



**UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP**  
**ESCOLA DE MINAS - EM**  
**DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO – DEPRO**



**RENATA AMBRÓSIO DE CARVALHO**

**MAPEAMENTO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARES  
DE UM SETOR DE TECNOLOGIA DA INFORMAÇÃO DE UMA  
INSTITUIÇÃO PÚBLICA DE ENSINO SUPERIOR E REDESENHO DO  
PROCESSO UTILIZANDO A METODOLOGIA *SCRUM***

**OURO PRETO**

**2019**

RENATA AMBRÓSIO DE CARVALHO

renata.ambrosio@aluno.ufop.edu.br

**MAPEAMENTO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARES  
DE UM SETOR DE TECNOLOGIA DA INFORMAÇÃO DE UMA  
INSTITUIÇÃO PÚBLICA DE ENSINO SUPERIOR E REDESENHO DO  
PROCESSO UTILIZANDO A METODOLOGIA *SCRUM***

Projeto de monografia do curso de Engenharia de  
Produção da Universidade Federal de Ouro Preto.

Orientador: André Luís Silva

OURO PRETO

2019

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C331m Carvalho, Renata Ambrosio de .

Mapeamento do processo de desenvolvimento de softwares de um setor de tecnologia da informação de uma instituição pública de ensino superior e redesenho do processo utilizando a metodologia SCRUM. [manuscrito] / Renata Ambrosio de Carvalho. - 2019.

70 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. André Luis Silva.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Escola de Minas.

1. Software. 2. Scrum (Desenvolvimento de software) . 3. Desenvolvimento de softwares - Processos. 4. Software de Mapeamento. I. Silva, André Luis. II. Universidade Federal de Ouro Preto. III. Título.

CDU 658.5

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB: 1716



MINISTÉRIO DA EDUCAÇÃO  
Universidade Federal de Ouro Preto  
Escola de Minas  
Departamento Engenharia de Produção, Administração e Economia

## ATA DE DEFESA

Aos 18 dias do mês de setembro de 2019, às 19hs, através de videoconferência foi realizada a defesa de monografia pela discente **Renata Ambrósio de Carvalho** sendo a banca examinadora constituída por: Prof. André Luis Silva, Prof. Helton Cristiano Gomes, Antônia Auxiliadora Ferreira Gomes e Ricardo Morato Fiuza Guimarães. A discente apresentou a monografia intitulada: **"Mapeamento do processo de desenvolvimento de softwares de um setor de tecnologia da informação de uma instituição pública de ensino superior e redesenho do processo utilizando a metodologia Scrum"**. A banca examinadora deliberou, por unanimidade, pela aprovação da discente, concedendo-lhe o prazo de 30 dias para incorporação no texto final das alterações sugeridas. Na forma regulamentar foi lavrada a presente ata que é assinada pelos membros da banca examinadora e pela discente.

Ouro Preto, 18 de Setembro de 2019.

Prof. André Luis Silva  
Professor Orientador/ Presidente

Prof. Helton Cristiano Gomes  
Convidado

Antônia Auxiliadora Ferreira Gomes  
Convidada

Ricardo Morato Fiuza Guimarães  
Convidado

Renata Ambrósio de Carvalho  
Discente

## **RESUMO**

Este trabalho refere-se a um estudo de caso o qual detalha a aplicação da metodologia *Scrum* em uma organização que produz softwares para uma instituição de ensino superior. Para tal aplicação, foi realizado um mapeamento do processo de desenvolvimento de softwares que a organização utilizava. Esta desenvolvia seus softwares utilizando a metodologia tradicional. Depois deste mapeamento foi realizado um redesenho do processo alterando a metodologia, de tradicional, para ágil. O objetivo deste trabalho é implantar a metodologia Scrum para desenvolver softwares no setor de TI de uma universidade pública do estado de Minas Gerais.

**PALAVRAS CHAVES:** *Software, Scrum, Processos, Mapeamento.*

## SUMÁRIO

1. Introdução.....	8
1.1 Contexto .....	8
1.2 Questão problema .....	10
1.3 Objetivos.....	11
1.3.1 Objetivo Geral.....	11
1.3.2 Objetivo Específicos .....	11
1.4 Metodologia .....	12
1.5 Justificativas .....	12
1.6 Estrutura do texto.....	13
2. Revisão de Literatura .....	14
3. Metodologia.....	18
4. Resultados.....	20
4.1 Mapeamento de processos .....	20
4.2 Processo de software SISP .....	24
4.3 Redesenho do processo.....	25
4.4 Implantação do processo.....	29
4.5 Melhoria no processo.....	30
5. Conclusão .....	31
6. Referências .....	35

## **Índice de Figuras**

Figura 1 - Organograma da organização .....	18
Figura 2 - Fase 1 do processo do SISP.....	24
Figura 3 - Processo geral.....	25
Figura 4 - Fase 1 do processo .....	26
Figura 5 - Parte 1 do Termo de abertura do projeto.....	28
Figura 6 – Parte 2 do Termo de abertura do projeto .....	29

## **Índice de Tabelas**

Tabela 1 - Método tradicional X Método ágil .....	15
Tabela 2 - Número de funcionários .....	19
Tabela 3 - Detalhe da atividade 1 .....	27

## **Índice de Gráficos**

Gráfico 1 - Resposta sobre as dificuldades do planejamento .....	21
Gráfico 2 - Resposta sobre a importância do protótipo .....	22

## 1. Introdução

### 1.1 Contexto

Um dos principais desafios na área de gerenciamento de projetos de desenvolvimento de software é a dificuldade de compreender os reais problemas, levantar os requisitos do software e, conseqüentemente, criar soluções que atendam efetivamente as necessidades dos clientes.

*Não são raras as situações em projetos de engenharia de software onde, ao final do projeto, percebe-se que o que foi desenvolvido não é exatamente o que o solicitante estava esperando (da Silva & Lovato, 2015).*

Esse fato acontece, na maioria das vezes, pela grande distância que pode existir entre o cliente e o desenvolvedor do software, e a dificuldade em identificar quais os objetivos e requisitos do sistema nas etapas iniciais do projeto.

*Métodos tradicionais de gerenciamento de projetos têm estruturas documentais rígidas que não permitem adaptações ou desvios de seus escopos originais, tornando-os lentos em seu tempo de execução e mais custosos para as empresas e tal rigidez pode ocasionar em um produto muito divergente do necessário no momento de sua entrega (Date et al., 2016).*

Contudo, em um mundo altamente tecnológico, dinâmico e integrado, a realização de projetos através desta metodologia está perdendo o seu lugar nas indústrias de softwares. Por isso, cada vez mais se populariza na área de desenvolvimento de softwares a utilização de metodologias ágeis para gerenciamento de projetos. A maior motivação nas metodologias ágeis se dá pela grande valorização das interações entre a equipe e o cliente, bem como o sucinto planejamento do projeto.

São muitos os métodos ágeis de gestão de projetos presentes na literatura. Entre os mais estudados estão, Scrum (Sutherland, 2016), Extreme Programming (XP) (Teles, M. V., 2014) e Adaptive Software Development (ASD) (Alnoukari, Alzoabi, Hanna, 2008).

