

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/303056197>

ESTUDO DE CASO: ABORDAGEM ÁGIL SCRUM PARA DESENVOLVIMENTO DE SOFTWARE EM AMBIENTE ITIL

Experiment Findings · May 2016

DOI: 10.13140/RG.2.1.3261.2724

CITATIONS

0

READS

2,043

2 authors:



[Calvin Bersch Scherer](#)

Universidade do Sul de Santa Catarina (Unisul)

1 PUBLICATION 0 CITATIONS

SEE PROFILE



[Oscar Dalfovo](#)

Sociedade Educacional de Santa Catarina (SOCIESC)

85 PUBLICATIONS 46 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Revista Interdisciplinar Científica Aplicada [View project](#)



Gestão e Tecnologia da Informação e do Conhecimento [View project](#)

ESTUDO DE CASO: ABORDAGEM ÁGIL SCRUM PARA DESENVOLVIMENTO DE SOFTWARE EM AMBIENTE ITIL

Prof. Calvin Bersch Scherer, Esp - calvin.scherer@gmail.com

FIESC / Especialista em Engenharia de Projetos de Software – UNISUL

Prof. Dr. Oscar Dalfovo – odalfovo@gmail.com

Professor e Pesquisador da UNISOCIESC Depto de Administração e Tecnologia / UNISUL

Pós-graduação Engenharia de Projetos de Software

RESUMO

Nas diversas abordagens metodológicas para desenvolvimento de software se destacam por seguirem valores e princípios ditos “ágeis”. O alto nível de valor agregado para os produtos, serviços de tecnologia, processos e negócios faz com que organizações busquem por soluções a fim de usufruir destes benefícios. O objetivo do presente trabalho propõe melhorias nos processos de uma equipe de tecnologia e desenvolvimento de software que adota o *Information Technology Infrastructure Library* (ITIL) para gerenciar seus serviços. As melhorias propostas estão alinhadas com as metodologias ágeis para desenvolvimento de software, tendo em vista os benefícios que esta abordagem proporciona. Após estudos e análise de um cenário real fornecido pela empresa estudada, o trabalho apresentou uma proposta de melhoria utilizando a técnica de estimativa ágil *Planing Poker* para ser aplicada junto com a ITIL.

Palavras-chave: Estudo de Caso, Desenvolvimento Ágil, SCRUM, ITIL.

ABSTRACT

Several methodological approaches to software development distinguish to follow values and principles called by "agile". The high level of value added for the products, technology service, processes and business does the companies take solutions to uses these benefits. This labour proposes improvements at process of a technology team and software development that uses the Information Technology Infrastructure Library (ITIL) to management its services. These improvements are according with agile methodology to software developed, according to the benefits that this point of view takes. After studies and analysis of a real situation provided from studied company, this labour produced a proposal of uplift using the agile estimative technique *Planing Poker* to apply with the itil.

Keywords: Case Student, Agile, SCRUM, ITIL.

1 INTRODUÇÃO

Os últimos anos foram marcantes para indústria de software com o advento do manifesto ágil para desenvolvimento de software em 2001, conforme apresentado pelo “Mercado Brasileiro de Software – Panorama e Tendências – 2014” publicado anualmente pela Associação Brasileira de Empresas de Software (ABES). A partir dele, ao longo dos anos, os processos, as metodologias e as ferramentas para desenvolvimento de software, rotuladas como “tradicionais”, foram substituídos ou remodelados para seguirem os princípios e valores ágeis. Os novos conceitos surgiram para orientar as equipes de desenvolvimento na entrega de software utilizando tais valores ágeis. O gerenciamento tradicional, aquele fundamentado nos “corpos de conhecimento”, ganha um novo olhar quando o resultado esperado envolve, principalmente, algum tipo de inovação para a empresa, para o mercado ou para o mundo. Projetos de software geralmente são encarados como sendo de âmbito inovador para a organização.

A demanda por sistemas mais complexos, abrangentes, flexíveis e funcionais fez com que fornecedores de soluções de software tivessem que se adaptar e buscar diferenciais competitivos para seus produtos e também processos. Em resposta ao desafio de produzir produtos de softwares, inúmeras alternativas (filosofias, arquiteturas, paradigmas, metodologias) têm sido propostas para garantir a criação de sistemas de valor no intuito de apresentar produtos de softwares competitivos ao mercado. Apesar desses esforços, a produção de um software confiável, correto e entregue dentro dos prazos e custos estipulados ainda não é uma realidade para esta área. Ainda não há consenso quanto a um processo de desenvolvimento de software que seja eficaz em todas as situações. Não obstante, algumas alternativas vêm apresentando bons resultados em diversos contextos. Dentre elas estão os processos, métodos e ferramentas ligadas aos valores e princípios do Manifesto para Desenvolvimento Ágil de Software (PRESSMAN, 2011) (BECK et al., 2001).

Conforme Shore (2008), as alterações na faculdade de desenvolvimento de software foram motivadas e impulsionadas por um mercado consumidor ansioso por soluções inovadoras. Pressman (2011) complementa o pensamento apresentando a existência de um quadro de intensa atualização tecnológica, tanto de hardware quanto de software, aliada à popularização do acesso a recursos para desenvolvimento de sistemas, o que aumenta ainda mais a competitividade entre os sistemas disponibilizados. Como resultado, tem-se um mercado com desafios imensos para fornecedores de soluções de TI no geral e de software em particular.

Assim como o software ganha importância no mercado pela imensa aplicabilidade num mundo tecnológico e, cada vez mais, conectado, a tecnologia da informação passou de uma área de apoio para uma área de negócios – *core* – da organização, atuando no planejamento estratégico. Em outras palavras, a TI deixou de ser um provedor de apoio para se tornar uma parceira fundamental no negócio da organização (OGC, 2007). Mediante este cenário, a TI passou a receber incentivo de estudos e experimentos a fim de alcançar uma estrutura propícia à extrair seu melhor para a organização (MAGALHÃES; PINHEIRO, 2007).

Esse novo cenário fez com que diversas propostas surgissem para aperfeiçoar processos e também as metodologias a cerca desses. A ITIL, um *framework* desenvolvido para o gerenciamento de processos e serviços da TI, foi idealizado visando tornar a TI parte estratégica da organização e não tática e operacional. Ela descrever um conjunto de melhores práticas de gestão, buscando usufruir de todos os serviços que a TI pode oferecer de forma eficiente e eficaz (MANSUR, 2005).

O trabalho tem como objetivo principal desenvolver uma proposta utilizando métodos ágeis aplicados em processos ITIL da empresa estudada. A empresa estudada no presente trabalho é chamada de Empresa X por questões de sigilo e segurança, a pedido da mesma. A organização é real, privada, considerada de grande porte e atuante há mais de sessenta anos em todo o território nacional. A Empresa X possui diversas frentes e áreas de atuação, sendo uma delas a área de tecnologia da informação e prestação de serviços de TI. Para isso, a organização utiliza o ITIL para orientar seus processos de desenvolvimento de sistemas e gestão de serviços. A atuação consiste em manter e desenvolver funcionalidades para um sistema de gestão de negócio. Por conta das características da Empresa X, essa servirá de base para análise e desenvolvimento da proposta de melhoria com base em métodos ágeis objetivada no trabalho.

2 REVISÃO DA LITERATURA

Neste capítulo intitulado revisão da literatura apresenta os conceitos introdutórios e necessários para fomentar a base do desenvolvimento do trabalho acadêmico.

2.1 MANIFESTO ÁGIL

Um grupo formado por dezessete profissionais de TI se reuniram em 2001 para discutir as técnicas existentes para projetos de desenvolvimento de software. O objetivo deles era descobrir novos valores e princípios para melhorar o desenvolvimento de software. O

resultado do grupo, autodenominado *Agille Alliance* (Aliança Ágil), foi uma nova ideia denominada: Manifesto para Desenvolvimento Ágil de Software (BECK et al., 2001).

Os valores e princípios ágeis preconizados por essa aliança e a sua relação com os valores tradicionais foram apresentados para ajudar as equipes a entregar melhores produtos de software. A estratégia está em valorizar determinadas ações, como: indivíduos e as interações entre eles; software funcionando; colaboração com o cliente e responder as mudanças mais do que outras ações que pertencem aos processos ditos tradicionais, como: processos e ferramentas; documentação abrangente; negociação de contratos e seguir um plano, respectivamente (BECK et al., 2001). Os valores ágeis preconizados e prezados por essa aliança e a sua relação com os valores tradicionais (não prezados) definidos no manifesto ágil são apresentados na Tabela 1.

