

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GABRIEL FERNANDES

IMPLANTAÇÃO DE METODOLOGIA ÁGIL
NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:
Um Estudo de Caso em uma Empresa de Varejo Supermercadista

Florianópolis/SC

2019

GABRIEL FERNANDES

**IMPLANTAÇÃO DE METODOLOGIA ÁGIL
NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:
Um Estudo de Caso em uma Empresa de Varejo Supermercadista**

Artigo apresentado como requisito parcial
para obtenção do título de Bacharel em
Sistemas de Informação, pelo Curso de
Sistemas de Informação da Universidade
do Vale do Rio dos Sinos - UNISINOS

Orientadora: Profa. Dra. Margrit Reni Krug

Florianópolis/SC

2019

**IMPLANTAÇÃO DE METODOLOGIA ÁGIL
NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE:
Um Estudo de Caso em uma Empresa de Varejo Supermercadista**

Gabriel Fernandes
Profa. Dra. Margrit Reni Krug

Resumo: Novos modelos de negócio e concorrentes no mercado supermercadista, estão impulsionando os supermercados tradicionais a buscarem formas de aumentar a produtividade. Observando-se a necessidade de mudança, propôs-se neste trabalho o mapeamento e aplicação de práticas ágeis na equipe de desenvolvimento de software em uma empresa supermercadista. Dado o mapeamento do processo anterior, treinamento e capacitação da equipe em métodos ágeis, um novo processo de desenvolvimento foi desenhado e implantado na empresa. Afim de avaliar as melhorias propostas aplicou-se um questionário aos colaboradores antes e depois da implantação do novo processo e realizou-se comparações de indicadores. Após análises dos dados, foi possível verificar, que os resultados obtidos mostraram-se satisfatórios, assim atendendo aos objetivos do trabalho e às necessidades da empresa, o qual era o de melhorar o processo de desenvolvimento de software da empresa estudada aplicando práticas da metodologia ágil.

Palavras-chave: Scrum. Método Ágil. Processo de Desenvolvimento de Software.

1 INTRODUÇÃO

O volume de vendas nos supermercados apresentou queda por dois anos seguidos, segundo Mattos (2017) em reportagem à revista Valor, a queda no volume de vendas foi de 1,2% em 2015 (MATTOS, 2016), seguido por 4,7% em 2016 (MATTOS, 2017), isto sem considerar o atacarejo, modelo relativamente novo, que vende para empresas e pessoas físicas, é uma modalidade de distribuição onde o atacado vende diretamente para o consumidor final, substituído o papel desempenhado pelo varejista (CALSAVARA, 2009). Em 2017, apesar da previsão de crescimento real segundo a Associação Brasileira de Supermercados (ABRAS) ter sido de 1,3% (MATTOS, 2017), na prática, já descontado a inflação, ficou em 1,25% (GLOBO, 2018).

A partir da situação relatada, as empresas do ramo de supermercados necessitam adequar-se ao contexto, e uma alternativa pode ser a melhoria de seus processos, com foco no aumento da produtividade das equipes, sem aumentar o quadro de colaboradores, e permitir a criação de espaço para a melhoria contínua, o que pode contribuir na redução de custos. Assim formulou-se a seguinte pergunta que norteou o estudo: **A aplicação de métodos ágeis no processo de desenvolvimento de software pode melhorar a produtividade da equipe de desenvolvimento, sem necessitar adicionar colaboradores?**

A empresa do estudo de caso necessitava de mudanças que revertessem ou minimizassem os impactos financeiros do cenário apresentado anteriormente. O Hippo é uma empresa familiar do ramo supermercadista com lojas conceito em gastronomia na Grande Florianópolis, que optou por desenvolver seu próprio sistema de gestão, e atualmente conta com quatro lojas, tendo o início das atividades em 1997.

Diante da situação descrita o objetivo geral deste trabalho representou a implantação da metodologia ágil no processo de desenvolvimento de software da empresa Hippo. Para alcançar a proposta de objetivo geral destacam-se os seguintes objetivos específicos:

- a) Modelar e avaliar o atual processo de desenvolvimento da empresa Hippo;
- b) Propor melhorias no processo de desenvolvimento de software da empresa baseando-se em boas práticas de métodos apresentados na literatura;
- c) Implantar o processo proposto e aplicá-lo em um projeto piloto;
- d) Avaliar, através da coleta de opiniões dos colaboradores, as percepções sobre o processo proposto.

2 FUNDAMENTAÇÃO TEÓRICA

No desenvolvimento de software conforme Pressman e Maxim (2016, p. 16) “Processo é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato” e segundo Sommerville (2011, p. 16), “O processo de software inclui todas as atividades envolvidas no desenvolvimento do software. Atividades de alto nível de especificação, desenvolvimento, validação e evolução são parte de todos os processos de software”.

Todo software, para ser construído, segue algum tipo de processo de desenvolvimento de software, mesmo aqueles projetos que não possuem um

processo claro e documentado, ou seja, os projetos que normalmente são construídos com base no modelo Codificar e Consertar. Wazlawick (2013, p. 21), o apresenta como, “[...] o antimodelo por excelência, que é Codificar e Consertar, que consiste na absoluta falta de processo.”. É o mais básico modelo de processo de desenvolvimento de software.

Segundo Pressman e Maxim (2016) identifica-se uma metodologia básica nos modelos de processos de desenvolvimento de software, com atividades genéricas que possam ser aplicadas em qualquer projeto de software. Essas atividades genéricas comum para as metodologias de processo de criação de software, conforme Pressman e Maxim (2016) são: comunicação; planejamento; modelagem; construção e; entrega.

2.1 Desenvolvimento Ágil de Software

De acordo com o Guia SBOK (2017), o acelerado crescimento das exigências do mercado e o avanço tecnológico dificultaram o uso de modelos tradicionais de gerenciamento de projetos, criando a oportunidade para o surgimento de modelos mais ágeis. Os modelos ágeis se apresentam como alternativa para a solução de falhas em processos dos modelos tradicionais de gerenciamento de projetos.

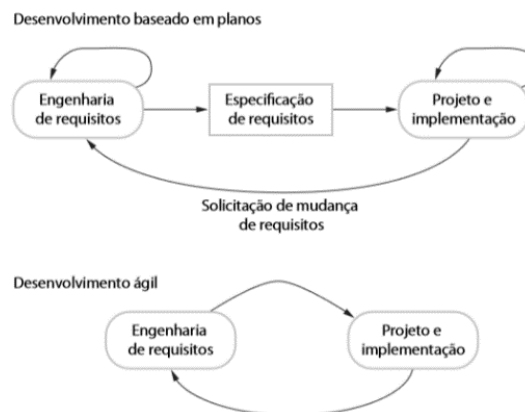
Conforme Pressman e Maxim (2016, p. 66), “Em 2001, Kent Beck e outros 16 renomados desenvolvedores, autores e consultores da área de software [...] assinaram o “Manifesto para o Desenvolvimento Ágil de Software”.”, esse encontro fez do ano de 2001 o marco inicial do desenvolvimento ágil. Embora, segundo o Guia SBOK (2017), os modelos com foco em valores ágeis já existirem desde a década de 1950, apenas aqueles que estão em conformidade com o “Manifesto para o Desenvolvimento Ágil de Software”, criado em 2001, são normalmente considerados modelos ágeis. O “Manifesto para Desenvolvimento Ágil de Software” atualmente é um documento público, originalmente escrito na língua inglesa, disponível na internet no endereço <http://agilemanifesto.org>.

Os métodos de desenvolvimento ágil de software são reconhecidos por darem mais valor para os fatores humanos, sem pormenorizar os fatores técnicos e organizacionais, Wazlawick (2013, p.45) afirma que “os modelos ágeis de desenvolvimento de software seguem uma filosofia diferente da filosofia dos modelos prescritivos”. Contudo, os métodos de desenvolvimento ágil não desprezam

totalmente os modelos prescritivos; e, trazem uma nova forma de priorizar as atividades, conforme Pressman e Maxim (2016, p. 66), “[..] engenharia de software ágil combina filosofia com um conjunto de princípios de desenvolvimento.[...] Os princípios de desenvolvimento priorizam a entrega mais do que a análise e o projeto (embora essas atividades não sejam desencorajadas)[...]”.

Na Figura 1, pode-se ver um comparativo simplificado do processo de especificação no desenvolvimento prescritivo baseado em planos e no desenvolvimento ágil. Nela pode-se observar que em relação ao desenvolvimento ágil há de pelo menos uma etapa a mais no desenvolvimento baseado em planos, a “Especificação de requisitos” e segundo Wazlawick (2013), especificar requisitos completos antes de iniciar a produção de código é uma tarefa bastante difícil no processo de desenvolvimento de software.

Figura 1 – Especificação de Desenvolvimento Baseado em Planos e Ágil



Fonte: Sommerville (2011, p. 43).

Ainda na Figura 1, pode-se verificar que métodos ágeis se concentram mais na entrega, passando da etapa de “Engenharia de requisitos” para a de “Projeto e implementação”, desta forma conforme Sommerville (2011), os métodos de desenvolvimento ágil são métodos que utilizam o modelo de desenvolvimento incremental que focam no desenvolvimento rápido, versões frequentes, redução de custos não previstos e a produção de código de melhor qualidade.

2.2 Metodologia Ágil Scrum

Conforme Wazlawick (2013), Scrum é um método ágil de gestão de projetos de software, que tem como um dos principais conceitos o *Sprint*, no qual consiste em um

período delimitado, ciclos de desenvolvimento, podendo variar de duas e quatro semanas. Ainda segundo Wazlawick (2013, p.56), “a concepção inicial do Scrum deu-se na indústria automobilística [...] e o modelo pode ser adaptado a várias outras áreas diferentes da produção de software.” e Sutherland (2014) afirma que o Scrum acelera todos os empreendimentos humanos, pois independente do tipo de projeto ou problema ele poderá ser utilizado para aprimorar desempenho e resultados.

