrum.org/>>. Acesso em: 23 abr. 2018.

SUTHERLAND, Jeff. **Scrum: A arte de fazer o dobro do trabalho na metade do tempo**. São Paulo: LEYA EDITORA LTDA., 2014. 261 p.

YIN, Ribert K. **Estudo de Caso: Planejamento e Métodos**. 2ª. ed. São Paulo: Bookman Companhia Editora Ltda., 2001. 205 p.