Tabela 1 - Valores do Manifesto Ágil

Valor Ágil (prezado)	Valor Tradicional (não prezado)
Indivíduo e interação entre eles	Processos e ferramentas
Software funcionando	Documentação abrangente
Colaboração com o cliente	Negociação de contratos
Responder a mudanças	Seguir um plano

Fonte: Beck et al. (2001)

A valorização dos indivíduos e interações entre eles mais que processos e ferramentas dentro do ambiente de desenvolvimento de software leva em conta que, em última instância, são os indivíduos que, como talento e habilidade, iram gera produtos e serviços através de suas características únicas expostas individualmente ou em equipe (HIGHSMITH, 2004). Conforme Highsmith (2002), as habilidades, personalidades e peculiaridades dos indivíduos são críticas para o sucesso dos projetos. Os indivíduos serão muitas vezes desorganizados e difíceis de entender, ao mesmo tempo em que são inovadores, apaixonados, criativos e exuberantes.

O trabalho em equipe e as habilidades individuais são características inseparáveis em projetos de desenvolvimento de software, sem ela não há como aplicar princípios da agilidade. Cabe ressaltar que os processos têm o propósito de guiar e apoiar o desenvolvimento de software, enquanto a ferramenta é utilizada para melhorar a eficiência dos indivíduos (HIGHSMITH, 2002). Processo deve ser adaptável à equipe de trabalho e deve ajudar na execução do projeto, não sendo substituinte de uma habilidade individual. Não é

possível obter sucesso nos projetos de software ao buscar seguir exclusivamente um processo por si só, mesmo com a adaptação ao mesmo.

Software funcionando mais do que documentação abrangente leva em consideração que os clientes se interessam por resultados palpáveis e visíveis, ou seja, software em funcionamento que entregue valor de negócio (HIGHSMITH, 2002). A entrega contínua do software em funcionamento possibilita um *feedback* confiável do processo de desenvolvimento e do produto por parte do cliente e usuários de uma forma que seria impossível conseguir através de uma documentação (HIGHSMITH, 2004).

A documentação é necessária para criar um canal de comunicação e colaboração entre a equipe, também para melhorar a gestão do conhecimento na organização, para documentar informações históricas dos sistemas, para sanar questões legais e regulatórias, entre outras necessidades. Encara-se, para fins, que a documentação é menos importante que versões funcionais do software. Sua existência nos projetos deve ser para facilitar a interação entre a equipe, funcionando como seu subproduto na forma de documentos, rascunhos, desenhos, anotações, entre outros (HIGHSMITH, 2004).

A colaboração com o cliente mais do que negociar contratos é um valor ágil que mostra a necessidade que ambos possuem na busca pela qualidade do produto de software. A relação entre a organização e o cliente deve existir sob o companheirismo, tomada de decisão conjunta e rapidez na comunicação para que as mudanças e as adaptações possam fluir de modo sustentável. Conforme Highsmith (2004), deve ser levado em conta que existe alta volatilidade, ambiguidade e incertezas no desenvolvimento de software, e que a relação entre o cliente e a equipe deve ser marcada por colaboração e compreensão, ao invés de negociações antagônicas de contratos de diversas naturezas.

A resposta às mudanças mais do que seguir um plano vem ao encontro dos projetos onde há alto nível de incertezas e a adaptabilidade dos projetos. A investigação e a concepção mental predominam sobre o planejamento e a execução restrita de tarefas planejadas quando há projetos incertos e com alto nível de inovação. A intenção é balancear o planejamento com o imprevisto no projeto utilizando o grau de incertezas e as constantes mudanças para direcionar as ações e atividades da equipe (HIGHSMITH, 2004).

Beck et al. (2001) apresenta um conjunto de doze princípios ágeis baseado nos quatro valores ágeis que também fazem completam o manifesto. Seu propósito é auxiliar e nortear as organizações a cumprirem os valores apresentados, dando embasamento e entendimento completo.

Utiliza-se os valores prezados e os princípios na fundamentação do manifesto ágil. Diversas ferramentas, metodologias e processos de desenvolvimento de sistemas surgiram ou foram reformuladas para atender o manifesto (BECK et al., 2001). Por consequência, receberam o título “ágil”.

As diretrizes do Manifesto Ágil são abrangentes e, conseqüentemente, existem diversas metodologias que as seguem. Recebendo o título de metodologias ágeis para o desenvolvimento de software, tais surgiram como uma alternativa para gerar resultados imediatos, sob as condições de constante mudança e incerteza, típicas do desenvolvimento de um produto de software.

2.2 MÉTODOS ÁGEIS

O termo “agilidade” começou a ser empregado no setor de tecnologia em meados dos anos 90. Era usado nas iniciativas de observar as dificuldades que existiam nos ambientes e modelos prescritivos, detalhados e lineares e melhorá-lo. O principal modelo adotado pela comunidade de software na época e que tinha essas características era o modelo em cascata, ou *waterfall*, descrito por Royce em 1970. O modelo em cascata era caracterizado por um conjunto bem dividido e sequenciado de fases de desenvolvimento. Recebeu o nome “cascata” porque cada processo somente iniciava-se após o término do seu posterior (PRESMANN, 2006 apud SCHERER, 2012). O modelo ficou conhecido como “tradicional” após o advento do manifesto para desenvolvimento ágil de software. Conforme Pressman (2011, p. 82), “os métodos ágeis se desenvolveram em um esforço para sanar fraquezas reais e perceptíveis da engenharia de software convencional”.

Em 1986, Frederick Brooks previu que o desenvolvimento de software utilizando metodologia linear, como o método cascata, não ofereceria um nível aceitável para o gerenciamento de sistemas. Não haveria, por volta dos anos de 1996, uma metodologia, técnica, ferramenta ou tecnologia que poderia oferecer um grande aumento em produtividade, confiança e simplicidade. (BROOKS, 1995). E conforme foi previsto por Brooks (1995), não havia algo que conseguisse acompanhar o ritmo de crescimento dos sistemas desenvolvidos.

Complementando e dando embasamento ao surgimento do desenvolvimento ágil e complementando o pensamento de Brooks (1995), Pressman (2011, p. 82) descreve que “na economia moderna é frequentemente difícil ou impossível prever como um sistema computacional (...) irá evoluir com o tempo”. Métodos tradicionais não conseguem acompanhar à dinâmica exigida pelos produtos de software. Finalizando seu conhecimento, Pressman (2011, p. 82) conclui que “é preciso ser ágil o suficiente para dar uma resposta ao

ambiente de fluido negócios”. Para o autor, para haver fluidez no projeto de software deve também haver a gestão das mudanças. Quando um projeto de software aceita as condições, valores e práticas ágeis, o mesmo deve adotar uma postura controlada e monitorada das mudanças a fim de diminuir os custos envolvidos. Em resposta a isso, um manifesto criticando a forma de desenvolvimento de software foi criado para proporcionar a geração de resultados num ambiente de alta incerteza e mudanças constantes.

De acordo com Shore (2008, p. 3), “o desenvolvimento ágil também não é uma solução”. A adoção de práticas e métodos ágeis para o desenvolvimento de software deve ser feita quando o desenvolvimento ágil ajudar a organização a ter mais sucesso. Conforme complementa Shore (2008, p.3) “quando conseguir responder a esta pergunta, saberá se o desenvolvimento ágil é apropriado para você”.

Reconhece-se que o desenvolvimento ágil oferece benefícios e vantagens importantes que podem ser um diferencial significativo ao projeto, entretanto há características que não são indicadas para todos os produtos, estruturas, pessoas e projetos. Tanto os defensores dos métodos ágeis quanto por defensores do método “tradicional” aceitam que a verdadeira questão é: Qual é a melhor forma de atingir a agilidade? Como desenvolver um produto de software que atenda as necessidades e a qualidade do cliente? Conforme o pensamento de Pressman (2011, p. 84) sobre o isso, “não há resposta absoluta para nenhuma dessas questões”. A existência da agilidade em projetos não faz com que ferramentas, métodos, processos e princípios da engenharia de software sejam descartados, mas sim evoluídos e adaptados para uma nova realidade. Vários conceitos ágeis são adaptações dos bons conceitos tradicionais da engenharia de software e que outras ideias são totalmente afastadas ou opostas desses conceitos. “Pode-se ganhar muito considerando o que há de melhor nas duas escolas e praticamente nada denegrindo uma ou outra abordagem” apresenta Pressman (2011, p. 84).