De acordo com o Guia SBOK (2017) o Scrum possui seis princípios que servem como diretrizes fundamentais para a aplicação da metodologia Scrum e devem obrigatoriamente serem usados em todos os projetos Scrum. Os seis princípios do Scrum apresentados no Guia SBOK (2017, p.10), seguem destacados a seguir.

O primeiro é o **Controle de Processos Empíricos**, segundo o guia esse princípio enfatiza a filosofia central do Scrum com base em três ideias principais: transparência, inspeção e adaptação. Em segundo vem **Auto-organização**, esse princípio está focado nos colaboradores atuais de uma organização, que entregam significativamente um maior valor quando são auto organizados e isto resulta em times mais satisfeitos e responsabilidade compartilhada; e em um ambiente inovador e criativo que é mais propício ao crescimento. (GUIA SBOK, 2017)

Depois a **Colaboração**, concentra-se nas três dimensões básicas relacionadas com o trabalho colaborativo: consciência, articulação e apropriação. Também defende o gerenciamento de projetos como um processo de criação de valor compartilhado, com times trabalhando e interagindo em conjunto para atingirem melhores resultados. (GUIA SBOK, 2017)

A **Priorização Baseada em Valor**, princípio que destaca o foco do Scrum em entregar o máximo de valor de negócio possível, durante todo o projeto. Seguido do **Time-boxing**, princípio que enfatiza que o tempo é uma restrição limitada em Scrum, e como ele é usado para ajudar a gerenciar o planejamento e execução do projeto com eficácia. Os elementos de Time-boxed em Scrum incluem os Sprints, as Reuniões Diárias, a Reunião de Planejamento do Sprint, e a Reunião de Revisão do Sprint. (GUIA SBOK, 2017)

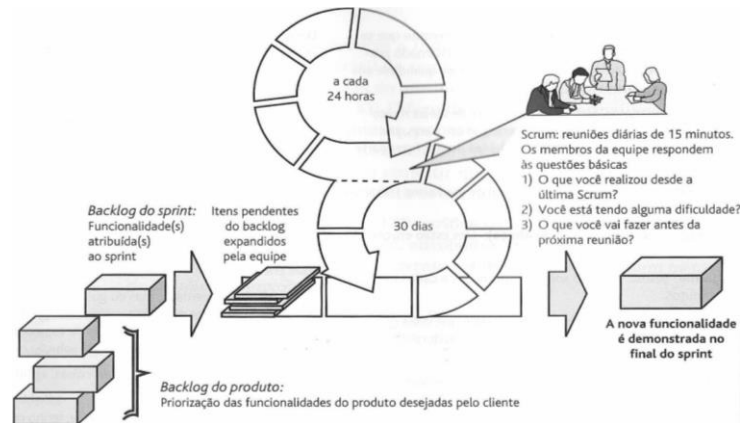
Por último o princípio do **Desenvolvimento Iterativo**, que define o desenvolvimento iterativo e enfatiza como administrar melhor as mudanças e criar produtos que atendam às necessidades do cliente. Também delinea as responsabilidades do Dono do Produto e da organização, com relação ao desenvolvimento iterativo. (GUIA SBOK, 2017)

No princípio de Controle de Processos Empíricos, destacados anteriormente, baseiam-se a fundamentação do Scrum, segundo Schwaber e Sutherland (2017), o controle de processos empíricos ou, simplesmente, empirismo indica que o conhecimento é adquirido através das experiências vivenciadas e as decisões tomadas diante do que é conhecido. Schwaber e Sutherland (2017) afirmam que há três pilares que apoiam a implementação do empirismo, conforme destacados (SCHWABER e SUTHERLAND, 2017):

- **Transparência:** Aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. A transparência requer que estes aspectos tenham uma definição padrão comum para que os observadores compartilhem um mesmo entendimento comum do que está sendo visto.
- **Inspecção:** Os usuários Scrum devem, frequentemente, inspecionar os artefatos Scrum e o progresso em direção ao objetivo da Sprint para detectar variações indesejadas. Esta inspecção não deve ser tão frequente que atrapalhe o objetivo dos trabalhos. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar.
- **Adaptação:** Se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o resultado do produto será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios.

Segundo Pressman e Maxim (2016, p.78), “Os princípios do Scrum são coerentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento [...] Em cada atividade metodológica, ocorrem tarefas realizadas dentro de um [...] Sprint.” e também que todo trabalho realizado em um Sprint pode ser adaptado ao problema em questão e modificado a qualquer momento pela própria equipe Scrum (PRESSMAN e MAXIM, 2016). Um diagrama do processo Scrum pode ser visualizado na Figura 2.

Figura 2 – Processo Scrum



Fonte: Pressman e Maxim (2016, p. 78).

O *Sprint*, segundo Wazlawick (2013, p.57), “é o ciclo de desenvolvimento de poucas semanas de duração sobre o qual se estrutura o Scrum” e de acordo com Pressman e Maxim (2016, p.79) os Sprints são “[...] unidades de trabalho solicitadas para atingir um requisito estabelecido no registro de trabalho(backlog) e que precisa ser ajustado dentro de um prazo já fechado [...]”, ainda segundo Pressman e Maxim (2016) os Sprints permitem o trabalho em um ambiente de curto prazo, porém estável.

O *product backlog (backlog)*, é um dos artefatos do Scrum que segundo Pressman e Maxim (2016, p.78), trata-se de “uma lista com prioridades dos requisitos ou funcionalidades do projeto que fornecem valor comercial ao cliente. Os itens podem ser adicionados a esses registros a qualquer momento (é assim que as alterações são introduzidas)”. Wazlawick (2013, p.56) destaca, “Um dos princípios do manifesto ágil é usado aqui: adaptação em vez de planejamento. Então o *product backlog* não precisa ser completo no início do projeto. Pode-se iniciar apenas com as funcionalidades mais evidentes, [...] para depois, [...] tratar novas funcionalidades [...]”.

Conforme Wazlawick (2013) existem no Scrum três perfis importantes de trabalho, que na metodologia são chamados de papéis, são eles:

- **Scrum Master (Mestre Scrum):** “[...] não é um gerente no sentido dos modelos prescritivos. Não é um líder, já que as equipes são auto organizadas, mas um facilitador [...] e um solucionador de conflitos. (WAZLAWICK, 2013, p.56).
- **Product Owner (Dono do Produto):** de acordo com Wazlawick (2013), é a pessoa responsável pelo projeto, responsável por indicar os requisitos dos projetos que devem ser trabalhados em cada *Sprint*, pelo retorno do

investimento e ser o principal, mas não exclusivo, canal com o cliente, deve ser conhecedor das necessidades deles.

- **Scrum Team (Time/Equipe Scrum):** Conforme Wazlawick (2013, p.56), “[...] é a equipe de desenvolvimento. Essa equipe não é necessariamente dividida em papéis como analista, designer e programador, mas todos interagem para desenvolver o produto [...]”. Para Schwaber e Sutherland (2017), o *Scrum Team* é o time completo, com um *Product Owner*, um *Scrum Master* e o time de desenvolvedores.

Conforme Schwaber e Sutherland (2017), “O Scrum prescreve quatro Eventos formais para inspeção e adaptação [...]” sendo eles o **Planejamento da Sprint**, a **Reunião diária**, a **Revisão da Sprint** e por fim a **Retrospectiva da Sprint**. De acordo com Wazlawick (2013), para abrir uma nova *Sprint* é feito a reunião de Planejamento da *Sprint* (*sprint planning meeting*), onde itens do *product backlog* são priorizados para serem implementados na *Sprint*, levando esses itens do *product backlog* para a *sprint backlog*, assim tem-se a lista de funcionalidades a serem implementadas num ciclo (*Sprint*) que se inicia.

A Reunião Diária (*daily Scrum*), que conforme Pressman e Maxim (2016, p.79), “[...] são reuniões curtas (tipicamente 15 minutos), realizadas diariamente pela equipe Scrum. São feitas três perguntas-chave que são respondidas por todos os membros da equipe”, destacadas por Sutherland (2014, p.80), a seguir:

- O que você fez ontem para ajudar a equipe a concluir o Sprint?
- O que você vai fazer hoje para ajudar a equipe a concluir o Sprint?
- Que obstáculos a equipe está enfrentando?

As reuniões diárias são conduzidas pelo Mestre Scrum, responsável por avaliar as respostas de cada membro da equipe e entender se alguma ação deve ser tomada, elas são realizadas diariamente com o objetivo de identificar problemas de forma precoce, bem como para compartilhar conhecimento e atualizar a posição atual do *Sprint*. (PRESSMAN e MAXIM, 2016).

Continuando o destaque para as cerimônias importantes no Scrum, temos por fim a Revisão e Retrospectiva do *Sprint*, que são respectivamente, segundo Schwaber e Sutherland (2017), a revisão do trabalho realizado para inspecionar as entregas e adaptar o *product backlog* se necessário e a retrospectiva é a reunião para a equipe inspecionar a si próprio, com foco na forma como o trabalho foi realizado e não nas

entregas ou incrementos, a retrospectiva visa melhorias na forma de executar o Scrum, dadas as particularidades de cada projeto e empresa.

2.3 Trabalhos Relacionados

O trabalho de Silva (2011), que relatou sobre o impacto da adoção de metodologias de desenvolvimento de sistemas em empresas de pequeno porte. Neste trabalho o autor destacou a resistência às mudanças, especialmente por parte dos funcionários antigos e equipes pequenas, por nunca terem utilizados qualquer tipo de método. Contudo Silva (2011) obteve resultados positivos com a implantação de uma adaptação do método XP, que mesmo com a resistência por parte dos colaboradores, as melhorias foram alcançadas e percebidas, como consequência melhorou-se o ambiente de trabalho da área de desenvolvimento e aumentou a qualidade de entregas, além de resultar em prazos de entrega melhor estimados.