Constâncio e Souza Neto (2016) apontam que o setor público precisa de uma reforma em sua gestão de projetos para que os produtos criados por ele e para ele tenha maior responsividade e valor agregado. Isso resultará em menos retrabalho nos projetos e aumentará o engajamento das equipes e o desempenho na entrega.

O Sistema de Administração de Recursos de Tecnologia da Informação (SISP) foi criado para auxiliar na melhoria da prestação dos serviços públicos. O SISP foi diretamente ligado com a Estratégia Geral de Tecnologia da Informação (EGTI). Com isso, foram criados alguns objetivos e metas. Algumas destas que estão descritas no EGTI biênio 2011-2012 são: Objetivo 4 – “Aperfeiçoar a gestão de processo de TI nos órgãos do SISP”; Meta 8 – “Adotar um processo formal de gestão de processo de TI nos órgãos do SISP”. Além disso, no planejamento estratégico 2011-2014 do Ministério do Planejamento, Desenvolvimento e Gestão (SLTI) está descrito como objetivo: “Melhorar processos de gestão e governança de TIC dos órgãos integrantes do SISP”.

Para o cumprimento desses objetivos e metas, foi criada a Metodologia de Gerenciamento de Projetos do SISP (MGP-SISP). Este consiste em um conjunto de boas práticas em gerenciamento de projetos para os órgãos integrantes do SISP. Os integrantes do SISP são os núcleos, gerências, coordenações, entre outros, de TI de todos os órgãos federais. O setor apresentado neste trabalho é um integrante do SISP.

No MGP-SISP contém um processo padrão de gerenciamento de projetos de desenvolvimento de softwares. Este deve ser implementado pelos setores de TI para padronizarem a forma de gerenciar os projetos e alinhar os processos de trabalho criando a documentação adequada. O processo é padrão mas deve ser desenhado para cada setor alinhado a sua realidade. Além disso, a metodologia de desenvolvimento de softwares, a qual é dividida em tradicional e ágil, fica a critério do setor escolher.

O estudo aqui apresentado foi realizado em um setor de Tecnologia da Informação (TI) de uma universidade pública do estado de Minas Gerais. A existência dos processos desenhados e/ou escritos é fundamental no momento do desenvolvimento de softwares. São esses processos que descreve como se dá um fluxo de trabalho, ponto crucial para o levantamento dos requisitos do software a ser desenvolvido.

A falta do mapeamento ou conhecimento dos processos de trabalho dos demais setores é um dos principais desafios do setor de TI dentro de uma instituição pública.

Principalmente quando para o desenvolvimento de softwares é utilizado o método tradicional, o qual necessita do levantamento de todos os requisitos e informações no início do projeto. O setor estudado utilizava até o ano de 2018 o método tradicional.

Com o surgimento da necessidade em melhorar a qualidade dos seus projetos e a criação do MGP-SISP, o setor de TI desta universidade decidiu alterar o método de desenvolvimento de softwares. O setor que antes utilizava o método tradicional, implementou e passou a usar a metodologia *Scrum*, a qual se caracteriza como um método ágil.

O uso desta metodologia permitiu padronizar o processo de desenvolvimento de softwares, alinhando os processos de trabalho e criando a documentação adequada. Fator que possibilitou melhorias na governança, refletindo no aumento da eficácia, eficiência, efetividade, produtividade, menor dependência dos indivíduos, aumentando a retenção do conhecimento na organização, além de melhorar a qualidade de todos os softwares desenvolvidos, aumentando assim, a satisfação da comunidade com o setor.

O contexto do projeto narrado nesta monografia se dá desta forma. Foi realizada a alteração da metodologia de desenvolvimento de softwares de uma instituição pública. Para isso, o processo que era utilizado foi mapeado e redesenhado. Todas as etapas do projeto, bem como seu detalhamento estão descritos neste trabalho.

## **1.2 Questão problema**

O Scrum é uma realidade para indústrias de softwares. As empresas que ainda não aderiram, provavelmente estão caminhando para isso. No contexto já descrito, onde há falta do conhecimento e mapeamento dos fluxos de trabalho por parte dos clientes, a metodologia tornou-se essencial. Isso ocorreu, pois a execução dos projetos em ciclos permite que os fluxos de trabalho do cliente, os quais são essenciais para a produção do software, possam ser conhecidos e mapeados aos poucos.

Na organização deste estudo de caso não houve nenhum projeto realizado com a metodologia Scrum até o ano de 2018. Para implementar a metodologia, a organização criou uma equipe teste. Enquanto a equipe executava o projeto e testava as ferramentas do Scrum, foi criada outra equipe para criar o processo padrão de desenvolvimento de

softwares utilizando a metodologia Scrum. Com o apoio deste processo mapeado, criaram-se então as outras equipes ágeis da organização no início do ano de 2019.

A organização em questão trata-se do setor de tecnologia de informação de uma instituição pública de ensino superior do estado de Minas Gerais. A organização produz softwares para serem utilizados na própria instituição, além de prover toda a infraestrutura necessária para isso. Os clientes da organização também se encontram dentro da instituição, sendo eles os responsáveis pelos demais setores.

Pode-se dizer que os próprios setores da instituição ainda estão consolidando os seus processos de trabalho. Mas para a criação de um software é necessário ter conhecimento dos processos que serão automatizados. Contudo, torna-se difícil a produção de softwares nesta instituição. Em contrapartida, a metodologia Scrum além de facilitar o entendimento do processo por dividir o projeto em ciclos, ajuda o próprio cliente e descrever os fluxos de seus processos.

Desta forma a questão problema pode ser assim colocada: Como adaptar a metodologia Scrum em um setor de TI de uma universidade pública?

### **1.3 Objetivos**

#### **1.3.1 Objetivo Geral**

O objetivo deste estudo é implantar a metodologia Scrum para desenvolver softwares no setor de TI de uma universidade pública do estado de Minas Gerais.

#### **1.3.2 Objetivo Específicos**

Os objetivos específicos podem ser descritos da seguinte forma:

- Estudo, descrição e desenho dos fluxos de desenvolvimento de software;
- Análise e seleção das etapas para implantação da metodologia *Scrum*;
- Implantação, descrição da metodologia e redesenho dos processos;
- Análise dos resultados após a implantação da metodologia.

## **1.4 Metodologia**

A pesquisa teve caráter científico e por esta razão há como ser classificada segundo os seguintes itens: natureza, objetivos e abordagem.

Quanto à natureza, trata-se de pesquisa aplicada, pois teve por função gerar conhecimento para o avanço científico, além de visar uma aplicação prática determinada / específica. Segundo Gil (2010), a pesquisa aplicada abrange estudos elaborados com a finalidade de resolver problemas identificados no âmbito das sociedades em que os pesquisadores vivem.

Quanto aos objetivos, pode-se apontar que a pesquisa é descritiva, pois nela se descrevem os dados da ‘população analisada’ que, neste caso, são os procedimentos adotados para a adaptação para criação de softwares definidos pelo projeto do governo. Lakatos e Marconi (2017) e Miguel (2011) afirmam que a pesquisa descritiva visa a detalhar fatos observados sem gerar interferências.

Por fim, quanto à abordagem, pode-se afirmar que se trata de pesquisa qualitativa, pois priorizou-se apontar os dados e as ações realizadas na instituição pesquisada.