A agilidade consiste em responder as mudanças de forma eficiente, contudo o termo abrange uma filosofia que incentiva a estruturação e a atitude dentro de uma equipe de trabalho, fundamentando na comunicação simples, rápida e fácil. Reconhece que deve existir uma entrega contínua, constante e rápida de trabalho e que o sucesso do projeto parte da exploração de habilidades, capacidades e atitudes dos indivíduos que devem atuando em conjunto. A ideia da agilidade em projetos aceita que há limites no planejamento em um ambiente de incertezas e que, conseqüentemente, o roteiro do projeto deve ser flexível o bastante para que mudanças possam ser introduzidas no projeto, sejam essas oriundas por

meio do cliente ou por meio da equipe de trabalho (PRESMANN, 2006 apud SCHERER, 2012).

A mudança em projetos ágeis de software pode gerar um alto custo para o projeto quando não gerida. O custo da mudança aumenta de forma exponencial em projetos com metodologias tradicionais de desenvolvimento. Pressman (2011, p. 84) apresenta seu pensamento na qual um “projeto ágil bem elaborado ‘achata’ o custo da mudança (...), permitindo que uma equipe de software assimile as alterações (...) sem um impacto significativo nos custos ou no tempo”.

As melhorias e benefícios que a utilização de práticas ágeis em projeto de desenvolvimento de sistemas são notáveis. Os benefícios vêm através do modo diferenciado de realizar o trabalho (metodologia de desenvolvimento), e não pela consequência de realizar este trabalho mais rápido. Além disso, enfatizar que a metodologia ágil torna o processo mais rápido pode trazer prejuízo e prejudicar a produtividade, uma vez que encoraja a equipe a ser menos rigorosa e usar de práticas mal interpretadas para tomar atalhos no desenvolvimento do produto (SHORE, 2008).

2.3 SCRUM

O SCRUM é um *framework* para desenvolvimento de software baseada nos valores e princípios do Manifesto ágil para desenvolvimento de software. Foi idealizado com intuito de gerenciar e resolver problemas de projetos complexos e adaptativos, por exemplo, de caráter inovador. Sua concepção baseia-se no fato de que o desenvolvimento de software é imprevisível, considerando as diversas características técnicas e ambientais que podem ser alteradas durante o desenvolvimento do software, tudo isso deixa estes projetos com um alto nível de incertezas. Assim, o foco do SCRUM está em descobrir uma maneira de produzir, de forma flexível, um produto de software onde às mudanças e as dificuldades inerentes do produto final são constantes. (SCHWABER, 2004).

Os princípios da metodologia são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora atividade estruturas de análise de requisitos, projeto, evolução, desenvolvimento e entrega (PRESSMAN, 2011). O SCRUM enfatiza o uso de padrões da engenharia de software que já provaram sua eficácia em projetos de software com prazo de entrega apertado, requisitos mutáveis e instáveis no seu negócio com o cliente. Conforme Schwaber e Sutherland (2015, p. 3), “o SCRUM deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que você possa melhorá-las”.

O objetivo principal do SCRUM é gerenciar o desenvolvimento de projetos de software grandes, complexos e imprevisíveis, onde não é possível prever todas as mudanças ou o resultado final do produto (EDES, 2005). Beedle et.al (1999) descreve que os padrões ágeis utilizados no SCRUM capacitam uma equipe a trabalhar e entregar produtos competitivos de software com sucesso. Isso ocorre mesmo em um ambiente onde não há possibilidade de eliminar a incerteza, pressupondo a existência do caos em projetos, permitindo que tais características negativas existam. Sua estrutura é apropriada a aceitar as mudanças que há no decorrer do ciclo de vida, fazendo com que os projetos sejam desenvolvidos de forma produtiva e criativa, gerando o mais alto valor agregado possível (SCHWABER, 2004). A estrutura do SCRUM oferece, a partir disso, um conjunto de práticas, processos e ferramentas para tornar possível o desenvolvimento de produtos com essas características visíveis e indesejáveis.

O *framework* do SCRUM é fundamentado no empirismo, ideia de que o conhecimento vem da experiência e da tomada de decisões baseadas no que é conhecido. Além disso, o SCRUM emprega uma abordagem iterativa e incremental a fim de aumentar a previsibilidade no ciclo de vida do projeto e controlar os riscos existentes (SCHWABER; SUTHERLAND, 2015). De acordo com Schwaber e Sutherland (2015, p. 4), “três pilares apoiam a implementação de controle de processo empírico: transparência, inspeção e adaptação”. A transparência denota os aspectos e resultados que devem ser visíveis do processo aos seus responsáveis. Inspeção detecta e direciona as variações nos processos e artefatos. A adaptação consiste nos ajustes necessários nos processos e artefatos do *framework*.

2.4 ITIL

A ITIL consiste num *framework* formado por um conjunto de publicações das melhores práticas para o gerenciamento de serviços de TI. Sua estrutura de trabalho orienta todos os tipos de organizações a compreender o ciclo de concepção, desenvolvimento e entrega de serviços, abordando os processos, funções e outras habilidades necessárias para atingir qualidade em produtos de serviços. A ITIL é atualmente a literatura mais utilizada no mundo para gerenciar serviços, tendo mais de vinte anos de existência, com a versão 3 (três) publicada no ano de 2011. Há várias razões positivas que levam uma organização a adotar o ITIL.

O governo britânico concebeu a ideia da ITIL com o objetivo de otimizar e melhorar os processos do seu departamento de TI. Inicialmente, a *Central Computing and*

Telecommunications Agency (CCTA), atual *Office of Government Commerce* (OGC), foi responsável por criar a estrutura inicial dos processos do *framework*. A OGC é responsável por padronizar os processos e métodos utilizados dentro do governo britânico atualmente. Desde o seu surgimento em 1980, as empresas e outras entidades do governo perceberam a relevância das práticas sugeridas e observaram que poderiam também se beneficiar dos seus processos de TI (MARTINS, 2006). O *framework* da ITIL desenvolvido pelo governo britânico foi bem visto por outros órgãos governamentais e empresas privadas e que, conseqüentemente, passaram a aderi-lo por volta da década de 1990. Por ser uma estrutura padronizada que apresenta as melhores práticas para gerir serviços, sem a intenção de dizer como a organização deve se comportar, auxiliou na popularização da ITIL. Além disso, a popularização também se deu pelo fato de ter sido concebida como um padrão aberto e com enfoque na qualidade, garantida pelas melhores práticas para um gerenciamento de serviços (MAGALHÃES; PINHEIRO, 2007).

A ITIL atua diretamente nos serviços de TI oferecidos pela organização para seus clientes. A OGC (2015) descreve o serviço de TI como sendo um meio que provedores de serviços possuem para entregar valor aos clientes, buscando facilitar os resultados pretendidos sem assumir altos riscos. O *framework* é composto por uma combinação da tecnologia da informação, das pessoas e dos processos. A ITIL gerencia os serviços de TI através da integração dos serviços com as necessidades do cliente, além da gestão das estratégias de negócios da organização (MAGALHÃES; PINHEIRO, 2007). Para isso, os serviços de TI devem buscar alcançar um desempenho desejado pelo cliente. Estabelecer Acordos de Nível de Serviço (ANS) entre a equipe de TI e as demais áreas envolvidas da organização é uma das estratégias que a ITIL utiliza para alcançar isso.

De acordo com a OGC (2015), os livros desenvolvidos na versão três da ITIL alcançaram todo o ciclo de vida do serviço ao dividir o conteúdo em etapas. Tais conteúdos formam um guia para as organizações de serviço de TI que contém práticas e funções para serem adotadas e adaptadas de acordo com a necessidade da organização. A Figura 2 descreve o ciclo de vida dos serviços de TI estabelecido pela ITIL, esse é dividido em cinco etapas que são: *Service Strategy* (Estratégias de Serviços); *Service Design* (Desenho de Serviços); *Service Transition* (Transição de Serviços); *Service Operation* (Operação de Serviços) e *Continual Process Improvement* (Melhoria contínua de serviços).