O trabalho de Ferreira (2007), apresentou uma pesquisa focada na implantação de um novo processo de software baseado no Scrum em uma empresa de software para gestão de empresas. Nessa pesquisa, Ferreira (2007), destacou que a percepção de melhorias para os envolvidos diretamente com o Scrum é mais clara, quando comparadas com a percepção dos demais colaboradores da empresa, o que pode estar diretamente relacionada ao desconhecimento dos desafios e dificuldades das atividades para a produção de software. Ferreira (2007) ressaltou, também a melhora no compartilhamento das informações entre as equipes, que além de melhorar a comunicação, permitiu maior conhecimento sobre os projetos em desenvolvimento para todas as equipes.

Souza (2014), em seu trabalho sobre a implantação da metodologia ágil Scrum em um ambiente de desenvolvimento de uma empresa de software de pequeno porte, com apenas quatro desenvolvedores, relatou que a mudança de papéis com a implantação do Scrum pode ser bastante traumática, seja com o surgimento de novos papéis ou com a extinção deles. Souza (2014) descreveu o impacto da inclusão dos novos papéis, sendo menor para o Dono do produto, por ter sido atribuído ao atual Analista de Sistemas da equipe e, assim tendo pouco impacto, pois ele já era a pessoa com maior conhecimento do produto e suas funcionalidades, que é uma das principais atividades do Dono do Produto. O papel de maior impacto foi o de Scrum Master, atribuído ao atual gerente de projetos, alterando sua rotina atual de trabalho, pois no

Scrum, o papel de Scrum Master não configura em um gerente, mas sim em facilitador, aqui ele precisa ser mais flexível e confiante na equipe para deixar o trabalho fluir.

Os trabalhos de Silva (2011), Ferreira (2007) e Souza (2014) se assemelham ao trabalho realizado no Hippo por terem cenário equivalentes. Todos foram aplicados em empresas de pequeno porte e envolveram colaboradores antigos. O trabalho de Souza (2014), em especial, configura um cenário similar ao estudado neste trabalho, por se tratar de uma equipe com apenas quatro desenvolvedores, um a mais que o time do Hippo. Em contrapartida, uma característica no projeto do Hippo, que pode configurar uma diferença nos resultados e no fluxo da implantação do processo, refere-se ao fato de se tratar de equipe de desenvolvimento exclusivo para clientes internos, ou seja, para uso próprio em um ambiente de varejo supermercadista, que possui dinâmica de trabalho bastante diferente das empresas de tecnologia.

3 MATERIAIS E MÉTODOS

A metodologia para a realização do trabalho baseou-se na pesquisa-ação, que segundo Thiollent (1986) trata-se de uma “linha de pesquisa associada a diversas formas de ação coletiva que é orientada em função da resolução de problemas ou de objetivos de transformação”, utilizando-se de aspectos quantitativos e qualitativos (AZEVEDO, 2011 apud ROESCH, 2009), para a ação de implantar a metodologia ágil em equipe de desenvolvimento de software.

As etapas para a aplicação da metodologia tiveram como referência o “Método do Arco”, de Charles Maguerez, baseado no estudo apresentado por Bordenave e Pereira (1991).

O esquema de trabalho proposto por Maguerez (BORDENAVE e PEREIRA, 1991), tem como ponto inicial a observação da realidade, com foco no que procura identificar-se, após verificado os aspectos da realidade, problemas são selecionados para serem estudados. A próxima etapa é a identificação dos pontos-chave, momento em que se define o que será pesquisado a em cada problema. A terceira etapa é a teorização, na qual é realizada uma investigação dos pontos-chave. Após a teorização, inicia-se a elaboração de hipóteses para solução. Por fim, a última etapa é da aplicação a realidade, onde soluções são aplicadas para a transformação da realidade, mesmo que seja em uma pequena porção dela.

Para a realização desta metodologia utilizou-se cinco etapas desenvolvidas a partir da realidade ou de um recorte dela, foram elas:

- Observação da Realidade: Diagnostico do processo atual de desenvolvimento de software;
- Pontos-Chave: Identificação e tabulação dos problemas existentes;
- Teorização: Análise dos problemas tabulados, conforme fundamentação teórica;
- Hipóteses de Solução: Sugestão de melhorias para o modelo de desenvolvimento de software;
- Aplicação à Realidade: Aplicação das sugestões no processo de desenvolvimento e mudança na realidade da equipe de desenvolvimento de software.

Para analisar o cenário atual, em pelo menos três aspectos, realizou-se a observação da realidade do processo de desenvolvimento utilizado, levantamentos de dados nos sistemas desenvolvidos, e elaboração e aplicação de um questionário para avaliar o processo junto aos colaboradores envolvidos. Após o acompanhamento e observação da realidade, foi realizada a análise quantitativa gerada a partir do atual sistema de controle de requisições.

Para finalizar a análise do cenário atual, foi aplicado um questionário na equipe de desenvolvedores, realizado com auxílio do departamento de recursos humanos do Hippo. O questionário foi respondido integralmente pelos três desenvolvedores de forma anônima, recebeu o nome de “Pesquisa de Clima Interno” e pode-se avaliar seu conteúdo na íntegra através do APÊNDICE A.

A partir desta análise foi realizado o mapeamento do processo atual, permitindo avaliar e identificar pontos de melhorias com base na fundamentação teórica dos métodos ágeis. Identificados os pontos de melhorias, uma proposta de mudança do processo de desenvolvimento foi apresentada para a equipe de desenvolvimento. Em seguida foi realizado um workshop com eles, contextualizando os métodos ágeis e apresentando uma proposta de construção do novo processo de desenvolvimento de forma cooperada, incluindo os participantes e o pesquisador na missão de construir um novo processo que melhor se adaptasse ao Hippo.

Ao final da construção e implantação do novo processo de desenvolvimento, foram feitas novas análises para avaliar os resultados do trabalho realizado. Novos dados foram coletados que permitiram a análise do resultado em dois aspectos:

quantitativo, a partir da leitura e análise das informações do sistema de controle de requisições, e do sistema de gestão de processo ágil, implantado durante a aplicação das mudanças; e qualitativo, aplicando novamente o questionário para avaliar a percepção dos colaboradores envolvidos que pode-se avaliar seu conteúdo na integra através do APÊNDICE B.

4 RESULTADOS

Esta pesquisa propôs o diagnóstico do processo atual de desenvolvimento de software, a identificação dos pontos de melhorias, a análise desses pontos e contextualização teórica, a sugestão de melhorias nos processos e a aplicação em projeto piloto no Hippo, empresa do ramo supermercadista que desenvolve seu próprio sistema de ERP há 20 anos, com o objetivo de aumentar a produtividade da equipe de desenvolvimento mantendo o quadro atual de colaboradores.

4.1 Cenário Atual

Em relação aos sistemas informatizados, são usados produtos de terceiros para: gestão de pessoas, recrutamento, seleção, segurança e medicina do trabalho, gestão de ponto eletrônico, gestão de ponto de venda, transferência eletrônica de fundos, entre outros sistemas menores para inteligência, relatórios e operações acessórias. O principal sistema do Hippo, o sistema de gestão empresarial, comumente conhecido como ERP (*Enterprise Resource Planning* ou Planejamento dos Recursos da Empresa) é desenvolvido internamente desde 1999. A equipe para desenvolvimento interno possui dois programadores seniores e um programador júnior. A equipe total da área de tecnologia da informação conta atualmente com 6 profissionais, sendo 2 técnicos de suporte, 1 gestor e 3 programadores.

A divisão de desenvolvimento de software no departamento de tecnologia da informação do Hippo, onde são projetados e codificados os aplicativos proprietários não utilizava uma metodologia consciente para desenvolvimento e atualmente tem-se uma coleção de aplicativos sem documentação que em conjunto formam o ERP do Hippo, conhecido internamente como SGE, Sistema de Gestão Empresarial.

Sem desmerecer as iniciativas destes profissionais, é preciso destacar que eles, resolveram alguns problemas, mas criaram uma série de outros, pois o modelo

Codificar e consertar (do inglês, *Build and Fix*), também conhecido como: *Cowboy Coding* ou Modelo Balbúrdia, é um modelo sem método, ou seja, os sistemas são desenvolvidos sem nenhum tipo de projeto ou documentação e na maioria das vezes passam da implementação para a implantação. De acordo com Wazlawick (2013, p. 22), “codificar e consertar: é considerado o modelo *ad-hoc*, ou seja, aquele que acaba sendo usado quando não se utiliza conscientemente nenhum modelo. Sendo assim, ele também não traz nenhuma contribuição, sendo considerado o “marco zero” dos modelos.”