Quanto ao procedimento técnico, utilizou-se a pesquisa direta com a aplicação de um questionário. Este questionário foi baseado em pesquisas semelhantes sobre implantação de metodologias ágeis. Ou seja, os questionários utilizados em outros estudos serviram de inspiração para a elaboração do questionário empregado na pesquisa aqui narrada.

Por fim, este trabalho trata-se de um estudo de caso. Pois é realizado um estudo de um acontecimento contemporâneo inserido na vida real. Por mais que a implantação da metodologia Scrum já esteja corriqueira, as particularidades da organização citada torna esse trabalho único.

## **1.5 Justificativas**

A justificativa por se trabalhar com a metodologia Scrum se deu pelo fato de que há uma facilidade para implantá-la no desenvolvimento de softwares, já que ela foi criada para este fim. A mudança do trabalho individual para o trabalho em equipe foi um dos principais pontos na alteração da metodologia de desenvolvimento de softwares do setor estudado.

Portanto, visto que os desenvolvedores trabalham diariamente na solução e a própria metodologia garante uma maior interação no trabalho em equipe, justifica-se o uso da metodologia Scrum.

Alterar a metodologia de trabalho de uma organização não é uma tarefa fácil. Entretanto, quando a mudança irá gerar resultados positivos tanto para a organização, quanto para comunidade, gera-se uma motivação para tal acontecimento. Portanto, o objetivo deste trabalho se justifica pelo estudo dos impactos que a alteração da metodologia de trabalho realmente gerou. Bem como o aumento da qualidade dos softwares, a eficiência do trabalho em equipe, e a diminuição do tempo de produção dos mesmos.

Os autores Carvalho e Mello (2012) aplicaram o método em uma empresa bem semelhante à deste estudo e concluíram que o método melhorou a comunicação e aumentou a motivação do time, diminuiu o custo, o tempo e o risco do projeto e aumentou a produtividade da equipe.

A utilização do estudo de caso para este trabalho se justifica porque há uma coleta de dados de um setor real a fim de analisar os impactos que a mudança da metodologia utilizada para o trabalho do dia-a-dia dos colaboradores causou. Portanto, neste trabalho há um estudo de um caso de um setor, bem como a análise dos resultados deste caso.

## **1.6 Estrutura do texto**

Este estudo está dividido em 5 capítulos, sendo este o primeiro. No segundo, a revisão de literatura, que contém pesquisas sobre desenvolvimento de softwares, metodologias tradicional, ágil e Scrum, e mapeamento de processos. No terceiro capítulo está presente a metodologia, onde é explicado como o estudo foi realizado. No quarto serão apresentados os resultados e análises da aplicação. No quinto e último, as conclusões do trabalho.

Ao final do estudo estão anexados a aplicação dos questionários utilizados para o desenvolvimento do projeto, bem como o projeto em si. Este detalha todas as etapas do processo de desenvolvimento de softwares da organização que se trata este estudo.

## 2. Revisão de Literatura

A expressão “Métodos Ágeis” no Brasil se torna cada vez mais conhecida nos últimos anos por utilizar uma abordagem simplificada, resumida e sintetizada. Porém, ser ágil nos dias de hoje é fazer a diferença em relação aos concorrentes e, ao contrário do que se imagina exige muita disciplina e organização. (RUBIN, 2017).

A primeira versão da metodologia Scrum foi criada por Ken Schwaber e Jeff Sutherland em 1995, tal como descrito em Sutherland (2016). O Scrum propõe que os projetos sejam divididos em ciclos, permitindo uma abordagem de forma iterativa e possibilitando entregas incrementais. Além disso, descreve as equipes realizadoras dos projetos como sendo autogerenciáveis. Este conjunto de ações possibilita o aumento da produtividade e agilidade do projeto, além de um maior contato com o cliente.

O Scrum é de suma importância, pois discorre de metodologias ágeis para a gestão de projetos de desenvolvimento de software, a qual expressa dinamismo no ambiente de negócios já que a Tecnologia da Informação (TI) transpassa pela constante mudança e conseqüentemente por pressões por inovações e resultados melhores do produto final (Machado & Medina, 2017).

A metodologia é fundamentada por seis características: “flexibilidade de resultados e prazos, times pequenos, revisões frequentes, colaboração e orientação a objetos”, que constituem como um guia para boas práticas com o intuito de alcançar o sucesso com práticas de gerenciamento (Schwaber & Sutherland, 2013).

A colaboração é um dos pontos principais do Scrum, pois é necessário trabalhar em equipe. Segundo Rubin (2017), uma comunicação transparente permite que haja entendimento claro do que está acontecendo evitando imprevistos e fomentando a confiança mútua entre os membros da equipe.

É importante salientar que mesmo havendo interação entre a equipe, a mesma também deve existir entre equipe e stakeholders. A metodologia ágil Scrum, como modelo para o gerenciamento dos projetos, permite que os stakeholders realizem o trabalho de forma integrada e comunicativa (Souza, 2016).

O sucesso do Scrum depende das pessoas se tornarem mais proficientes em serem comprometidas, corajosas, focadas, mente abertas e respeitadas (SUTHERLAND; SCHWABER, 2017).

Para que se obtenha tal sucesso, a equipe é dividida com diferentes funções. O Scrum possui três papéis principais (ALLIANCE, 2016):

- Product Owner: Aquele que possui a visão do produto. Determina o que precisa ser feito e define a prioridade das funcionalidades de maior valor.
- Scrum Master: Aquele que ajuda o time de desenvolvimento a usar Scrum da melhor maneira possível para construir o produto.
- Time de desenvolvimento: Aqueles quem constroem o produto

O time deve ser auto-organizável e multifuncional, de modo que o ele define a melhor maneira para trabalhar e possui o conhecimento necessário para completar seu trabalho (SCHWABER; SUTHERLAND, 2016). Apesar de a equipe ser o ponto principal para o sucesso da metodologia, o Scrum é uma metodologia que detalha todas as cerimônias necessárias que, junto à organização do time eleva-se o sucesso do desenvolvimento dos projetos. Na tabela 1 é possível visualizar as maiores diferenças entre os métodos tradicional e ágil.

Tabela 1 – Método tradicional X Método ágil

<b>Método Tradicional</b>	<b>Método Ágil</b>
Planejamento preditivo e rígido	Planejamento adaptativo e flexível
Trabalho determinado pelo gerente	Trabalho escolhido pelos membros da equipe
Pouco envolvimento do cliente	Comunicação constante com o cliente
Modelo cascata	Modelo iterativo e incremental
Burocracia	Dinamismo
Entrega direta do produto final	Entrega por etapas
Requisitos estáveis	Requisitos mutáveis

Rivas e Souza (2014) realizaram uma comparação entre a utilização dos métodos tradicional e ágeis em fábricas de softwares. Para isso, eles levaram em consideração alguns fatores culminantes na criação de softwares, são eles: gestão de requisitos,

processos e controle, envolvimento do cliente e gestão da documentação. Porém, os autores encontraram algumas limitações e por isso generalizaram alguns fatores para realizar a comparação entre os dois métodos. Eles afirmam que o modelo comercial da organização, as regulamentações locais e o contexto tecnológico, são fatores que se alteram de fábrica para fábrica e que devem ser considerados para a escolha da metodologia.