O ciclo de vida do serviço na ITIL é um modelo de trabalho estruturado de forma a providenciar o que for necessário para a concepção de serviços de TI até a sua retirada. A primeira etapa é a estratégia de serviço, que é o eixo que move todo o ciclo de vida, buscando

alinhar o serviço com o negócio da organização e identificando quais serviços atendem as necessidades e as demandas organizacionais. A etapa de desenho de serviço transforma a estratégia em um plano de trabalho, busca projetar e definir a estrutura do serviço. A transferência de serviço faz a adaptação do serviço desenhado para um ambiente de produção de forma controlada. A operação de serviço produz o valor esperado pelo serviço. Nesta etapa o resultado da estratégia esperada pela organização poderá ser visto pela primeira vez. Por fim, todas as etapas da ITIL são amparadas pela etapa de melhoria contínua de serviços, que é responsável por promover uma estrutura de integração com as demais etapas do ciclo de vida e garantindo a contínua geração de valor para o cliente e negócio com a identificação de melhorias.

A ITIL possui, além das cinco etapas do ciclo de vida, vinte e seis processos que estão relacionados com alguma das cinco etapas e que devem ser executados com o propósito de entregar um serviço de TI. A Figura 3 apresenta os processos, as funções e as respectivas etapas de desenvolvimento do serviço de TI. Os quatro paralelogramas utilizados na fase de operação de serviços representam funções exercidas dentro da organização prestadora de serviços, cujas funções são: central de serviço, gerenciamento técnico, gerenciamento de operações de TI e gerenciamento de aplicativos. Os demais retângulos representam os processos que devem ser executados nas respectivas etapas do ciclo de vida da ITIL.

2.5 EMPRESAS DE SERVIÇO

As organizações trabalham e investem recursos administrativos para vencer um desafio: aumentar seu ganho de produtividade. Para isso, há investimento constante em soluções e inovações da tecnologia da informação para obter melhorias e ganho de desempenho. A simples redução de custos é insuficiente para vencer o desafio, torna-se necessário também buscar por eficiência nos recursos humanos, a simplificação dos processos e a perfeita execução dos mesmos (TACHIZAWA; ANDRADE, 2003). Nakagawa (1993, p. 32) complementa o pensamento da busca por eficiência ao definir que “independentemente de se automatizar e/ou robotizar as atividades de produção, vendas, transportes, distribuição e as de apoio e suporte, haverá sempre oportunidades para a introdução de melhorias de produtos e de processos [...] e eliminação de todas as formas de desperdício”.

As organizações de alto nível e consideradas referências no mercado deixam de focar exclusivamente na redução de custo do serviço para também focar no valor gerado pelos mesmos serviços, investindo, por exemplo, em indicadores de desempenho para monitorar a execução do serviço. De acordo com Magalhães e Pinheiro (2007, p.32):

O Gerenciamento de Serviços de Tecnologia da Informação é o instrumento pelo qual uma área pode iniciar a adoção de uma postura proativa em relação ao atendimento das necessidades da organização, contribuindo para evidenciar a sua participação na geração de valor. O Gerenciamento de Serviços de TI visa alocar adequadamente os recursos disponíveis e gerenciá-los de forma integrada, fazendo com que a qualidade do conjunto seja percebida pelos seus clientes e usuários, evitando-se a ocorrência de problemas na entrega e na operação dos serviços de Tecnologia da Informação.

A área de tecnologia da informação dentro de uma organização deixou de ser um benefício para tornar-se uma área essencial e indispensável, pensada na apresentação de serviços de TI para suportar a organização, com suas metas, seus desafios e seus produtos. A complexidade, entretanto, de aplicar soluções tecnológicas nas organizações aumenta constantemente (CABRAL; THIVES JR, 2005). Dessa forma, existe a necessidade de existir uma equipe de profissionais especialistas no gerenciamento dos serviços de TI prestados pela organização. Esta gestão será a base de conhecimento para prover as soluções dos problemas, amadurecendo o negócio da organização por meio da tecnologia, visando qualidade e proporcionando sistemas capazes de auxiliar a organização nas dificuldades enfrentadas (BERKHOUT et. al., 2000).

As organizações apresentam características e dificuldades ímpares durante o ciclo de vida dos serviços de TI. Controlar a qualidade do serviço prestado é uma dificuldade que precisa ser enfrentada, uma vez que a organização não consegue avaliar com efetividade seus serviços prestados de TI até que sejam executados e postos em produção no cliente. Dessa maneira, põe-se em risco a capacidade do produto de suprir as necessidades de negócio da organização e do cliente. Outra característica que organizações prestadoras de serviço de TI possuem é o fato de dispensarem o gerenciamento de estoques regulares, visto que os serviços de TI não necessitam de espaço físico de armazenamento e são replicáveis. Além disso, a mão de obra torna-se indispensável e insubstituível na maioria dos serviços de TI da organização, não sendo possível substituir a mão de obra por máquinas ou sistemas automatizados para realizar alguma operação ou processo (ANTHONY; GOVINDARAJAN, 2001).

3 PROCEDIMENTOS METODOLÓGICOS

Os procedimentos metodológicos servem para guiar e apresentar o caminho utilizado para desenvolver os objetivos. Identifica-se por quais processos e metodologias o presente trabalho deve se orientar a fim de torná-lo confiável e com respaldo científico. A natureza deste trabalho é classificada como pesquisa bibliográfica aplicada e exploratória,

tornando claro e entendível o problema, além de gerar conhecimento para aplicação prática, dirigidos à solução específica. A partir da utilização da pesquisa, foi possível busca por padrões, ideias ou hipóteses sobre o assunto (GIL 1999, apud BEUREN et al., 2008). O resultado apresentado pelo autor não inclui a realização de testes ou confirmação da proposta de melhoria, de forma que o trabalho explora o conhecimento para apresentar seu resultado.

Para atingir os objetivos propostos foi adotada a pesquisa exploratória e a pesquisa bibliográfica como procedimentos técnicos. Tal aplicação tem o objetivo de alinhar e definir o modelo conceitual e operacional de pesquisa. Para Gil (1999, apud BEUREN et al., 2008) a pesquisa exploratória busca explorar padrões, ideias ou hipóteses sobre determinado assunto. A pesquisa bibliográfica serve para obter conhecimento sobre os temas tratados e utilizados ao longo do trabalho e, com a utilização de materiais já elaborados, dá-se sustentação teórica e científica.

Para auxiliar a pesquisa exploratória e a pesquisa bibliográfica, o trabalho utiliza o procedimento técnico de estudo de caso para identificar e abordar as características e cenários da empresa de serviço estudada. A abordagem serve para complementar e proporcionar a realização do objetivo do trabalho e a conclusão do mesmo. O estudo de caso consiste em aprofundar e detalhar com minúcias e meticulosidade um processo ou caso.

4 RESULTADOS

Neste capítulo apresenta-se a análise e resultado interpretante a utilização dos métodos ágeis de desenvolvimento de software e SCRUM aplicados em ambientes organizacionais de serviço ITIL. O subcapítulo “análise dos resultados” apresenta os conceitos e aplicabilidades para as metodologias ágeis de desenvolvimento de software, ITIL e empresa estudada. O subcapítulo “interpretação da proposta” identifica e apresenta a interação e ligação que há entre as metodologias ágeis ITIL e SCRUM. Por fim, o subcapítulo “apresentar a proposta métodos ágeis do SCRUM para ambientes de serviço ITIL da empresa” consiste na proposta de abordagem com métodos ágeis e SCRUM para o ambiente de serviço ITIL da empresa estudada.

4.1 ANÁLISE

Foram levantados os conceitos e aplicabilidade das metodologias ágeis de desenvolvimento de software, com seus valores e princípios, além dos fatores que motivaram a sua criação. Em seguida, foram apresentados os conceitos do *framework* SCRUM, com suas práticas metodológicas para desenvolvimento ágil de software, sua estrutura e seu ciclo de

vida. Após o trabalho abordou as técnicas para estimativas de software em ambientes ágeis de desenvolvimento do SCRUM com *Planning Poker*. O trabalho também apresentou o *framework* ITIL para gestão de serviços de TI, com seu ciclo de vida, processos e funções. Por fim, informações sobre as empresas de tecnologia que prestam serviços de TI foram levantadas.