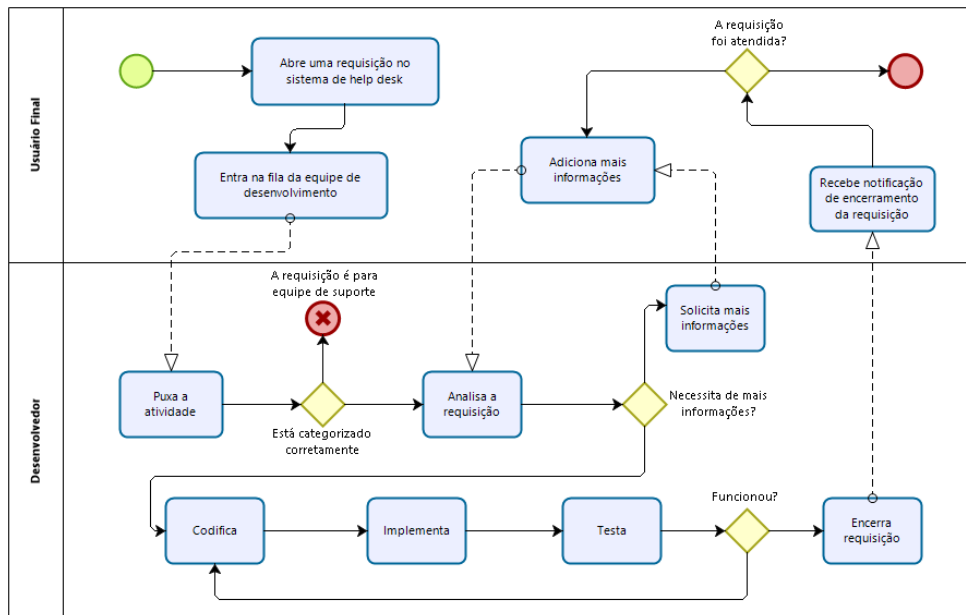
Um dos problemas do modelo Codificar e Consertar é a percepção errada de que tudo é urgente, um ambiente de stress e pressão, insatisfação dos usuários, desgaste físico e psicológico, qualidade duvidosa, difícil manutenção e também a sensação de que falta tempo para evoluir tecnicamente, bem como reescrever os sistemas, promover sustentabilidade no desenvolvimento do ERP e fazer com que esse desenvolvimento seja baseado em processos, no lugar de pessoas.

O processo de desenvolvimento analisado foi direcionado para o principal sistema do Hippo, o sistema de gestão empresarial, denominado internamente como SGE. O SGE é desenvolvido pela equipe de desenvolvimento do Hippo desde 1999, tendo atualmente em seu corpo de colaboradores dois programadores seniores e um programador júnior. Além disso também há equipe de suporte, responsável por fazer o atendimento primeiro nível no SGE e por toda a infraestrutura de informática do Hippo.

4.1.1 Processo de desenvolvimento

O processo de desenvolvimento observado está representado na Figura 3. As requisições de suporte, alteração e manutenção eram realizadas por qualquer usuário e enviadas através de sistema de *tickets* diretamente para os desenvolvedores, sem qualquer tipo de triagem. Os desenvolvedores tentavam atender o número máximo de requisições possíveis, respeitando a fila de entrada, alguns pedidos eram priorizados após muitas negociações com usuários ou por imposição da diretoria ou gestores.

Figura 3 – Processo de desenvolvimento no início do estudo



Fonte: Elaborado pelo autor.

Diante da observação do processo de trabalho existente no início desse estudo, conforme ilustrado na Figura 3, pode-se afirmar que não havia um método de trabalho organizado propositalmente, mas sim do tipo consequente, uma adaptação empírica da equipe com objetivo de atender às demandas existentes. De acordo com Wazlawick (2013, p.22), o processo utilizado era o método Codificar e Consertar, que representa o método sem método, aquele que é utilizado quando não se utiliza um método conscientemente.

4.1.2 Sistema de requisições de suporte, alteração e manutenção

No começo deste estudo as requisições de suporte, alteração e reporte de erros do SGE eram realizadas a partir de sistema de *tickets* chamado Help Desk Software HESK (Hesk), um sistema gratuito e simplificado para gestão de *helpdesk*, trata-se do sistema de entrada para a solução completa de gestão de *helpdesk* da empresa SysAid. As requisições chegavam no Hesk diretamente por e-mail ou através de formulário no sistema Hesk. As análises foram realizadas sobre os dados coletados do período compreendido entre os meses de janeiro até dezembro do ano de 2018.

O sistema de *tickets* concentrava as requisições, mas não garantia a qualidade delas, portanto o primeiro problema identificado é a qualidade das solicitações, pois qualquer usuário do SGE, que possuía um e-mail corporativo, poderia abrir uma

requisição, seja para reportar um problema, para pedir orientação e até mesmo requisitar alteração na funcionalidade do software.

Outro ponto de destaque na análise ao sistema de *tickets* é a concentração dos esforços da equipe na manutenção do sistema SGE, onde praticamente todo o trabalho exercido está em manter o sistema em funcionamento. Evidenciando um excesso de esforço em manutenção, que pode impactar diretamente na geração de valor que a equipe de desenvolvimento entrega para a empresa, pois novos recursos e melhorias acabavam por ficar em segundo plano.

Por fim, na análise quanto às prioridades dos *tickets*, independente da categoria, nota-se também grande esforço em *tickets* classificados como alta prioridade, reflexo da ausência de qualquer método consciente de desenvolvimento, resultando em um número maior de defeitos críticos e conseqüentemente no excesso de tickets nessa classificação.

4.1.3 Questionário de avaliação respondido pelos desenvolvedores da equipe

Analisando as respostas deste questionário constatou-se problemas nas informações para iniciar a execução das tarefas, tendo duas repostas afirmando que raramente elas chegam de forma clara para quem vai desenvolver a tarefa.

Quando perguntado sobre o planejamento das tarefas, talvez influenciado pelo fato de as informações não chegarem de forma clara, também notou-se que existiam oportunidade de melhorias, visto que todas as três respostas para essa questão ficaram entre as opções “às vezes” e “raramente”.

Quanto às orientações da liderança, as respostas mostraram que existia boa comunicação entre a equipe de desenvolvimento e gestor imediato. Relacionado à gestão, identificou-se pelas respostas, a necessidade de melhoria no *feedback* para os membros da equipe. Outro ponto identificado referia-se à percepção de que a autonomia na realização das tarefas era limitada, o que ia na contramão da teoria dos métodos ágeis.

As três perguntas finais do questionário, as quais tinham como objetivo compreender sobre: a carga de trabalho, a colaboração e o relacionamento entre membros da equipe. Obteve-se como resultado a percepção de que carga de trabalho para a maioria da equipe era ruim. Todavia nos dois questionamentos sobre relacionamento e a colaboração, teve-se um resultado positivo, com todas as opções

entre “regular” e “excelente”. Assim, configurou-se como mais um ponto de melhoria, a oportunidade de definir a carga de trabalho, tornando-se um item a ser privilegiado, pois refere-se diretamente à sinergia entre os membros da equipe.

4.2 Implantação do método ágil Scrum

Analisado os resultados obtidos através do levantamento do processo atual de desenvolvimento (relatórios do sistema Hesk e do questionário aplicado aos desenvolvedores), destaca-se alguns pontos chave, que evidenciaram as oportunidades de melhorias, sendo eles:

- a qualidade do conteúdo das requisições feitas por usuários para alterações e correções de erros nos sistemas;
- o maior esforço do trabalho da equipe de desenvolvimento concentra-se na manutenção dos sistemas, restando pouco tempo para novos recursos e melhorias;
- a existência de muitos tickets de manutenção dos sistemas desenvolvidos classificados como alta prioridade;
- a equipe de desenvolvimento raramente recebe de forma clara as informações necessárias para a execução das tarefas;
- falta *feedback* da gestão para os membros da equipe de desenvolvimento;
- a percepção de autonomia da equipe de desenvolvimento na realização das tarefas é limitada, e;
- a equipe de desenvolvimento considera ruim a carga de trabalho.

Por fim, as análises levam à conclusão que de fato era utilizado um processo sem método. Criando um ambiente com muitas oportunidades de melhorias de impacto aos usuários e aos sistemas produzidos, sem esquecer o grande desafio interno, de organização das tarefas e método de desenvolvimento.

4.2.1 Proposta e preparações necessárias

Para a aplicação do processo ágil baseado no Scrum realizou-se um estudo das ferramentas disponíveis no mercado para gestão do *backlog* do produto. Nesse estudo considerou-se, não apenas as funcionalidades dos sistemas, mas também os custos com a licença e implantação. Desta forma a ferramenta selecionada foi o Jira

Software, módulo para rastreamento de projetos e problemas de software com evolução escalável (ATLASSIAN, 2019).