Da Rocha, Oliveira e de Vasconcelos (2004) abordaram em sua pesquisa o processo de desenvolvimento de softwares. Para tal estudo, foi considerado o mapeamento dos processos quanto às diferentes metodologias de desenvolvimento. Com este estudo é possível perceber que para o conhecimento e gestão da metodologia de desenvolvimento de softwares, é importante conhecer os processos ditados pela metodologia e os processos que são executados.

Carvalho e Mello (2012) realizaram uma pesquisa em uma pequena empresa de base tecnológica, a qual aplicou o método *Scrum* no processo de desenvolvimento de softwares. Eles analisaram a implantação do método na empresa e concluíram que o método melhorou a comunicação e aumentou a motivação do time, diminuiu o custo, o tempo e o risco do projeto e aumentou a produtividade da equipe. Esses resultados proporcionaram para a organização uma maior competitividade no mercado, além de se tornar mais sucedida no desenvolvimento de softwares.

A metodologia *Scrum* ganhou tanto espaço na área de gerenciamento de projetos que atualmente está sendo aplicada em empresas de diversos ramos. Salgado *et al.* (2010) conta da aplicação *Scrum* para processos de softwares em uma empresa de produtos químicos. Já Rodrigues *et al.* (2017) falam sobre a aplicação do método em projetos na empresa do ramo de metalurgia.

Apesar de a metodologia *Scrum* contemplar todas as fases para o gerenciamento de projetos, ela pode também ser aplicada de maneira híbrida. Essa acontece quando na aplicação são usadas duas ou mais metodologias diferentes. De Souza, Nery e Maccari (2017) implementaram a metodologia tradicional em conjunto a ágil em uma indústria metalúrgica.

No âmbito de universidades federais, a metodologia *Scrum* pode ser aplicada em diversos setores, e não só no setor que desenvolve softwares. Ela pode se aplicar em

qualquer setor que realiza projetos, e até mesmo em locais que estes são desenvolvidos por alunos. Um exemplo é a aplicação do *Scrum* em grupos PET's (Programa de Educação Tutorial), como descrito em Gomes *et al.*, que aplicou esse método no PET da Universidade Federal de Juiz de Fora.

O *Scrum* define como essencial o trabalho em equipe, e inicialmente julgamos como equipe o trabalho presencial. Este é importante, pois a metodologia descreve encontros diários, bem como outras cerimônias importantes que são intercaladas em um pequeno espaço de tempo. Porém, o *Scrum* já foi implementado também em ambientes em que o trabalho é realizado à distância. Um exemplo pode ser visto em Silva, Barbosa e Carvalho (2016). Estes propuseram e avaliaram o ensino à distância do método e concluíram que o *Scrum* é viável para ser realizado em um ambiente virtual.

### 3. Metodologia

O local que foi aplicada a metodologia *Scrum* trata-se de um setor de Tecnologia da Informação de uma instituição de ensino superior no estado de Minas Gerais. O setor atende exclusivamente a instituição, desenvolve softwares para os outros setores e presta os demais serviços de TI para todos os envolvidos da instituição.

A organização é dividida em áreas macro de TI. Seu organograma é composto da seguinte maneira: direção que é responsável por responder diretamente ao órgão máximo da universidade (a reitoria) e aos responsáveis diretos das pró-reitorias. Abaixo dele, formando a equipe tática da organização, têm-se o Diretor de Projetos, o Diretor de Serviços e o Diretor de Governança. Cada um desses é responsável por uma equipe, os quais costumam ser Analistas de TI. O escritório de governança tem como um dos fins, melhorar os processos da empresa, e todos os diretores juntos definem e executam projetos para variados fins. A figura 01 apresenta o organograma da organização.

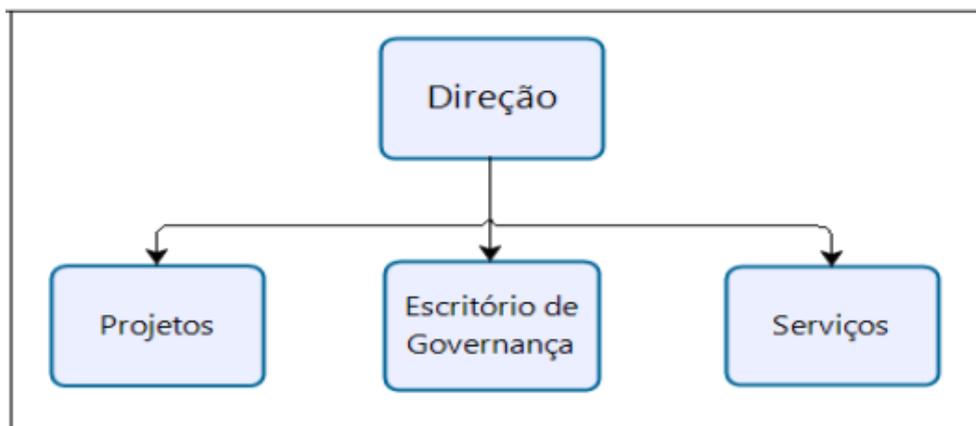


Figura 1 - Organograma da organização

Em junho de 2019, a organização possuía em seu quadro 40 funcionários, sendo que 20 trabalhavam como desenvolvedores de softwares. Destes 40, 21 possuíam cargo de Analista, 19 de Técnicos e 1 Secretário Executivo. Os mesmos eram divididos pelas macro áreas de TI. A tabela 2 detalha as equipes ali presentes. Vale ressaltar que o projeto o qual é tratado neste trabalho foi aplicado na área de desenvolvimento.

Tabela 2 - Número de funcionários

EQUIPES	ATIVIDADES	MEMBROS
Microinformática	Manutenção de computadores e equipamentos eletrônicos.	3
Serviços externos	Manutenção de serviços externos (Moodle, Pergamum, etc).	4
Conectividade	Serviços relacionados ao acesso à internet.	4
Telefonia	Serviços relacionados à telefonia.	3
Desenvolvimento	Desenvolvimento de projeto de software.	20

A mudança da metodologia de desenvolvimento de softwares, desenvolvimento este que se dá pelos analistas de TI, que trabalham dentro da equipe de projetos, criando sistemas para a universidade, foi uma demanda levantada pelos diretores. Esta demanda tinha como finalidade melhorar tal processo da organização. Antes dessa troca, eles desenvolviam sistemas através da metodologia tradicional, e por isso na maioria das vezes, cada analista era responsável sozinho e diretamente por um sistema.

Desde o início do setor já foram desenvolvidos cerca de 150 softwares, sendo que alguns deles já estão inativos e/ou já foram substituídos por softwares mais sofisticados. Todos estes foram executados através da metodologia tradicional, e só após a implementação do *Scrum*, em 2018, que os projetos passaram a ser executados com a metodologia ágil.

Para a implementação, foi escolhido um projeto e uma equipe para ser a equipe teste e de treinamento da metodologia. Enquanto a equipe começou o projeto e os testes junto com o diretor de projetos, foi criada outra equipe para mapear os processos dos projetos que eram realizados com a metodologia tradicional e ágil. Após esse mapeamento foi feito um desenho dos fluxos, bem como uma descrição das atividades que devem acontecer com a metodologia ágil, para que então, fossem formadas as outras equipes para executar projetos juntos com a metodologia *Scrum*.