O manifesto para desenvolvimento ágil iniciou um movimento para tornar a metodologia de desenvolvimento de sistemas mais dinâmica, simples e rápida. Foram definidos quatro valores e doze princípios que norteiam os *frameworks* e metodologias para criar produtos de software com qualidade em um ambiente de constantes mudanças, colaborativo e com dinamismo nos processos. A valorização dos indivíduos, do software funcionando, da colaboração com o cliente e das respostas às mudanças são recomendadas e defendidas pelo manifesto ágil. O seu incentivo surgiu a partir das dificuldades e problemas apresentados pelos métodos pesados e rigorosos de desenvolvimento, chamados métodos tradicionais. Pressman (2011) complementa o pensamento dizendo que os métodos ágeis vieram com o propósito de sanar fraquezas da engenharia de software convencional, também conhecida pelo título de metodologias tradicionais de software. Tais métodos não conseguiam acompanhar a dinâmica e flexibilidade que ambientes de negócios turbulentos necessitavam para desenvolvimento de software. Assim, para tentar acompanhar as novas necessidades, o manifesto para desenvolvimento ágil de software, com seus valores e princípios, orienta a criação de novas metodologias que receberam o título de metodologias ágeis de software (BECK et al., 2001).

De acordo com Schwaber (2014), o SCRUM é um *framework* baseado nos valores e princípios do manifesto para desenvolvimento ágil de software, que leva em consideração a imprevisibilidade, abstração, incerteza e inovação que há no ciclo de vida do desenvolvimento de software. O objetivo é gerenciar projetos de software grandes, complexos e incertos, onde há algo grau de mudanças e dificuldades em prever o resultado final do produto (EDES, 2005). O SCRUM assume uma abordagem iterativa, adaptativa e incremental para aumentar o valor e a previsibilidade no ciclo de vida do projeto, além de controlar os riscos existentes em ambientes com alto nível de mudanças (SCHWABER; SUTHERLAND, 2015). O SCRUM oferece um ciclo de vida específico para o desenvolvimento de projetos de software, com artefatos, processos, papéis e responsabilidades bem definidos. A proposta idealizada para o SCRUM visa a sua utilização em projetos de desenvolvimento de software, embora também possa ser utilizada em qualquer contexto no qual um grupo de pessoas trabalhem em prol de

um objetivo comum, inclusive em equipes de manutenção, infraestrutura, etc. (SCHWABER, 2004).

O SCRUM possui um conjunto de eventos e artefatos que juntos definem o ciclo de vida de desenvolvimento de software. Essas são: *Product Backlog*, *Sprint Backlog*, *Sprint* e Retrospectiva da *Sprint*. O ciclo de vida começa com a definição das funcionalidades macro para o *Product Backlog*, seguido da *Sprint Planning*, que define os pacotes de trabalho para as *Sprints* que serão executadas durante o intervalo de duas a quatro semanas. Durante o desenvolvimento da *Sprint*, o time SCRUM reúne-se diariamente para acompanhar o andamento do projeto. Conclui-se o ciclo de vida com a retrospectiva e revisão da *Sprint*.

O ciclo de vida do desenvolvimento de software utiliza técnicas de estimativa de software para dar suporte e apoio operacional e gerencial. O objetivo é disponibilizar informações sobre o comportamento do software o mais cedo possível, como tamanho, esforço, duração e equipe necessária, com a intenção de apoiar o planejamento e controle do desenvolvimento (VAZQUEZ et al., 2010). A estimativa é um dos processos mais críticos porque, mesmo sem precisão, as informações estimadas são importantes e úteis. De acordo com Vazquez et al. (2010), as informações existentes em projetos de software são tão particulares e específicas que se tornam estimativas as quais podem ser consideradas como reais. O *Planning Poker* é uma das técnicas para se estimar objetivamente um produto de software, apresenta Grenning (2009). A técnica permite a participação de todos os integrantes na estimativa e cria um resultado democrático do ponto de vista estimado. A dinâmica consiste em realizar um jogo com cartas para demonstrar o valor da estimativa: cada integrante escolhe um valor individualmente e, após apresentarem suas estimativas, um valor médio é decidido e discutido em conjunto.

A ITIL é um *framework* que reúne e apresenta as melhores práticas para gerenciamento de serviços de TI a fim de orientar as organizações a desenvolverem e fornecerem serviços, sendo uma coleção holística de ideias, processos e funções sobre como definir, desenhar, transitar, executar e, finalmente, melhorar continuamente os serviços. Magalhães e Pinheiro (2007) descreve que os serviços de TI estão alinhados às necessidades do cliente, de forma a diminuir o custo e aumentar o desempenho desejado pelo cliente, estabelecendo Acordos de Nível de Serviço (ANS) entre a TI e as demais áreas envolvidas da organização. A ITIL assume o ponto de vista operacional: considera que o valor será entregue para o cliente durante a etapa de operação de serviço e, portanto, reflete os requisitos operacionais definidos da etapa de desenho de serviço (LICHTENBERGER, 2014).

OGC (2009) apresenta cinco etapas bem definidas para o *framework* da ITIL que são: Estratégia de Serviço, Desenho de Serviço, Transição de Serviço, Operação de Serviço e Melhoria Contínua de Serviço. A Estratégia de Serviço identifica quais serviços iram agregar valor para o negócio da organização e para seus clientes. O Desenho de Serviço projeta, desenvolve e levanta o que é necessário para oferecer o serviço imaginado. Transição de Serviço cria e oferece um ambiente de produção adequado para desenvolver o serviço. A Operação de Serviço desenvolve e mantém o serviço em um bom estado de operação. A etapa de Melhoria Contínua de Serviços observa e melhora continuamente o valor gerado pelos serviços e processos.

As organizações investem recursos para aumentar seu ganho de produtividade, querendo melhorar o desempenho e a competitividade através da qualidade da TI e de seus serviços. Aplicar processos que garantam a melhor utilização de recursos é essencial para vencer desafios e, por isso, a gestão de TI deixou de ser uma área de suporte para se tornar uma área estratégica nas organizações. Observando a importância que essa recebeu, o trabalho busca estudar a ITIL e as organizações de serviços de TI para identificar fraquezas ou oportunidades de melhoria em seus processos, contribuindo assim com a gestão de TI. Para manter a privacidade e o anonimato da organização de serviços de TI e da equipe interna de tecnologia e desenvolvimento de serviços de TI estudada, por solicitação das mesmas, serão utilizados nomes fictícios para citá-las ao longo do desenvolvimento da proposta. A empresa que promove e oferece os diversos serviços de TI será referenciada como “Empresa X” e a equipe como “Equipe Y”.

A Empresa X é uma companhia privada que trabalha em prol da competitividade e em defesa de determinado setor econômico. Suas entidades atuam com sinergia para favorecer os negócios, a qualidade de vida, educação e promoção da inovação ao setor. Por meio de sua atuação e influência política, a Empresa X defende os interesses deste setor junto ao poder público, além de oferecer produtos e serviços para garantir o acesso a novas tecnologias, sistemas de gestão, conhecimento, formação profissional e práticas de responsabilidade social e ambiental, além de congrega vários sindicatos do setor. A Empresa X é mantida por inúmeras outras empresas do setor e gerida com a mesma excelência e critérios aplicados às grandes companhias privadas.

A Equipe Y é uma equipe interna da Empresa X que trata exclusivamente das tarefas de natureza problema do ANO, sendo constituída, atualmente, por dois desenvolvedores de sistemas, por um analista de sistemas e por um auditor técnico. Esta equipe está incluída no contexto da ITIL para gerir seus processos e serviços de TI. A principal

função da equipe perante a Empresa X é identificar os problemas, realizar a análise das soluções, desenvolver a correção do problema e auditá-lo para garantir que o problema foi identificado, analisado, corrigido e solucionado. Para identificar os possíveis problemas no sistema, a equipe levanta e analisa as tarefas de duas naturezas do ANO e ANS: incidente e solicitação de serviço. A análise consiste em observar padrões incorretos nas tarefas identificadas anteriormente e, a partir disso, inspecionar o sistema em busca do possível problema. Em seguida, caso o mesmo ocorra, prioriza-se o problema e propõem-se uma solução que será desenvolvida no sistema. A identificação, análise e priorização dos problemas são feitos em uma reunião semanal com a presença de toda a equipe, que também discute situações específicas como o planejamento semanal do desenvolvimento e da auditoria, além de gerar um relatório sobre as situações.

O cenário tratado, estudado e analisado neste trabalho mostra que a Empresa X definiu e adotou um ANS e ANO para melhorar a eficiência nos processos com as partes envolvidas na composição e fornecimento de serviços de TI. A adoção é utilizada para reger os contratos de prestação de serviço de TI para os clientes e para orientar e organizar as tratativas com equipes internas, utilizando as boas práticas descritas pela ITIL. A combinação dos acordos deixa claro o que é necessário e qual a responsabilidade de cada parte envolvida com o serviço de TI. O principal objetivo ao definir um ANS e um ANO é manter a qualidade, o bom funcionamento e a sustentabilidade dos processos organizacionais para corresponderem às expectativas do cliente.