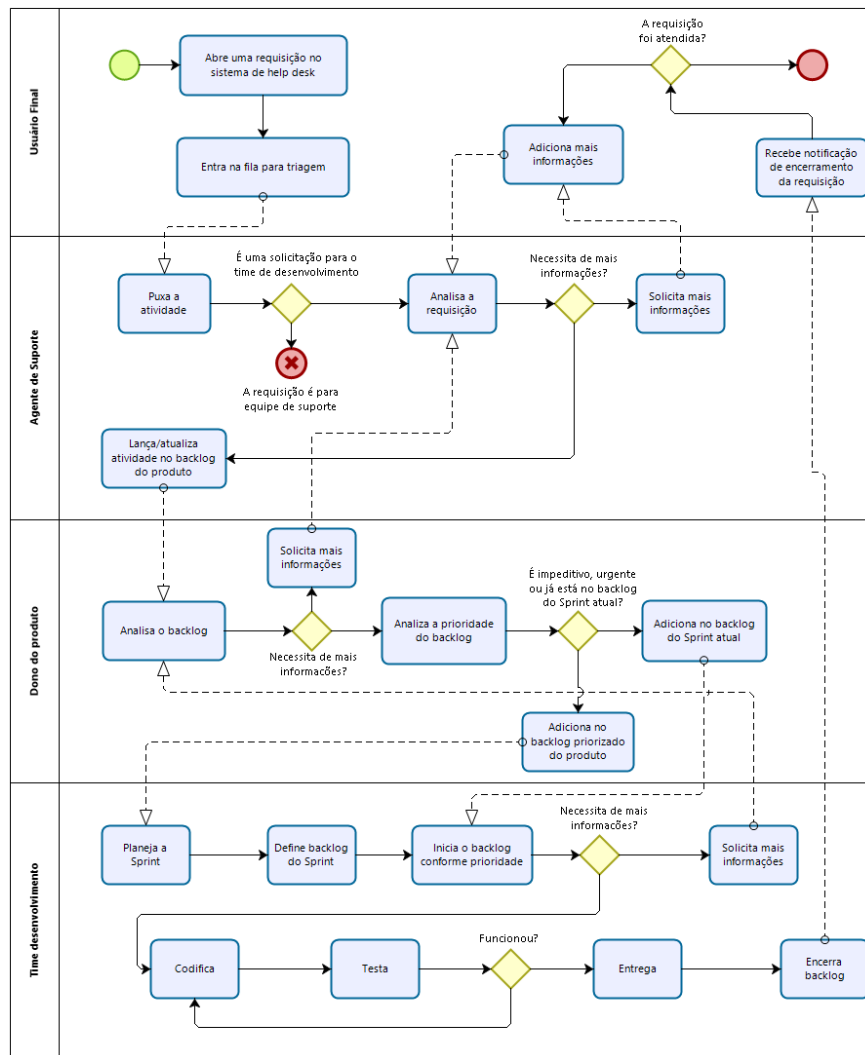
O Jira não é um software simples de configurar, mas é altamente personalizável em nível de usuário, permitindo ampla adequação para qualquer fluxo de trabalho de desenvolvimento de software, seja qual for o método ágil que se deseja implantar. A decisão pela ferramenta, também levou em consideração o seu custo de licenciamento e implantação, no caso do Jira foi adquirido a licença permanente limitada à 10 usuários para instalação em servidor auto gerenciado ao custo de US\$ 1,00 (um dólar) por usuário. Na implantação o Jira também se destaca por ter ampla comunidade de usuários e pela quantidade de documentação disponível.

Com o objetivo de engajar a equipe no processo de adoção do método ágil foi realizada uma apresentação pautada no tema da melhoria contínua, trazendo exemplos de sucesso em várias organizações.

Apresentou-se, também, o conceito de bolha da felicidade conforme Jeff Sutherland descreve no livro “Scrum: a arte de fazer o dobro do trabalho na metade do tempo” (SUTHERLAND, 2014), como introdução para a análise da situação atual, caracterizando o método utilizado pela equipe de desenvolvimento antes da aplicação do método ágil como o “método sem método”, apontando algumas características negativas para as pessoas e projetos feitos sem qualquer método, mas também ressaltando características relevantes da personalidade de indivíduos que possuem força e coragem para encarar esse tipo de trabalho, como responsabilidade e comprometimento em um ambiente de pressão, urgência e stress. A estratégia aqui teve o objetivo de diminuir a resistência da equipe, mostrando que há um jeito melhor para se trabalhar em todos os aspectos, não focado apenas na melhoria da entrega, mas também na felicidade das pessoas.

Ao final da apresentação, uma proposta de método ágil baseado no Scrum, utilizando a ferramenta Jira foi apresentado para a equipe, conceituando alguns aspectos importantes do modelo ágil, como os Sprints, os papéis do Dono do Produto, Mestre Scrum e Time Scrum, os artefatos existentes como as histórias de usuários, o backlog do produto, o backlog dos Sprints e as cerimônias mais importantes como a reunião de backlog do Sprint, as reuniões diárias, as reuniões de revisão e retrospectiva dos Sprints. Como tarefa todos receberam alguns links com vídeos explicativos sobre métodos ágeis, Scrum e o Jira Software para aprofundarem os conhecimentos. A Figura 4 mostra o processo melhorado proposto.

Figura 4 - Processo de desenvolvimento proposto



Fonte: Elaborado pelo autor.

A primeira melhoria percebida no processo proposto (Figura 4) foi a triagem das requisições, já que no processo anterior elas seguiam do usuário final para o desenvolvedor, processo que promovia uma menor produtividade do time, além de, por vezes, gerar discussões desnecessárias entre usuários e programadores. O segundo grande avanço no processo foi a criação do *backlog* do produto, bem como institucionalização do papel de dono do produto. Com a triagem das requisições e enriquecimento de informações pelo time de suporte, as atividades chegam no *backlog* do produto mais detalhadas, facilitando o trabalho do dono do produto, que faz nova análise, podendo solicitar mais informações e/ou adicionar ao *backlog* priorizado.

Outro aspecto importante no novo processo, conforme representado na Figura 4, é o congelamento do escopo do time de desenvolvimento em ciclos curtos de

tempo, ou seja, a adoção dos *Sprints*, no processo de desenvolvimento. Entretanto o novo processo, mesmo com escopo fechado, permite que antes de iniciar e durante o *Sprint*, os desenvolvedores, possam solicitar novas informações sobre o *backlog* ao dono do produto, que poderá solicitar novas informações ao suporte e aos usuários envolvidos, com o objetivo de reduzir o tempo de desenvolvimento, refinar a atividade e melhorar a entrega.

4.2.2 Treinamento da equipe para o novo processo e transição

Após os primeiros contatos da equipe com a metodologia ágil Scrum, foi realizado um *workshop* para criação do *backlog* do produto, definição dos papéis, o primeiro Sprint backlog e ao mesmo tempo treinar a equipe para o novo processo de desenvolvimento. O primeiro trabalho foi fazer uma triagem dos *tickets* pendentes no sistema Hesk, separando por prioridade, projeto e classificando como histórias e tarefas.

Dada a organização de todas as pendências em aberto, foi efetuado o lançamento no software Jira, incluindo uma revisão das descrições, anexando documentos e imagens quando necessário, com o objetivo de melhorar o entendimento do que precisa ser realizado em cada atividade do backlog, criando assim a primeira versão do backlog do produto. Foi um trabalho desgastante que exigiu grande esforço da equipe, levando aproximadamente 10h para a conclusão, mas que deixou todos da equipe bastante satisfeitos com o resultado, pois já foi percebida a riqueza no processo de triagem melhorando as descrições e organizando as pendências por prioridade e projeto

No segundo dia de *workshop* a missão foi criar o *backlog* do primeiro Sprint, mas antes foi necessário definir os papéis de cada membro no time Scrum. Essa estratégia foi intencional, para não inibir a participação dos membros da equipe no processo de triagem e ao mesmo tempo tentar identificar de forma natural quem seria o líder da equipe, ou seja, no Scrum, quem seria o Mestre Scrum. Antes de iniciar o *workshop* a pessoa mais provável para ser Dono do Produto e Mestre Scrum era o gestor da área, mas há uma premissa de que o Mestre Scrum não deve ter a função de chefia, precisa estar presente no dia a dia da equipe e disponível para todas as cerimônias. Assim naturalmente um dos membros da equipe foi selecionado pela própria equipe para ser o Mestre Scrum de forma unânime, foi um processo muito

interessante, por ter sido de forma natural e de consenso, sem imposição da chefia, simplesmente o analista mais organizado destacou-se e recebeu o título de Mestre Scrum e desenvolvedor.

Com os papéis definidos, tendo como Dono do Produto o gestor da tecnologia da informação e como Mestre Scrum um dos membros da equipe de desenvolvimento, foi definido o prazo para os eventos e cerimônias, assim dado o fato da existência de um ambiente com escopo bastante variável foi definido que os Sprints terão a duração de 10 dias úteis, as reuniões diárias serão realizadas sempre às 14h, respeitando o limite de até 15 minutos de duração, a revisão do Sprint deve ser realizada até o sexto dia do Sprint ou a qualquer momento, sem restrição de tempo para a duração deste, pois vai depender da demanda, e as reuniões de retrospectiva e planejamento de um novo Sprint serão realizadas ao final do Sprint e terão a duração de até 4 horas cada uma.

Por fim, foi selecionado algumas atividades do *backlog* do produto, respeitando a ordem de prioridades, foi definido um objetivo para o primeiro Sprint, estimados os esforços de cada uma das atividades e criado o *backlog* deste Sprint, limitado à 50% do tempo disponível de cada desenvolvedor, considerando que durante o período de transição de processo haveria muito para aprender sobre a capacidade da equipe em gerenciar um escopo fechado, mesmo que seja em um curto período de tempo, bem como a capacidade de acerto no apontamento de estimativas de tempo das atividades do *backlog* do Sprint.

4.2.3 Mudanças no processo de desenvolvimento

Com o *backlog* do primeiro Sprint pronto, foram definidas as datas para início e fim do Sprint e aberto o Sprint para a realização das atividades e cerimônias. As primeiras reuniões diárias foram catastróficas, pois na prática a equipe não tinha entendido seu principal objetivo e para melhorar esse momento foi colocado em um quadro branco as três perguntas que devem ser respondidas na reunião diária e o objetivo do Sprint. Desta forma antes de iniciar a reunião o Mestre Scrum lembrava qual era o objetivo do Sprint e que as perguntas respondidas nesta reunião devem focar nesse objetivo.

Para acompanhamento das atividades de forma transparente optou-se por utilizar o quadro de atividades digital disponibilizado pelo Jira Software, recurso

substitutivo aos clássicos quadros físicos feitos com Post-it. A visualização do backlog do Sprint em um quadro de atividades digital permitiu a alteração dos estados das atividades de forma simples e auxiliou na transparência, bem como facilitou o acompanhamento. Com o objetivo de melhorar as reuniões diárias, o quadro foi disponibilizado em um monitor de vídeo grande, para que todos da equipe pudessem visualizar e acompanhar as atividades da equipe em todos os momentos, inclusive durante as cerimônias do Scrum.