Dos 20 funcionários da área de desenvolvimento, 16 foram entrevistados na etapa do mapeamento do processo. Os outros 4 estavam indisponíveis no momento por motivos tais como, férias, licença do trabalho, entre outros. Quanto aos clientes, foram realizadas 12 entrevistas com aqueles que se mostraram mais disponíveis, que ainda trabalhavam no local e que tinham conhecimento sobre o processo do desenvolvimento do sistema que a ele foi realizado.

## 4. Resultados

Este capítulo contém os resultados do projeto. Aqui é detalhado como as etapas foram executadas para obter o resultado esperado. O projeto para mudança da metodologia de desenvolvimento de softwares foi dividido em algumas etapas. Primeiro, o mapeamento dos processos, depois os desenhos dos novos fluxos, e por último a implementação. Para um melhor entendimento, este capítulo está dividido em seções de acordo com estas etapas.

### 4.1 Mapeamento de processos

A primeira fase da implementação da metodologia *Scrum* para o desenvolvimento de softwares da organização se deu através do mapeamento do processo de desenvolvimento de softwares. O objetivo deste mapeamento era entender como estava sendo executados os projetos e como deveriam ser utilizando o *Scrum* alinhado com os processos padrões do PDTI.

Para a execução do mapeamento do processo foi elaborado um questionário no *Google Forms* contendo perguntas de todas as fases da execução. Este tinha como finalidade entender se essas etapas eram realmente seguidas e executadas em todos os projetos. As perguntas do questionário foram elaboradas de acordo com os processos ditados pela metodologia tradicional.

O questionário foi dividido em sete seções, cada uma delas era composta por perguntas sobre as fases do desenvolvimento de softwares através da metodologia tradicional. As fases eram as seguintes:

- Informações pessoais, com 11 perguntas;
- Planejamento, com 15 perguntas;
- Prototipagem, com 4 perguntas;
- Execução, com 13 perguntas;
- Testes, com 4 perguntas;
- Treinamento, com 8 perguntas;
- Execução em grupo, com 4 perguntas.

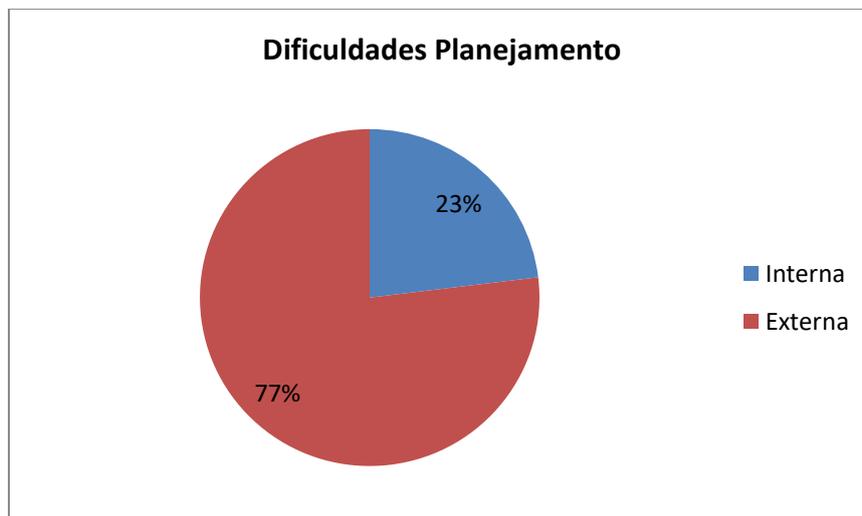
A primeira seção tinha o objetivo de entender o perfil dos colaboradores bem como definir o projeto base da entrevista. A última tinha como finalidade fazer um levantamento de quantos desenvolvedores já trabalharam em grupo e quantos conheciam a metodologia *Scrum*. Já as demais foram elaboradas para que os entrevistadores explicassem o projeto e detalhassem todas as etapas, respondendo se as que estavam no questionário realmente foram executadas e como foram, e se não, o porquê disso.

Para o preenchimento do questionário, foram realizadas 16 reuniões com desenvolvedores. Essas reuniões eram realizadas de forma presencial aonde a equipe ia preenchendo as respostas no próprio questionário. Após essa reunião a equipe respondia outro questionário no *Google Forms* para detalhar melhor as respostas, pois na hora da reunião a equipe havia pouco tempo para escrever e muitas vezes faziam daquela resposta um rascunho.

O gráfico 1 contém as respostas de uma pergunta do questionário para melhor entendimento. Esta pergunta está inserida na seção 2 do questionário, planejamento. O questionário completo encontra-se em anexo.

Pergunta: Quais foram as maiores dificuldades encontradas no planejamento?

Gráfico 1 – Resposta sobre as dificuldades do planejamento



Além de saber através do questionário se todas as fases realmente estavam sendo executada, uma das principais observações feitas era se a mudança da metodologia iria ajudar a diminuir os problemas. Por exemplo, no caso dessa pergunta, a maioria dos 77%

que afirmaram ter problemas externos na fase de planejamento, afirmaram ainda, que esses problemas eram causados pela distância do cliente. Além disso, outro ponto constatado foi a falta de conhecimento do processo que iria ser informatizado, além da falta de conhecimento dos requisitos necessário para desenvolvimento do software. Este caso foi analisado como um ponto a ser sanado com a metodologia *Scrum*, pois com ela, os requisitos não precisam ser levantados no início do projeto.

Para entender melhor ainda como se dava o desenvolvimento, a equipe realizou 12 reuniões com alguns clientes da empresa, que eram na maioria das vezes, cargos táticos e/ou estratégicos de pró-reitorias da universidade. Essas reuniões foram realizadas com o mesmo método das reuniões com os desenvolvedores. Foi criado um questionário no *Google Forms* para facilitar o trabalho e a compilação dos dados para a equipe.

Essas entrevistas eram marcadas por *e-mail* ou telefone. O entrevistado respondia as questões referentes ao desenvolvimento do software e tinha um espaço para expor sua opinião quanto do software, tanto do setor. O gráfico 2 contém a resposta de uma pergunta feita para os clientes referente ao protótipo, que era executado na fase de planejamento do software.

Pergunta: O protótipo do software foi essencial para você entender como ele realmente seria?

Gráfico 2 - Resposta sobre a importância do protótipo



Com essa pergunta, foi possível analisar que para a maioria dos projetos que houve protótipo, o mesmo foi essencial para o cliente entender e saber na primeira fase como o software seria no final. Porém, a metodologia *Scrum* não propõe a criação de protótipo porque o projeto é realizado de forma incremental. Mas, o cliente não deixaria de ter conhecimento sobre a execução do software, pois ao final de cada *Sprint* ele receberá uma entrega.

O questionário para os clientes havia apenas uma seção com 18 perguntas. Estas foram baseadas nas etapas que realmente deveria haver a participação do mesmo na execução do projeto. O intuito era descobrir se o cliente possuía contato com o desenvolvedor e se este contato realmente foi o necessário para a execução do sistema. Além disso, no final do questionário foi adicionada uma pergunta de *feedback* baseada na metodologia NPS (*Net Promoter Score*). Essa pergunta tinha como objetivo fazer o uso de uma métrica para medir a satisfação dos clientes com a empresa.