Tabela 2- Relação ANS e ANO, parametrizado por natureza da tarefa.

Natureza	Primeira Resposta	Vencimento
Acordo de Nível de Serviço		
Incidente	03 horas	16 horas
Solicitação de Serviço	03 horas	40 horas
Acordo de Nível Operacional		
Incidente	01 hora	04 horas
Solicitação de Serviço	02 horas	16 horas
Problema	08 horas	40 horas

Fonte: Autor (2015)

A Tabela 2 extrai parte do ANS e ANO firmados pela Empresa X, com o tempo máximo de primeira resposta e vencimento das tarefas assumidas e tratadas pela organização. O ANS define o limite de tempo para efetuar uma primeira resposta e o vencimento da solução das tarefas de incidentes e solicitações de serviço oriundas do cliente. O ANO

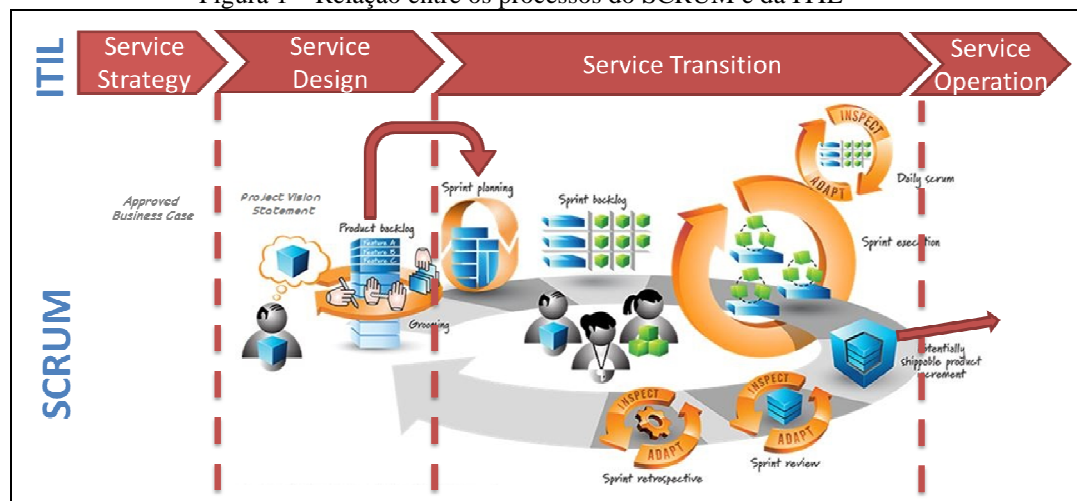
padroniza o limite para tratar as tarefas de natureza incidente, solicitação de serviço e problema, ambas oriundas das equipes internas operacionais envolvidas com o serviço de TI.

4.2 INTERPRETAÇÃO DA PROPOSTA

De acordo com Lichtenberger (2014) e complementando com Magalhães e Pinheiro (2007), Schwaber e Sutherland (2015) e Edes (2005), não há incompatibilidade entre o *framework* da ITIL e do SCRUM, de forma que não se encontra contradições entre ambos. Entre ambos os *frameworks* há somente abordagens e objetivos diferentes para a organização. O desafio está em criar uma interação entre esses. Lichtenberger (2014) defende a ideia de que a ITIL e o SCRUM se encaixam perfeitamente em um alto nível de abstração entre suas etapas. Do ponto de vista ITIL, o SCRUM começa com a etapa de desenho de serviços e termina com a entrega de um produto ou serviço.

De acordo com Lichtenberger (2014), os *frameworks* SCRUM e ITIL possuem objetivos e estruturas diferentes para uma organização. Juntos podem agregar e gerar valor em áreas de atuação ao relacionar suas etapas. O Desenho de Serviço da ITIL é feito após a estratégia de negócio para o serviço ser aprovada na etapa anterior de Estratégia de Serviço, onde os requisitos são analisados, reunidos e priorizados. A ITIL, sendo um conjunto de boas práticas, descreve o que o processo precisa garantir, contudo não apresenta como atingir tais objetivos ou o que utilizar para tal. Utilizando os conhecimentos apresentado, o autor apresenta um diagrama que reuni as cinco etapas da ITIL e o ciclo de vida do SCRUM. A Figura 1 ilustra as etapas do ciclo de vida da ITIL com relação abstrata de alto nível com os processos e artefados o SCRUM.

Figura 1 – Relação entre os processos do SCRUM e da ITIL



Fonte: Lichtenberger (2014)

A Figura 1 mostra uma união possível e compatível entre os processos da ITIL e do SCRUM, demonstrando graficamente como ambos estão relacionados. Utilizando uma representação temporal para os eventos, o lado esquerdo marca o início do ciclo de vida e o lado direito marca o final do mesmo. Na parte superior da figura é retratado os processos existentes da ITIL, e na parte inferior da figura é retratado o ciclo de vida do SCRUM, com seus processos e artefados. De acordo com Lichtenberger (2014), as histórias de usuário podem ser utilizadas na identificação da estratégia do serviço, o resultado equivale-se em alto nível de abstração ao *product backlog* priorizado para o SCRUM. Além disso, a etapa de Transação de Serviço da ITIL é feita, em analogia ao SCRUM, através de *Sprints*. A *Sprint* começa com uma reunião de planejamento. Sendo o SCRUM um processo iterativo e incremental e a ITIL servindo como um guia de boas práticas, não há casos que impeçam que a etapa de transação de serviços seja realizado através da execução de *Sprints*.

4.3 APRESENTAR A PROPOSTA MÉTODOS ÁGEIS DO SCRUM PARA AMBIENTE DE SERVIÇO ITIL DA EMPRESA

O estudo realizado neste trabalho consiste em propor melhorias aos processos executados por uma equipe de tecnologia e desenvolvimento de software que utiliza o *framework* ITIL para gerenciar o ciclo de vida dos seus serviços de TI. A melhoria proposta está alinhada com os valores e princípios do manifesto ágil para desenvolvimento de software, tendo em vista os benefícios e vantagens que esse proporciona. Após um estudo criterioso da literatura da ITIL, do manifesto ágil e da metodologia ágil SCRUM, o trabalho aborda os processos, características e estruturas gerais das empresas prestadoras de serviços de TI. Em seguida, apresentou-se o respaldo científico da harmonia e complemento em alto nível de abstração entre ambos os *frameworks* ITIL e SCRUM. Isso mostra que os objetivos e aplicabilidade são diferentes, não havendo conflito entre si, mas complementando o valor gerado por ambos.

Para propor uma melhoria nos processos da Equipe Y, um cenário de estudo específico com as informações e os processos metodológicos que a equipe utiliza foi identificado e estudado. Posteriormente, uma análise foi feita sobre as horas trabalhadas em tarefas de natureza problema do ANO de dois colaboradores com o cargo de analista e desenvolvedor de sistemas da Empresa X e que atuam na Equipe Y. O objetivo desta análise é comparar a quantidade total de horas necessárias para finalização de todas as tarefas assumidas e tratadas pelos colaboradores com o prazo de vencimento do ANO. O resultado

servirá como base e como justificativa para desenvolver uma proposta de melhoria aos processos definidas pela Empresa X. Tratando-se das abordagens ágeis e benefícios apresentados aos projetos de software e com a intenção de introduzir técnicas e ferramentas ágeis à equipe Y, havendo respaldo e base no resultado da análise, será proposta a aplicação da técnica de estimativa ágil com *Planning Poker*.

O trabalho irá mapear e descobrir informações sobre a quantidade total de horas necessárias para concluir as tarefas de natureza problema acordadas no ANO e tratadas pela Equipe Y. Posteriormente, uma comparação será feita entre as informações mapeadas das tarefas tratadas e as informações de vencimento acordada no ANO. A intenção é descobrir se o valor acordado para vencimento das tarefas realmente condiz com a quantidade total de horas necessárias para concluir tais tarefas, seguindo uma amostragem real de dados disponibilizada pela Empresa X. Após será feita uma análise e avaliação das comparações com o intuito de justificar a necessidade de propor mudanças aos processos da Equipe Y. Por fim, havendo respaldo, uma proposta de melhoria é apresentada.