Durante o Sprint, as novas solicitações que entraram pelo sistema Hesk, passaram pela triagem do suporte e quando caracterizado como atividade para a equipe de desenvolvimento, seja para manutenção, correção de erros ou melhorias, a equipe de suporte atribuía a tarefa ao Dono do Produto, para nova triagem e enriquecimento de informações antes de submeter ao backlog do produto, desta forma nenhuma nova atividade poderia ser enviada diretamente para o time de desenvolvedores.

O Dono do Produto analisava as requisições antes de adicioná-las ao backlog do produto, considerando a priorização por valor e enriquecendo a solicitação reescrevendo-a como backlog ou história de usuário, anexando documentos e conversando com os stakeholders quando necessário. Vale ressaltar que manter o escopo do Sprint fechado é o objetivo principal, mas como trata-se de equipe de desenvolvimento inserida num contexto dinâmico de um varejo supermercadista, atividades de cunho impeditivo ou urgente, aquelas que afetam um grande número de usuários, têm observância legal ou impactam diretamente no resultado da empresa, poderiam ser adicionadas ao Sprint de forma arbitrária pelo Dono do Produto, item que reflete diretamente no escopo e resultado do Sprint.

Na primeira reunião de revisão do Sprint com o time de desenvolvimento, Mestre Scrum e Dono do Produto, percebeu-se que muitas atividades entraram após o início do Sprint, gerando grande alteração no escopo inicial, entende-se que isso faz parte do processo de aprendizado da metodologia, por trata-se da execução do primeiro Sprint. Uma dificuldade apontada pelos desenvolvedores é a insistente comunicação direta de usuários com eles, esse é um dos maiores desafios, pois está diretamente vinculado a cultura da empresa, já que por muitos anos o desenvolvimento próprio funcionou assim, um a um, desenvolvedor e usuários construindo tudo sem qualquer triagem ou priorização por valor.

Por fim, fechado o primeiro ciclo Scrum, aconteceu a reunião de retrospectiva e percebeu-se que nem todos os entregáveis estavam prontos, alguns em decorrência das alterações de escopo e outros por não terem sido estimados corretamente. Ao mesmo tempo, alguns já haviam sido entregues, ou seja, nesse ponto nota-se uma característica do tipo de negócio e devido a especificidade do ambiente os Sprints normalmente terão mais de um objetivo e as entregas que não possuem interdependência serão realizadas no decorrer do Sprint. Também se identifica a necessidade de melhorar o processo de estimativas dos esforços em cada backlog, bem como reservar uma parte do tempo do time de desenvolvimento para as atividades impeditivas e urgentes, a fim de mitigar as alterações de escopo e garantir os entregáveis de acordo com os objetivos do Sprint.

4.3 Análise do processo implantado

Após a aplicação das melhorias no processo, coletou-se novos dados para analisar os resultados obtidos, que serão apresentados e discutidos nessa seção. Para tanto foram utilizadas formas de obtenção de informações que permitiram a análise do resultado em pelo menos dois aspectos: quantitativo, a partir da leitura e análise das informações do sistema de helpdesk, o Hesk, e também do recém implantado sistema de gestão do backlog, o Jira Software; qualitativo, aplicando questionário para avaliar a percepção dos colaboradores envolvidos.

4.3.1 Análise do questionário de avaliação do processo melhorado

O mesmo questionário aplicado antes da implantação do novo processo, foi utilizado para avaliar a percepção da equipe de desenvolvimento após seis Sprints completas, aproximadamente quatro meses da implantação. Os três desenvolvedores responderam todas as perguntas de forma anônima, seguindo a orientação da área de recursos humanos, o segundo questionário recebeu o nome de “Pesquisa de Reavaliação de Clima Interno” e pode-se avaliar seu conteúdo na integra através do APÊNDICE B.

Em comparação com as respostas anteriores, quanto a clareza das informações para se executar uma tarefa constatou-se avanço na percepção dos desenvolvedores, pois na primeira vez em que o questionário foi aplicado tinha-se

duas repostas afirmando que raramente as informações chegavam de forma clara e depois do processo teve-se duas respostas na opção “muitas vezes” e uma em “sempre”.

Outro ponto de melhoria identificado foi quanto à orientação da liderança na execução das tarefas, com a implantação do processo de triagem pelo time de suporte e posteriormente nova triagem e priorização pelo gestor de TI no papel de dono do produto, possibilitou-se melhoria significativa na comunicação e orientação para a execução das tarefas entre o gestor de TI e o time de desenvolvimento. Ainda falando de comunicação entre o time de desenvolvimento e a liderança de TI nota-se a partir das respostas que o time percebe uma melhoria na passagem de *feedback*, seja positivo ou negativo, provavelmente estimulado pelas cerimônias do Scrum, principalmente pela reunião diária, que cria uma certa cumplicidade entre os membros da equipe, por compartilharem diariamente informações sobre as atividades, o que evidencia os aprendizados e dificuldades a partir dos desafios inerentes a construção de softwares.

Quanto ao planejamento prévio para a realização das tarefas a equipe não percebeu melhoria, mantendo uma visão de que somente as vezes há de fato um planejamento. A percepção da equipe quanto aos recursos, equipamentos e materiais disponíveis para a realização das tarefas, se manteve de forma positiva, o que reafirma a existência de uma boa infraestrutura para o trabalho. Outro ponto identificado é que a percepção de autonomia na realização das tarefas, também melhorou bastante e de forma unanime entre o time de desenvolvimento com todas as respostas afirmando que na maioria das vezes eles se sentem autônomos na execução das tarefas, provavelmente influenciado por todo o envolvimento e riqueza de informação que o processo ágil permitiu com implantação das cerimônias e organização do backlog.

Para fechar a análise do questionário nas três perguntas finais, sobre carga de trabalho, colaboração e relacionamento entre membros da equipe, percebe-se pequenas melhorias. Na carga de trabalho na primeira vez em que o questionário foi aplicado teve-se uma percepção mais negativa e na segunda não há melhoria significativa, mantendo todas as respostas entre “ruim” e “regular”. Quanto a colaboração entre membros, há pequena melhoria, pois ela já era boa conforme as respostas da primeira aplicação do questionário, mas após as mudanças, todos concordaram ter uma boa colaboração entre eles. No relacionamento entre os

colaboradores, manteve-se positivo, exatamente igual à primeira aplicação do questionário.

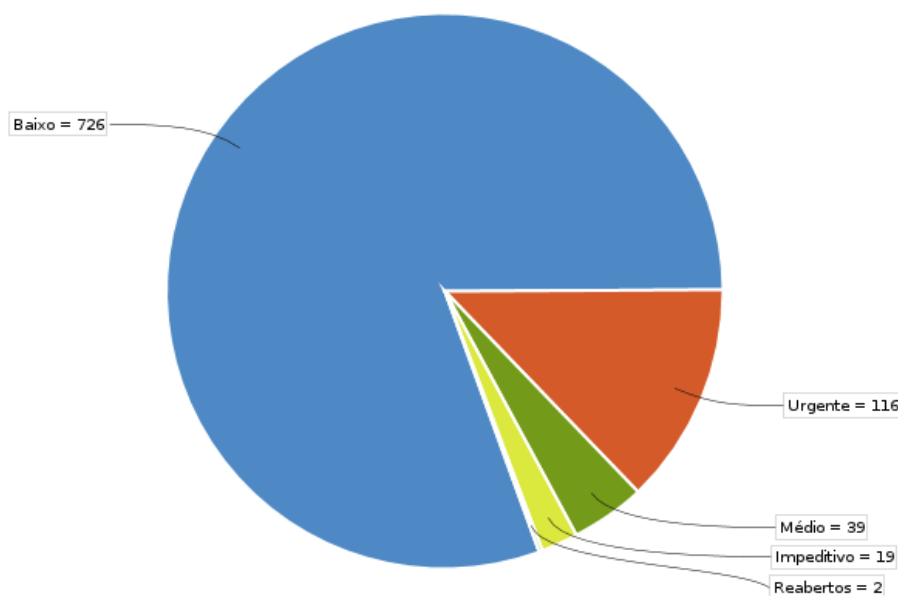
4.3.2 Análise das informações dos sistemas de apoio e gestão do backlog

Para a validação do processo proposto utilizou-se o próprio sistema de requisições de suporte, alteração e reporte de erros do SGE acessível aos usuários(Hesk). Os dados foram coletados no período entre os meses de janeiro até abril do ano de 2019, ocorrendo após a implantação do método ágil.

Observando-se os dados coletados via o sistema Hesk, não percebe-se alterações nas características das requisições, nem resultados significativos quanto à quantidade de *tickets* de alta prioridade no período analisado. Isto ocorreu pelo fato de que o escopo do trabalho não abordava em alterações na forma como os usuários incluem novas solicitações, seja para reportar problema, pedir orientação ou requisitar alteração na funcionalidade do software. Embora, a aplicação do processo de triagem no time de suporte, tenha aumentado significativamente a troca de mensagens com os usuários, o que permitiu maior clareza e documentação do que o usuário realmente está solicitando, mas ainda não refletiu no modo como os usuários fazem a primeira requisição.

Já com relação ao software Jira, pode-se perceber que houve melhoras na geração de informações, o que era inexistente com o uso do processo anterior. O período de coleta de informações foi referente a seis *sprints* completos, compreendido entre os meses de janeiro a abril do ano de 2019. Com o Jira, foi possível verificar que, apesar das requisições no Hesk não terem sofrido grandes alterações, a equipe está trabalhando melhor com a priorização do *backlog* a partir do processo de triagem, como pode ser visto no gráfico da Figura 5.