Após a coleta de todas as respostas, foi executada a compilação dos dados e desenho do processo atual. A equipe analisou todas as etapas do processo, como e quando elas eram realizadas e criaram um documento de texto detalhando cada uma das etapas. Esse texto continha os números, na maioria das vezes em porcentagem, dizendo quantos por cento realizaram ou não cada etapa. Ainda, quão difícil era a realização de cada etapa e quais foram as maiores dificuldades, tanto dos desenvolvedores, quanto dos clientes. Bem como, quais etapas não poderiam deixar de existir, ou seja, o que os desenvolvedores achavam essencial na execução dos sistemas.

O texto com a compilação dos dados também ajudou a equipe que realizou o projeto a comparar a metodologia tradicional com a ágil e já perceber quais eram os pontos positivos e negativos da troca da metodologia. Após a criação deste texto, foi alinhado e apresentado para o diretor de projetos, que trabalha diretamente com esses desenvolvedores, bem como os *feedbacks* do cliente. Os quais surpreenderam os diretores e fez com os que tivessem expectativa em melhorar o processo de desenvolvimento de softwares.

## 4.2 Processo de software SISP

O Núcleo de Padronização Tecnológica tem como uma de suas competências a definição de padrões e processos. Contudo, o mesmo realizou algumas pesquisas em 2010 para identificar quais as principais necessidades de padronização do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP).

Depois de identificado as necessidades, o núcleo criou um processo padrão de desenvolvimento de software para o SISP. Este processo é um guia para as entidades públicas de Tecnologia da Informação seguirem. O mesmo tem como um dos principais objetivos elevar o nível de maturidade dos órgãos em processos de gestão de projetos, e promover a padronização de processos e artefatos.

Vale ressaltar que foi criado um processo que serve como um guia para o desenvolvimento de softwares, e a escolha da metodologia para esse processo fica a critério de cada órgão. Por isso, que o processo do órgão a qual se refere esse trabalho foi baseado no padrão de desenvolvimento do SISP e na metodologia *Scrum*.

Encontra-se na internet, no site do SISP, o documento usado de base para o trabalho. O mesmo encontra-se também nas referências deste texto. Mas para melhor visualização, a figura 2 detalha o fluxo das atividades da primeira etapa do processo de desenvolvimento de softwares elaborado.

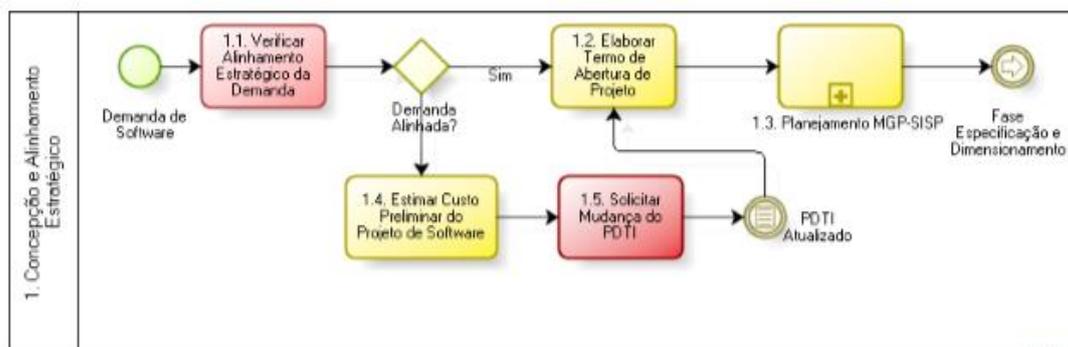


Figura 2 - Fase 1 do processo do SISP

Além desses fluxos, nos documentos têm-se também o detalhamento de cada atividade, e textos que explicam como deve ser a execução de cada etapa. É possível encontrar no SISP também, os artefatos padrão.

O documento é bem detalhado e o processo foi muito bem estruturado, porém não é viável e nem recomendado implantar o processo como está descrito. Foi por isso que para o novo processo do setor estudado, foi feito um mapeamento de como era o desenvolvimento, um estudo do SISP e um estudo da metodologia *Scrum*. Para que ao final, a equipe criasse um processo que era baseado no SISP, adequado à realidade do setor e executado com a metodologia *Scrum*.

### 4.3 Redesenho do processo

A terceira fase do projeto se deu com o redesenho do processo de desenvolvimento de sistemas. Após o mapeamento do processo que era utilizado, foi criado um novo processo alinhando-se o padrão do governo e a metodologia *Scrum*.

Para essa atividade foi utilizado o programa *Bizaggi* para auxiliar na construção das fases e das atividades de todo o processo. A primeira etapa a ser desenhada foi o processo com as macro etapas. A figura 4 representa o fluxo geral.

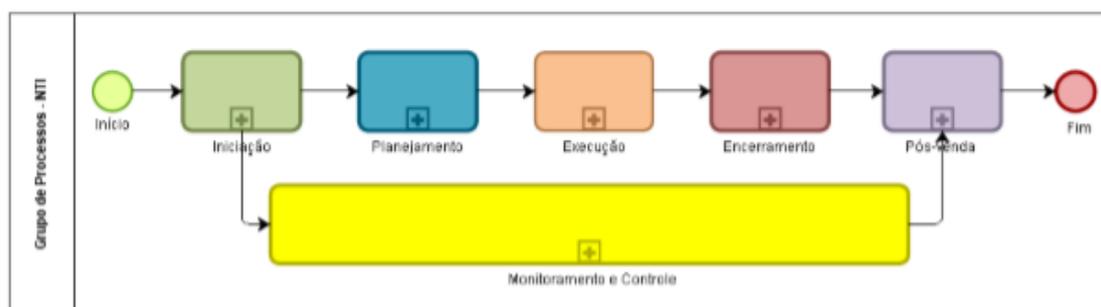


Figura 3 - Processo geral

Após essa construção, foram criados os fluxos de cada macro etapa. Para isso, a equipe se reunia e com a experiência e conhecimento, decidiam o fluxo ideal. Para dar continuidade no trabalho e não haver retrabalho, a equipe fazia a validação com o gerente de projetos e, quando necessário, as alterações. A figura 4 detalha a primeira etapa das atividades.

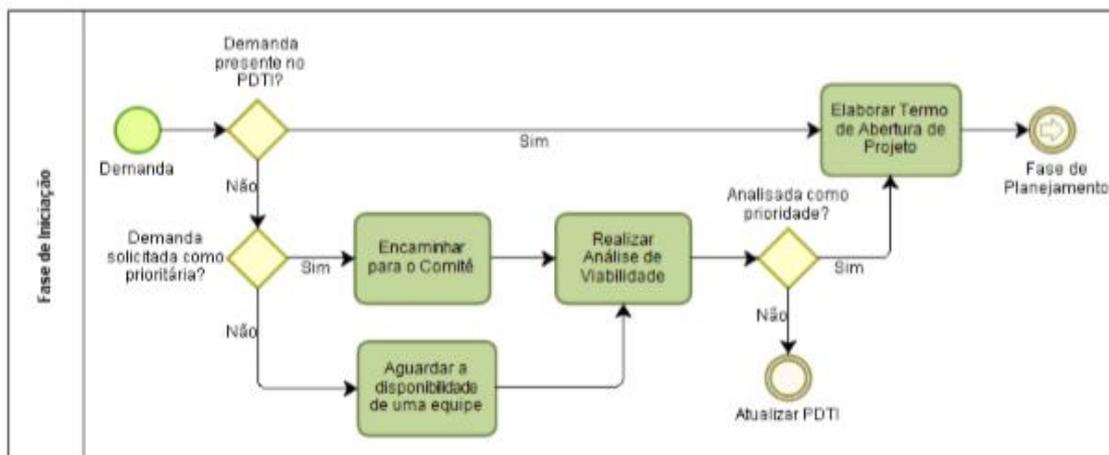
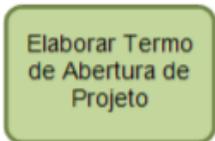