A tabela 3 apresenta e compara informações sobre as tarefas de natureza problema com a quantidade total de tarefas, a carga horária total necessária para concluir todas as tarefas, o tempo médio de tratativa, a tarefa com a maior carga horária para vencimento e a tarefa com a menor carga horária para vencimento. Para formular a tabela, utilizaram-se todas as tarefas de natureza problema do ANO tratadas pela Equipe Y desde a sua criação. As informações dos tempos necessários para concluir as vinte e duas tarefas tratadas pela Equipe Y serão utilizadas para propor a melhoria objetivada pelo trabalho.

Tabela 3 - Relação das tarefas de natureza problema do ANO

Natureza	Total de Tarefas	Carga Horária Total	Carga Horária Média	Maior Carga Horária	Menor Carga Horária
Problema	22	293h55	13h21	80h29	00h08

Fonte: Autor (2015).

Observa-se que o tempo médio necessário para tratar as vinte e duas tarefas de natureza problema foi de 13h21 (treze horas e vinte e um minutos), o que corresponde a 33,37% do valor acordado para vencimento no ANO. Houve uma tarefa que necessitou de 80h29 (oitenta horas e vinte e nove minutos) de trabalho para ser concluída, essa corresponde a 201,20% do tempo acordado no ANO. Além disso, observa-se que uma tarefa de menor

carga horária de vencimento precisou de oito minutos para ser concluída, equivalendo a 0,33% do tempo acordado no ANO.

Após analisar as informações coletadas sobre as tarefas de natureza problemas do ANO tratados pela Equipe Y, o trabalho buscou atacar as dificuldades que a equipe possui em realizar estimativas das tarefas, de modo a querer melhorar a eficiência e eficácia. Conforme Silveira (2000), a medição de produtos de software é considerada uma das atividades mais importantes para a melhoria de processos e produtos de software. Bassi (2010) complementa o pensamento sobre estimativas descrevendo que quanto maior o nível de detalhamento da atividade estimada, maior será a probabilidade de acerto na estimativa. Após realizar o processo de estimar, as informações coletadas e armazenadas podem servir para tomada de decisão, além de aumentar a credibilidade e tornar os planos mais realistas. Conforme Pressman (2011) descreve, as estimativas devem ser feitas no início do projeto para gerarem informações e subsídios aos processos de planejamento e controle do ciclo e vida. Obter a estimativa no final do projeto não é uma prática válida, visto que não gera resultados significativos para o ciclo de vida do projeto.

A utilização de uma abordagem ágil para propor a melhoria nos processos da Empresa X aplicados na Equipe Y fundamenta-se no fato de que tal abordagem apresenta inúmeros benefícios para o ciclo de vida do desenvolvimento de software. A Tabela 4 compara os projetos com a abordagem utilizada no desenvolvimento de software, mostrando o percentual de projetos classificados como bem sucedidos no desenvolvimento de software, postos em risco e os cancelados, Conforme o *The Standish Group* (2013), o percentual de projetos que obtiveram sucesso foi maior na abordagem ágil e os projetos arriscados e cancelados foram menores na abordagem ágil.

Tabela 4 - Comparativo de desempenho entre abordagem tradicional e ágil.

Classificação / Abordagem	Tradicional	Ágil
Bem sucedido	14%	42%
Posto em risco	57%	49%
Cancelado	29%	9%

Fonte: *The Standish Group* (2013).

A proposta de melhoria sugerida para a Empresa X consiste em introduzir a técnica de estimativa ágil *Planning Poker* nos processos da Equipe Y criando, assim, a cultura de estimativa, a fim de identificar antecipadamente o tempo necessário para vencimento de

determinada tarefa de natureza problema do ANO. As tarefas devem assumir que o tempo de vencimento será o valor estimado, e não o tempo de quarenta horas acordado no ANO para as tarefas de natureza problema. Semanalmente a equipe se reúne com o propósito de identificar problemas no sistema a partir dos incidentes e solicitações de serviço vindas do ANO e ANS, entre outras atividades realizadas na reunião. A estimativa deve ser feita no momento que se identifica um problema no sistema e abre-se uma tarefa para mapear a causa, para posterior desenvolvimento da solução. Como proposta, a equipe irá aplicar a técnica de *Planning Poker* nesta reunião semanal e cadastrar a tarefa com o tempo estimado para vencimento, se houver, ao invés do tempo acordado pelo ANO. Segundo as regras para a prática de estimativas com *Planning Poker*, toda a equipe deve participar da estimativa.

A implantação das melhorias propostas para a Equipe Y busca aumentar a qualidade do nível de gestão e controle das tarefas de natureza problema. Busca-se seguir o pensamento de Vazquez et al. (2010) quando apresenta a estimativa de software como sendo um processo de busca de benefícios no planejamento e controle do ciclo de vida dos projetos. Florac et al. (1997) complementa descrevendo que a estimativa pode fornecer informações estratégicas para apoiar o gerenciamento de processos, além de tornar o projeto de software planejável e controlável. Sendo assim, desta forma, a estimativa torna-se uma das atividades mais críticas e fundamentais para o ciclo de vida do projeto de software. A intenção de aplicar um processo de estimativa ágil na Equipe Y não elimina a necessidade de existir um ANO e ANS. Não havendo a possibilidade de realizar a estimativa de software por eventuais contratemplos, a equipe deve assumir devidamente o tempo de vencimento acordado.

O trabalho buscou estudar a Equipe Y da Empresa X para identificar um cenário que utilize o ITIL e que as abordagens ágeis para desenvolvimento de software pudessem ser inseridas para, assim, melhorar a qualidade dos serviços de TI prestados. Ao mapear as aplicabilidades dos *frameworks* ITIL e SCRUM, concluiu-se que ambos possuem uma estrutura e objetivos diferentes para a organização, não havendo incompatibilidades e contradições entre ambos (LICHTENBERGER, 2014). O objetivo do trabalho ao estudar um cenário da Equipe Y é apresentar uma proposta de melhoria para torná-la mais eficiente e eficaz nos seus processos, possibilitando que haja melhorias no planejamento e controle.

5 CONSIDERAÇÕES FINAIS

Este trabalho propôs, por meio do estudo da arte, pesquisar e desenvolver uma proposta de melhoria para ser aplicada na Equipe Y da Empresa X, num contexto ITIL de serviços de TI e utilizando as abordagens ágeis para desenvolvimento de software. Analisa-se um cenário real específico vivenciado pela equipe e, após estabelecer um relacionamento entre os *frameworks* ITIL e SCRUM, foi possível criar uma fundamentação teórica e científica para propor melhorias aos processos da empresa. Com isso, haja vista a proposta, o objetivo deste trabalho foi atingido ao realizar um estudo de caso para as abordagens ágeis de software num ambiente ITIL e, se viável, propor uma melhoria para a Empresa X.

Realizou-se uma pesquisa bibliográfica sobre os temas abordados e estudados neste trabalho, sendo as abordagens ágeis para desenvolvimento de software, as metodologias ágeis, o *framework* SCRUM, o ITIL e as empresas de desenvolvimento de software e serviços de TI. A pesquisa realizada serviu com base para o mapeamento de ambos os *frameworks* ITIL e SCRUM a fim de torná-los compatíveis e, através da união deles, propor uma melhoria para a Empresa X. A Equipe Y e seus processos e práticas foram estudados com o propósito posterior de identificar fraquezas ou oportunidades de melhoria, nas quais as abordagens ágeis conseguissem ser aplicadas para o benefício da empresa. Percebe-se, após isso, que há inúmeros cenários existentes numa organização na qual as abordagens ágeis poderiam, após estudo de viabilidade, ser inseridas de maneira a proporcionar melhorias na qualidade dos processos e serviços.

Para desenvolver uma proposta utilizando métodos ágeis aplicados em processos ITIL da Empresa X, que é o objetivo deste trabalho, foi necessário cumprir alguns objetivos. Levantou-se os conceitos e aplicabilidades para as metodologias ágeis de desenvolvimento de software, ITIL e empresa estudada; isso foi possível através do levantamento bibliográfico necessário dos conceitos utilizados, a análise dos resultados e a aplicabilidade para os temas. Em seguida, identificou-se a interação entre as metodologias ágeis ITIL e SCRUM, com a interpretação da proposta de Lichtenberger (2014). Realizou-se um mapeamento entre ambos os *frameworks* e concluiu que não há incompatibilidades entre eles, visto que as suas abordagens são diferentes em uma organização. Para propor uma abordagem voltada a ambientes de serviços ITIL da empresa estudada com métodos ágeis e SCRUM o trabalho reuniu o conhecimento conceitual adquirido, as informações da empresa estudada e, após uma análise do cenário, este trabalho propôs uma melhoria aos processos do mesmo.