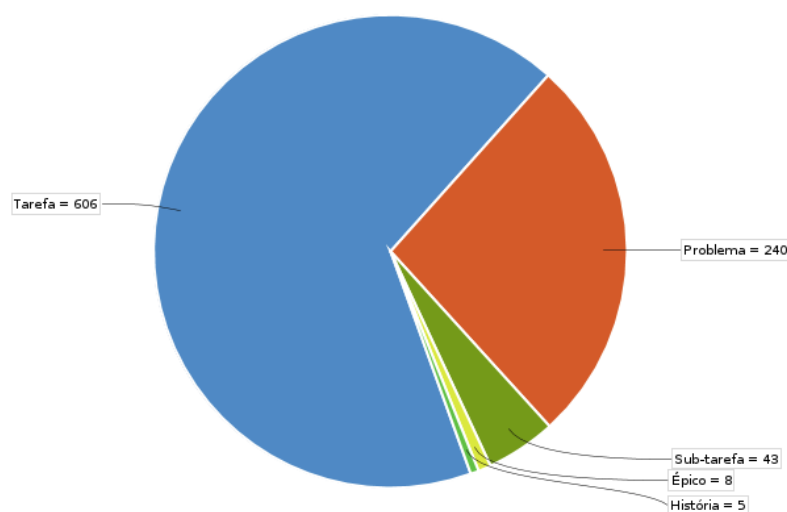
Figura 5 – Gráfico de pendências no backlog por prioridade



Fonte: <http://jira.hippo.com.br> acessado em 25/05/2019.

No gráfico da Figura 5, que representa as pendências no *backlog* por prioridade, mostra que a quantidade de itens concentram-se agora na categoria de nível baixo de prioridade, o que permite ter um ambiente mais controlado, apesar de haver algumas pendências impeditivas e urgentes que resultam em frequentes alterações de escopo nos *Sprints*. A Figura 6 representa graficamente o backlog por tipo de pendência, com ele é possível verificar novamente a importância do processo de triagem e enriquecimento das informações na documentação das atividades.

Figura 6 – Gráfico de backlog por tipo de pendencia



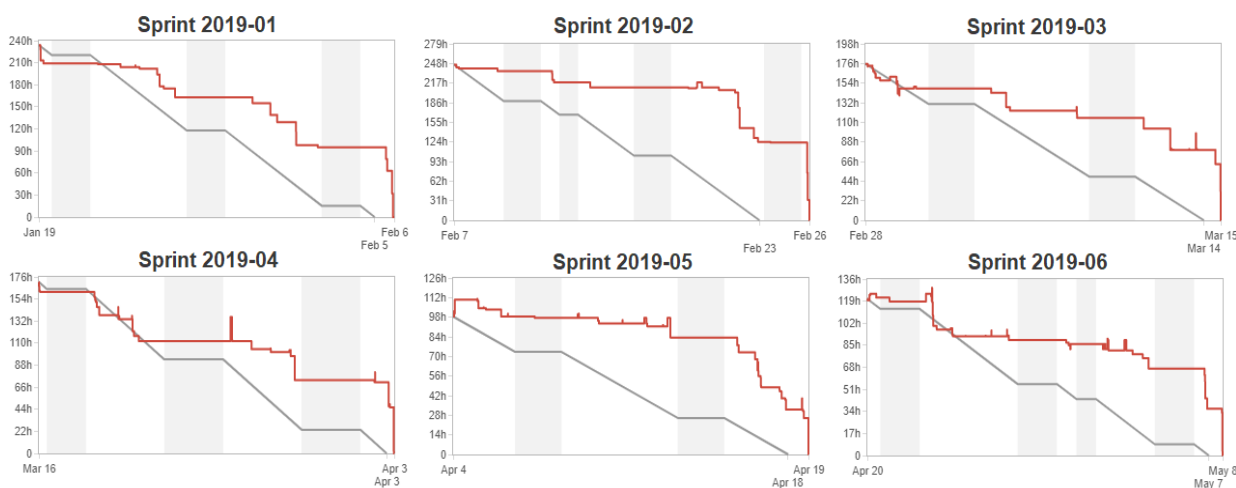
Fonte: <http://jira.hippo.com.br> acessado em 25/05/2019.

Com a representação gráfica do backlog por tipo de pendência, conforme Figura 6, percebe-se uma melhora na distribuição das tarefas, com alta concentração das atividades na produção de tarefas, contra um menor índice de trabalho na resolução de problemas. Como os dados apresentados no gráfico da Figura 6 pode-se afirmar que, a cada quatro atividades no *backlog*, é provável existir três novas tarefas e um problema para ser resolvido, considerando a perspectiva desse gráfico temos que um quarto de todo o esforço do time de desenvolvimento está alocado na resolução de problemas.

Uma nova ferramenta para a análise da performance do time de desenvolvimento, foi conhecida pela equipe durante a implantação do *Scrum*, trata-se do gráfico de *Burndown*, que segundo Wazlawick (2013), “consiste basicamente de duas linhas: a primeira indica a quantidade de trabalho por fazer e a segunda indica a quantidade de trabalho feito ao longo do tempo”.

Na Figura 7 pode-se visualizar o gráfico de *Burndown* dos seis primeiros Sprints finalizados, a linha cinza representa o escopo dos *Sprints*, ou seja, a quantidade de trabalho selecionada e estimada para ser realizado no *Sprint*, que trata-se do esforço estimado para a realização das tarefas do *Sprint Backlog*, a linha vermelha representa o trabalho que realmente foi realizado pela equipe, considerando os erros de estimativas e as alterações de escopo do período, desta forma quanto mais a linha vermelha se aproximar da linha cinza, melhor foram as estimativas realizadas pelo time de desenvolvimento e também menor foram as alterações de escopo do *Sprint Backlog*.

Figura 7 – Gráficos de Burndown dos seis primeiros Sprints

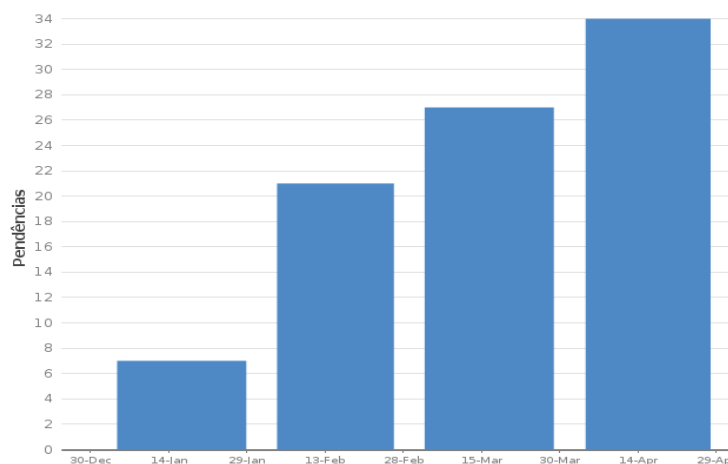


Fonte: <http://jira.hippo.com.br> acessado em 25/05/2019.

A leitura dos gráficos de *Burndown*, conforme Figura 7, demonstram uma característica do ambiente de trabalho no Hippo, a persistente alteração de escopo, dado o dinamismo das atividades no varejo de supermercado e a atual situação dos recursos do sistema SGE, que demandam manutenção significativa, fruto de um legado de anos sem qualquer metodologia de trabalho. Outro ponto para se observar, refere-se ao fato de que ainda há muito que se aprender quanto ao planejamento das *Sprints*, pois mesmo com a percebida melhoria na triagem e informações disponibilizadas para a realização das atividades, boa parte do distanciamento da linha vermelha em relação a cinza não se dá apenas pelas alterações de escopo, mas também pela deficiente estimativa das tarefas, na qual, em sua maioria foram subestimadas, concretizando em esforço maior do que o estimado pelo time de desenvolvimento no início dos *Sprints*.

Para finalizar a análise das informações concebidas com a implantação do Jira Software no processo de desenvolvimento do Hippo, observa-se a partir do gráfico de histórico de backlog resolvidos entre o período de janeiro até abril de 2019, conforme representado na Figura 8, o crescente aumento de resolução de pendências a cada mês.

Figura 8 – Gráfico de resolução de backlogs por mês



Fonte: <http://jira.hippo.com.br> acessado em 25/05/2019.

No gráfico de resolução de *backlogs* por mês representado na Figura 8, verifica-se que o time vem evoluindo na quantidade de *backlogs* resolvidos de forma bastante significativa a cada mês, saindo de menos de dez entregues no primeiro mês, para mais de trinta entregas no quarto mês do novo processo de desenvolvimento.

Ao relacionar os gráficos da Figura 7, gráficos de *Burndown*, com os da Figura 8, resolução de pendências, nota-se que o processo de desenvolvimento ágil é evolutivo e vem promovendo melhor autoconhecimento das capacidades da equipe, seja na hora de estimar as tarefas, bem como de resolve-las, dado o crescente número de entrega versus os gráficos de *Burndown* distantes do trabalho estimado.

5 CONSIDERAÇÕES FINAIS

Ao final do trabalho realizado na empresa alvo, pode-se afirmar que um processo de desenvolvimento ágil seguindo as práticas e recomendações do *Scrum* foi implantado e que pode-se perceber melhora nos resultados dos processos de desenvolvimento e suporte, apesar desse último não ter sido alvo deste trabalho. Nesse sentido a primeira conclusão é que a adoção de métodos ágeis em ambientes de trabalho tende a permitir um processo de melhoria continua evolutivo.

No início do trabalho, conforme visto em trabalhos relacionados, era esperado grande resistência de colaboradores antigos, o que não se configurou na prática. Com uma abordagem de construção de processo colaborativo, focada na melhoria do trabalho, e não em seguir um plano à risca, conforme descrito na seção 4.2, somado a uma característica intrínseca da empresa, que trabalha a valorização do ser humano e o relacionamento interpessoal de forma muito latente, o resultado foi que nos momentos de resistência discussões foram abertas, abordando os conflitos e divergências de ideias de forma construtiva. Então apesar de serem colaboradores antigos, com muitos anos de empresa, pode-se perceber a maturidade desses profissionais, além do compromisso e responsabilidade com a empresa. Essa característica refletiu nas pesquisas aplicadas antes e depois da implantação do processo ágil, que se mantiverem em crescente sinergia o relacionamento e colaboração entre os membros do time.