Figura 4 - Fase 1 do processo

Depois de validado pelo gerente cada fluxo, fazia-se o detalhamento e o resumo da etapa. Vale ressaltar que o gerente validava e que em vários momentos havia discussão entre a equipe e o gerente para chegarem a um consenso sobre o melhor fluxo. Para o detalhamento, elaborava-se um texto sobre a etapa em geral, e uma tabela de cada atividade. Tal tabela foi baseada no SISP, e continha o objetivo da atividade, os responsáveis por ela, entradas, descrições das atividades e saídas. A tabela 4 detalha as atividades.

Tabela 3 - Detalhe da atividade 1

<b>1. Elaborar Termo de Abertura de Projeto (TAP)</b>	
	<b>Objetivo:</b> Documentar a abertura do projeto para iniciar a fase de planejamento.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Oficialização de Demanda (DOD);</li> <li>• Plano Diretor de Tecnologia da Informação (PDTI).</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Verificar se a demanda recebida está alinhada com os objetivos estratégicos;</li> <li>• Descrever demanda de forma sucinta;</li> <li>• Estimar os custos do projeto;</li> <li>• Identificar os riscos do projeto;</li> <li>• Aprovar o Termo de Abertura de Projeto (TAP).</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Termo de Abertura do Projeto (TAP).</li> </ul>	

Na maioria das atividades, as entradas e saídas dependem e contém alguns artefatos. Estes são documentos padronizados que devem ser preenchidos e guardados para além de guardar e centralizar as informações durante o projeto haja uma cogestão futura para o setor. Estes artefatos também foram elaborados pela equipe e baseados no SISP. Segue abaixo um documento para melhor entendimento.



TERMO DE ABERTURA DO PROJETO (TAP)

**1. ABERTURA DO PROJETO**

Alinhado ao Plano Diretor de Tecnologia da Informação (PDTI) e após a realização da Análise de Viabilidade do Projeto (AVP), este documento tem como finalidade formalizar a abertura do projeto para dar início à fase de planejamento do mesmo.

**2. IDENTIFICAÇÃO DO PROJETO**

Nome do projeto:	Sigla:
Área/Setor:	Sigla:

**3. PARTES INTERESSADAS**

Nome	Telefone(s)	E-mail

**4. RESPONSÁVEIS PELO DESENVOLVIMENTO DO PROJETO**

Nome	Responsabilidade	Telefone(s)	E-mail
	<Product Owner>		
	<Scrum Master>		
	<Dev Team>		
	<Dev Team>		
	<Dev Team>		

**5. DESCRIÇÃO DO CENÁRIO/PROBLEMA**

<Descrever qual o objetivo para o qual órgão está realizando o projeto, que pode ser: descrição concreta de que o projeto quer alcançar; uma posição estratégica a ser alcançada; um resultado a ser obtido; um produto a ser produzido ou um serviço a ser realizado. Os objetivos devem ser específicos, mensuráveis, realizáveis, realísticos, e baseados no tempo.>

**6. ALINHAMENTO ESTRATÉGICO**

<Descrever qual alinhamento com o Plano Diretor de Tecnologia da Informação (PDTI), com o setor, ou com o governo.>

Figura 5 - Parte 1 do Termo de abertura do projeto



**7. DO REQUISITANTE**

Declaro estar ciente sobre o início da fase de planejamento do projeto, onde serão estipuladas as tarefas necessárias para a realização do software de forma sucinta e a estimativa de prazo para realização do projeto.

**8. ASSINATURA DOS RESPONSÁVEIS**

---

Nome: <Nome do requerente>  
Cargo/Função: <Nome do cargo ou função>  
Data: <dd/mm/aaaa>

---

Nome: <Nome do responsável pelo projeto>  
Cargo/Função: <Nome do cargo ou função>  
Data: <dd/mm/aaaa>

Figura 6 – Parte 2 do Termo de abertura do projeto

#### 4.4 Implantação do processo

Para a implantação do novo processo de desenvolvimento de softwares foi necessário realizar algumas alterações na equipe. O primeiro passo foi a criação das equipes ágeis, para desenvolver os projetos em grupo. Antes, como era utilizada a metodologia tradicional, todos os integrantes trabalhavam individualmente. Na equipe de desenvolvimento foi criada cinco equipes com quatro integrantes em cada uma delas.

Um ponto a ser considerado é que a empresa mudou de sede e isso ajudou muito para a mudança do processo. A antiga sede era composta por pequenas salas e alguns desenvolvedores dividiam a sala com outros, mas trabalhavam separadamente. Já a nova sede, é composta por uma sala grande e esta contém oito ilhas com quatro mesas de trabalho juntas.

Além disso, dentro da própria sala de desenvolvimento contém uma sala menor para reuniões. Esta foi estrategicamente construída para a realização de reuniões, bem como as reuniões diárias que é descrita pela metodologia *Scrum*.

O gerente de projetos ministrou um curso sobre metodologia *Scrum* para todos os desenvolvedores para que estes ficassem aptos a executar os projetos. Mas um curso nem sempre é o suficiente para consolidar uma mudança. Portanto, o gerente acompanhou ativamente todas as equipes para dar suporte, explicar mais a metodologia e retirar impedimentos que poderiam impossibilitar a execução do software.

#### **4.5 Melhoria no processo**

O processo descrito no SISP contém as fases de iniciação, planejamento, execução e encerramento. Nele não contém a última fase do processo da organização, que é a de pós venda.

Essa fase tem como objetivo acompanhar a implantação do software e realizar um *feedback* com o cliente. Neste *feedback* é coletado informações sobre o desenvolvimento da equipe durante o projeto e informações sobre o software. Essas visam entender se o software chegou em seu objetivo final, se ele atendeu aos requisitos e às expectativas do cliente, entre outros.

O *feedback* é realizado através de uma reunião, entre o cliente e uma equipe da gerência de governança. A equipe é formada por pessoas que não estiveram envolvidas no desenvolvimento do software para que o cliente se sinta mais à vontade para expressar suas opiniões.

No final do *feedback*, é usado uma métrica que tem como objetivo medir o grau de lealdade dos clientes com a organização. O NPS (*Net Promoter Score*) é calculado com base nas respostas de uma única pergunta: Qual é a probabilidade de você indicar a nossa organização a um colega? A resposta dessa pergunta deve ser um número, de 0 a 10.

A média de todas as respostas possibilita a organização medir o grau de qualidade de seus projetos. O objetivo da organização é ter uma média entre 9 e 10. As respostas são entendidas da seguinte maneira: um resposta entre 0 e 6, o cliente é rotulado como

detrator, ou seja, subentende-se que esse não ficou satisfeito com o projeto e não vai indicar a organização para outras pessoas. Respostas entre 7 e 8, é um cliente neutro, aquele que não ficou muito satisfeito e nem insatisfeito. Já as respostas entre 9 e 10, rotula-se os clientes como promotores. Estes ficaram muito satisfeitos com a organização, podem-se fidelizar e ainda fazer propaganda para outros clientes.