Este trabalho contribuiu para a ampliação dos conhecimentos e aplicabilidades dos conceitos nele apresentados. Pessoalmente, destaca-se o crescimento intelectual em

métodos ágeis e sua abordagem organizacional, além do *framework* ITIL. A empresa escolhida para ser alvo dos estudos obteve a proposta de melhoria – resultado deste trabalho – e essa será útil para aumentar a qualidade dos processos internos. A empresa poderá ainda utilizar os conhecimentos em novos cenários a fim de propor novas melhorias baseadas neste trabalho. Tal aperfeiçoamento organizacional é válido e necessário no ambiente dinâmico de negócios. Pressman (2011) apresenta que as organizações precisam ser ágeis o suficiente para dar uma resposta em um ambiente de fluidos negócios. Conclui-se que observar os processos internos da organização em busca de melhorias é uma ação que proporciona mudanças e isso, que por sua vez, faz parte dos valores do manifesto ágil.

Houve conquistas ao longo do desenvolvimento deste trabalho e resultados significativos foram atingidos, contudo houve, também, a superação de dificuldades enfrentadas e limitações que precisam ser contornadas. Antes de prosseguir com o desenvolvimento deste trabalho, houve a necessidade de abordar a empresa estudada a fim de apresentar a proposta e solicitar autorização para utilizar os processos e dados. A dificuldade enfrentada estava no fato de apresentar uma proposta abstrata e uniforme neste momento do trabalho, ou seja, não se sabia com exatidão qual seria a metodologia científica utilizada e qual seria o resultado final do trabalho. A empresa, contudo, prontificou-se e disponibilizou as informações necessárias para a conclusão do mesmo.

REFERÊNCIAS

ANTHONY, Robert N.; GOVINDARAJAN, Vijay. **Sistemas De Controle Gerencial**. São Paulo: Atlas, 2001. 1019 p

BASSI, F. D. L., **Experiências com desenvolvimento Ágil, São Paulo**, <http://www.teses.usp.br/Dissertacao_Metodos_Ageis_Dairton_Bassi.pdf>. Acesso em: 05 out. 2016.

BECK, Kent. et al. **Manifesto for Agile Software Development**. 2001. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em: 23 jul. 2015.

BEEDLE, M.; DEVOS, M.; SHARON, Y.; SCHWABER, K.; SUTHERLAND, J. **SCRUM: A Extension Pattern Language for Hyperproductive Software Development**. In: Harrison, N.; Foote, B.; Rohnert, H. *Pattern Languages of Program Design 4*. Addison-Wesley, 1999.

BEUREN, Ilse Maria et al. **Como elaborar trabalhos monográficos**. 3. ed. São Paulo: Atlas, 2008.

BERKHOUT, Michiel, et al. **Service Support: ITIL – The key to Managing IT Services**. The Stationary Office for OGC. Noruega, 2000. CD-ROM. ISBN 0113300158.

BROOKS, Frederick P. 1995. *The Mythical Man-Month: Essays on Software Engineering, 20^o Anniversary Edition*. Boston: Addison-Wesley Professional.

CABRAL, R. B.; THIVES JR, Juarez Jonas. **Do núcleo de informática à tecnologia da informação: a governança de TI em um estudo de caso**. In: XVI Encontro Nacional dos Cursos de Graduação em Administração. Belo Horizonte, 2005.

EDES, G. C. et al. **Agile Project Management: Um Novo Enfoque para o Gerenciamento de Projetos de Desenvolvimento de Sistemas de Tecnologia de Informação**. São Paulo: Universidade de São Paulo –USP. 6f, 2005.

FLORAC, William A.; et al. "*Practical software measurement: measuring for process management and improvement*" [S.l.], 1997. Disponível em: <<http://www.sei.cmu.edu/pub/documents/97.reports/pdf/97hb003.pdf>>. Acesso em: 20 ago. 2016.

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**.5. ed. São Paulo: Atlas, 1999.

GRENNING, James; **Planning Poker Party: Planning Poker Party**. 2009. Disponível em: <<http://www.renaissancesoftware.net/blog/archives/36>>. Acesso em: 15 jul. 2015.

HIGHSMITH, J. *Agile Software Development Ecosystems*. Boston: Addison Wesley, 2002.

HIGHSMITH, J., *Agile Project Management, Creating innovative products*, AddisonWesley, 2004.

LICHTENBERGER, Alex. **Integrating Agile and ITSM**. 2014. Disponível em: <<http://blog.itil.org/2014/07/allgemein/integrating-agile-and-itsm/>>. Acesso em: 08 ago. 2015.

MAGALHÃES, Ivan L.; PINHEIRO, Walfrido B. **Gerenciamento de Serviços de TI na Prática: Uma abordagem com base na ITIL**. 1a Ed. São Paulo: Novatec, 2007.

MANSUR, Ricardo, **Governança de Tecnologia – ITIL**, 2005. Disponível em: <<http://pt.scribd.com/doc/3101205/itil>>. Acesso em: 01 jan. 2011.

MARÇAL A. S., FREITAS B. C. F. et al. **Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI**. Disponível em: <<http://www.dimensaotech.com/2010/09/estendendo-o-scrum-segundo-as-regras-de-processo-de-gerenciamento-de-projetos-do-cmmi>>. Acesso em: 22 jul. 2011.

MARTINS, Márcia Missias Gomes. **Gerenciamento de Serviços de TI: uma proposta de integração de processos de melhoria e gestão de serviços**. Dissertação de Mestrado. Universidade de Brasília. Distrito Federal, 2006 Disponível em: <<http://www.ene.unb.br/public/mestrado/marciamissias.pdf>>. Acesso em: 17 jul. 2016.

MOUNTAIN GOAT SOFTWARE. **Introduction to Scrum**. Disponível em: <<https://www.mountaingoatsoftware.com/agile/scrum/team>>. Acesso em: 10 set. 2015.

NAKAGAWA, Masayuki. **Introdução a controladoria**. São Paulo: Atlas, 1993.

OGC. **ITIL v3 Service Strategies**. Inglaterra: TSO 2007. Vol1.

OGC. **ITIL v3 Service Design**. Inglaterra: TSO 2007. Vol2.

OGC. **ITIL v3 Service Transition**. Inglaterra: TSO 2007. Vol3.

OGC. **ITIL v3 Service Operation**. Inglaterra: TSO 2007. Vol4.

OGC. **ITIL v3 Service Continual Service Improvement**. Inglaterra: TSO 2007. Vol5.

PRESSMAN, Roger S. **Engenharia de Software**. 6. ed., São Paulo: McGraw-Hill, 2006.

PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 7. ed., São Paulo: McGraw-Hill, 2011.

SCHERER, Calvin Bersch. **Gestão de riscos aplicada às metodologias ágeis: Integrando PMBOK e SCRUM**. Florianópolis, 2012 f. Trabalho de Conclusão de Curso de Tecnologia – Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Faculdade de Tecnologia do SENAI, Florianópolis, 2012.

SCHWABER, Ken; SUTHERLAND, Jeff. **The SCRUM Guide of SCRUM.org**. Disponível em: <<http://www.SCRUMguides.org/docs/SCRUMguide/v1/SCRUM-Guide-Portuguese-BR.pdf>>. Acesso em: 30 ago. 2015.

SCHWABER, Ken. **Agile Project Management with Scrum**: Microsoft Professional. Redmond, Washington: Microsoft Press, 2004. 192 p.

SHORE, J. **A arte do desenvolvimento ágil**. Alta Books, 2008.

SILVEIRA, M. Estimando Projetos de TI: Arte ou Ciência, 2000. Disponível em: <<http://www.bfpug.com.br/Artigos/EstimandoProjetosDeTI.htm>>. Acesso em: 03 out. 2015.

TACHIZAWA, Takeshy; ANDRADE, Rui Otávio Bernardes. **Tecnologias da Informação Aplicadas às Instituições de Ensino e às Universidades Corporativas**. São Paulo: Atlas, 2003.

THE STANDISH GROUP. **Chaos Manifesto 2013**: Think Big act Small. 2013. Disponível em: <<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>>. Acesso em: 29 mai. 2015

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de Pontos por Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 9a. Edição. Érica. São Paulo: 2010.