A partir do momento que melhorias na forma de trabalhar eram percebidas, o engajamento da equipe aumentava, em contraponto a inercia de anos sem qualquer método de desenvolvimento consciente, nesse item é possível observar que havia um anseio por mudança retraído, pois a ausência de método implementava um ambiente de alto stress, onde tudo era urgente e prioritário. E após a implantação do método ágil nota-se crescente evolução na produtividade, conforme analisado na seção 4.3,

relativa aos resultados percebidos a partir da leitura de informações geradas pelos sistemas de apoio e gestão.

Para finalizar, pode-se concluir que os objetivos da pesquisa foram alcançados, resultando no sucesso da aplicação de método ágil e no aumento da produtividade da equipe de desenvolvimento sem adicionar novos membros. A análise de resultados realizada deixou evidente as melhorias conquistadas, com o enriquecimento das informações nas requisições, com a redução de requisições de alta prioridade e manutenção de problemas, com o significativo aumento crescente do número de entregas dos backlogs a cada *Sprint* e com o aumento do engajamento dos colaboradores envolvidos no processo, assim respondendo-se a questão de pesquisa apresentada ao iniciar este projeto, a qual teve como provocação à realização desta pesquisa alcançar a resposta ao questionamento, claro que no contexto da empresa Hippo: **A aplicação de métodos ágeis no processo de desenvolvimento de software pode melhorar a produtividade da equipe de desenvolvimento, sem necessitar adicionar colaboradores?**

Entretanto é importante destacar que durante o processo e a análise de resultados, percebeu-se algumas necessidades que abrem espaço para trabalhos futuros, os quais devam permitir a melhoria contínua e evolução no método de estimativa de tempo para a resolução das tarefas, nas estratégias de comunicação com os usuários, especialmente no que se refere ao fato de encontrar um ponto de equilíbrio na relação entre o *backlog* dos *Sprints* e a frequente necessidade de alteração de escopo, pois na relação com usuários notou-se grande dificuldade no entendimento da importância do escopo fechado para a evolução do produto desenvolvido pelo time e o valor que essa estratégia pode gerar para a empresa ao longo do tempo.

REFERÊNCIAS

CALSAVARA, A. R. **Atacarejo: Estratégia Planejada ou Mera Coincidência?**. Mestrado em Administração - Programa de Mestrado e Doutorado em Administração, Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, 2009. Disponível em: <http://revista.unicuritiba.edu.br/index.php/admrevista/article/viewFile/72/47>. Acesso em: 20 abr. 2019.

Departamento de Economia e Pesquisa da Abras. Associação Brasileira de Supermercados (ABRAS) - Site. **Pesquisa postada no site ABRAS**. Disponível em: <http://www.abras.com.br/economia-e-pesquisa/indice-de-vendas/historico/>. Acesso em: 17 março 2018.

FERREIRA, F. I. G. **Proposta de Implantação de um Novo Processo de Software Baseado no Scrum em uma Empresa de Software ERP**. TCC (Graduação) - Curso de Bacharel em Ciência da Computação, Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, 2016.

GUIA SBOK. **Guia SBOK: Um Guia para o Conhecimento em Scrum** – Edição 3rd, SCRUMstudy, VMEdU, Arizona, USA, 2017. Disponível em: <https://www.scrumstudy.com>. Acesso em: 02 dez. 2018.

Hippo Institucional - Site. **Texto postado no site Hippo**. Disponível em: <http://www.hippo.com.br/sobre-o-hippo/>. Acesso em: 17 março 2018.

LACERDA, D. P. et al. **Gestão por processos**. Unisinos, São Leopoldo, 2010

MATTOS, A. **Queda de vendas leva a recorde negativo**. São Paulo, 31 janeiro 2017. Texto postado no Portal Valor Econômico o portal da revista Valor Econômico. Disponível em: <http://www.valor.com.br/empresas/4852852/queda-de-vendas-leva-recorde-negativo>. Acesso em: 17 março 2018.

MATTOS, A. **Vendas reais de supermercados subiram 1,25% em 2017, aponta Abras**. São Paulo, 29 janeiro 2018. Texto postado no Portal Valor Econômico o portal da revista Valor Econômico. Disponível em: <http://www.valor.com.br/empresas/5288789/vendas-reais-de-supermercados-subiram-125-em-2017-aponta-abras>. Acesso em: 17 março 2018.

MATTOS, A. **Volume de vendas em supermercados cai 1,2% em 2015**. São Paulo, 29 março 2016. Texto postado no Portal Valor Econômico o portal da revista Valor Econômico. Disponível em: <http://www.valor.com.br/empresas/4502378/volume-de-vendas-em-supermercados-cai-12-em-2015>. Acesso em: 17 março 2018.

O GLOBO. **Vendas nos supermercados fecham 2017 com alta real de 1,25%**. Rio de Janeiro, 21 janeiro 2018. Texto postado no site do O GLOBO o portal do jornal O GLOBO. Disponível em: <https://oglobo.globo.com/economia/vendas-nos-supermercados-fecham-2017-com-alta-real-de-125-22342051>. Acesso em: 17 março 2018.

PRESSMAN, R. e MAXIM, B. **Engenharia de software: uma abordagem profissional**, 8.ed. Bookman, Porto Alegre, 2016

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum: Um Guia Definitivo para o Scrum: As Regras do Jogo.** 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Portuguese-Brazilian.pdf>>. Acesso em: 02 dez. 2018.

SILVA, D. L. B. **O Impacto da Adoção de Metodologias de Desenvolvimento de Sistema em Empresas de Pequeno Porte.** TCC (Tecnólogo) - Curso de Tecnólogo em Processamento de Dados, Faculdade FATEC (FATEC), São Paulo, 2011.

SOMMERVILLE, I. **Engenharia de software.** 9. ed. Pearson Prentice Hall, São Paulo, 2011

SOUZA, D. R. **Implantação da Metodologia Ágil Scrum em um Ambiente de Desenvolvimento.** TCC (Graduação) - Curso de Bacharel em Tecnologias da Informação e Comunicação, Universidade Federal do Estado de Santa Catarina(UFSC), Araranguá, 2014.

SUTHERLAND, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo,** Leya, São Paulo, 2014

WAZLAWICK, R. S. **Engenharia de software: conceitos e práticas.** Elsevier, Rio de Janeiro, 2013

ATLASSIAN. **Jira Software.** Texto postado no site da ATLASSIAN a proprietária do software Jira Software. Disponível em: <<https://br.atlassian.com/software/jira>>. Acesso em: 10 janeiro 2019.

THIOLLENT, M. **Metodologia da pesquisa-ação.** 2. ed. Cortez, São Paulo, 1986.

BORDENAVE, J. D.; PEREIRA, A. M. **Estratégias de ensino-aprendizagem.** 12. ed. Vozes, Petrópolis, 1991.

AZEVEDO, D. et al. **Métodos e procedimentos de pesquisa: do projeto ao relatório final.** Unisinos, São Leopoldo, 2011

APÊNDICE A – QUESTIONÁRIO ANTES DO MÉTODO ÁGIL

Tabela questionário Clima equipe desenvolvimento (3 participantes)

Perguntas/Respostas	1-Nunca	2-Raramente	3-Às vezes	4-Muitas vezes	5-Sempre
As informações para a execução das tarefas chegam de forma clara:	0	2	1	0	0
Você recebe orientação da liderança para executar suas tarefas:	0	0	1	2	0
Você recebe feedback positivo (elogios) e negativo (pontos de melhoria) da sua liderança:	0	2	1	0	0
Neste setor existe um planejamento prévio para a realização de tarefas:	0	1	2	0	0
Você tem disponível os materiais, equipamentos e os recursos necessários para realizar suas tarefas:	0	0	0	2	1
Você considera que possui autonomia (liberdade) para realizar suas tarefas:	0	1	2	0	0
Perguntas/Respostas	1-Péssima	2-Ruim	3-Regular	4-Bom	5-Excelente
Como você considera sua carga de trabalho (quantidade de tarefas na jornada diária):	0	2	1	0	0
A colaboração entre os integrantes da sua equipe de trabalho é:	0	0	1	2	0
Como você avalia o relacionamento entre os colaboradores no setor:	0	0	0	2	1

APÊNDICE B – QUESTIONÁRIO APÓS IMPLANTAR MÉTODO ÁGIL

Tabela questionário Clima equipe desenvolvimento (3 participantes)

Perguntas/Respostas	1-Nunca	2-Raramente	3-Às vezes	4-Muitas vezes	5-Sempre
As informações para a execução das tarefas chegam de forma clara:	0	0	0	2	1
Você recebe orientação da liderança para executar suas tarefas:	0	0	0	1	2
Você recebe feedback positivo (elogios) e negativo (pontos de melhoria) da sua liderança:	0	0	2	1	0
Neste setor existe um planejamento prévio para a realização de tarefas:	0	1	2	0	0
Você tem disponível os materiais, equipamentos e os recursos necessários para realizar suas tarefas:	0	0	0	2	1
Você considera que possui autonomia (liberdade) para realizar suas tarefas:	0	0	0	3	0
Perguntas/Respostas	1-Péssima	2-Ruim	3-Regular	4-Bom	5-Excelente
Como você considera sua carga de trabalho (quantidade de tarefas na jornada diária):	0	1	2	0	0
A colaboração entre os integrantes da sua equipe de trabalho é:	0	0	0	3	0
Como você avalia o relacionamento entre os colaboradores no setor:	0	0	0	2	1