Essa métrica é utilizada por diversas organizações de vários setores. Através dela, é possível perceber em apenas uma resposta o grau de satisfação do cliente com o resultado do projeto. A aplicação dela junto com o *feedback* é um passo importante da organização para melhorar a qualidade do software depois de implantado.

## **5. Conclusão**

O objetivo inicial deste trabalho era implantar e analisar a metodologia *Scrum* para desenvolver softwares no setor de TI de uma universidade pública do estado de Minas Gerais. Primeiramente, a equipe que realizou o trabalho fez-se um estudo sobre desenvolvimento de softwares e metodologias desenvolvimento de softwares. Assim, a equipe ficou apta para entender como era realizado esse processo.

Com um mapeamento de processos, foi possível descrever os fluxos que eram realizados com a metodologia tradicional. Após esse mapeamento foi realizado um redesenho do mesmo fluxo alterando para a metodologia *Scrum*. Quando esse redenho foi validado pelo gerente de projetos da organização, fez-se um documento detalhando todas as fases, etapas e atividades do novo processo.

Este documento foi apresentado para todos os gerentes, os desenvolvedores e os clientes da organização. Após essa apresentação a equipe de projetos que consiste no gerente e nos desenvolvedores ficou apta a implantar o processo. Vale ressaltar que a mudança de sede da organização foi um ponto crucial para esta mudança, visto que os desenvolvedores começaram a trabalhar em mesas mais próximas e tiveram mais recursos para implantar o processo. Bem como, uma sala para realizar as reuniões, uma televisão dentro da sala para planejar as atividades do projeto, entre outros.

Portanto, o objetivo do trabalho foi atingido pois o processo de desenvolvimento de softwares através da metodologia *Scrum* foi implantado junto com a mudança da sede,

em janeiro de 2019. De janeiro até o momento da conclusão deste trabalho, a equipe ficou analisando a implantação do processo e realizando pequenas alterações, quando necessário.

Os autores que realizaram trabalhos parecidos com estes, os quais estão citados na revisão de literatura, relataram que o processo de desenvolvimento de softwares com a metodologia *Scrum* houve melhorias quando comparado com o uso da metodologia tradicional. Em resumo, essas melhorias foram listadas como: melhor comunicação da equipe, aumento da motivação da equipe, diminuição do tempo e do risco do projeto, e o ponto principal, aumento da produtividade da equipe para o desenvolvimento do software.

Essas melhorias também foram constatadas neste estudo. Ainda não foi realizada uma análise com indicadores para provar a comparação das metodologias. Mas durante os seis meses que a metodologia está sendo utilizada no processo, foi possível observar, tanto pelo gerente de projetos, quanto pelas equipes desenvolvedoras essas melhorias.

A comunicação entre a equipe e entre os clientes melhorou significamente. Isso foi constatado pelo desenvolvimento das atividades, que estão sendo realizadas de forma mais clara, objetiva e assertiva. O fato de haver reuniões com o cliente ao final de cada Sprint, já se prova como uma melhor comunicação, pois essas reuniões não eram realizadas com o método tradicional.

A motivação da equipe também foi um ponto constatado pelos gerentes e pela própria equipe. As reuniões diárias têm como objetivo reunir a equipe para que possa ser discutido o que foi realizado no dia anterior, o que será realizado no próprio dia e se há algum impedimento para isto. Durante esse encontro a equipe conversa e planeja as atividades de forma mais assertiva. Assim, eles se tornam mais motivados pois veem que o trabalho está gerando resultado e qualquer impedimento que possa travar a atividade eles conseguem se auto ajudar.

Quanto ao tempo de execução e o risco do projeto, estes também diminuiram significativamente. Anteriormente à metodologia tradicional os desenvolvedores executavam os projetos sozinhos e, às vezes, não tinham nem um tempo limite de conclusão, pois sempre apareciam impedimentos ao trabalho. Como agora os projetos são desenvolvidos em equipe, o desenvolvimento do software se dá de forma mais rápida.

Com a metodologia *Scrum*, o projeto é realizado em ciclos e cada ciclo tem uma meta que é alinhada com o cliente. As metas de cada ciclo compõe o objetivo final do projeto. Com esses ciclos, é mais difícil haver erros e retrabalhos no projeto. Por isso, o risco do projeto também diminuiu.

A produtividade da equipe é constituída por vários fatores, mas principalmente, o quanto a equipe consegue produzir em determinado tempo. Esta produtividade aumentou significativamente com a metodologia *Scrum*. Os pontos ditados acima completam e comprovam isso. Bem como, a comunicação e a motivação da equipe, o tempo e o risco do projeto.

Com este trabalho, foi possível perceber que a metodologia ágil para projetos de desenvolvimento de softwares é atualmente a melhor maneira de produzir com mais assertividade. A aplicação da metodologia *Scrum* na organização aumentou a autoestima dos colaboradores.

A metodologia *Scrum* possibilita gerenciar diversos projetos, e não só o desenvolvimento de softwares. A equipe que implantou também utiliza a metodologia para se autogerenciar. Assim, a equipe tem autonomia sobre suas decisões, bem como mais assertividade nas metas dos projetos.

Apesar da aplicação da metodologia ter sido feita na equipe de desenvolvimento de softwares, esse projeto possibilitou também uma visão de melhoria contínua para a organização inteira. Todos os colaboradores sentiram que estão avançando em termos de qualidade e estão se esforçando para melhorar ainda mais todos os processos da organização.

Como o processo é algo que foi implementado atualmente, pode-se dizer que ele é fluido e incremental. Por isso, até o momento da escrita deste trabalho, o processo está sofrendo atualizações para melhoria contínua. Porém, o trabalho que está em anexo foi a primeira versão do processo.

Depois dessa versão já foi feita algumas atualizações. A primeira fase, iniciação, foi dividida em duas, considerando as duas formas que o projeto pode ser solicitado ao setor. Todas as fases foram atualizadas dividindo as atividades em raias com seus respectivos donos. A fase de pós-venda teve o nome alterado para melhoria contínua.

Essas e outras foram atualizações realizadas para se aproximar ainda mais da realidade do cliente e dos desenvolvedores.

A equipe que realizou esse projeto continua na organização desenvolvendo outros projetos de melhoria contínua do setor. Assim, a equipe está próxima do processo e sempre que necessário, realiza as alterações. Atualmente, a equipe estudou o *ITIL*, um *framework* de gerenciamento de serviços de TI.

Um trabalho que deve ser realizado futuramente é a criação de indicadores no processo de desenvolvimento de softwares. Através de indicadores, poderá ser medida efetivamente a produtividade dos colaboradores. Assim, pode-se comparar o efeito que a mudança da metodologia gerou para a organização, para os colaboradores e os clientes.

## 6. Referências

ALLIANCE, S. **Scrum Roles Demystified**. 2016. Disponível em: <<https://www.scrumalliance.org/agile-resources/scrum-roles-demystified>>. Acesso em: 15/05/2019. Atlas, 2011.

Constâncio, F. L.; E Neto, J. S. **Correlação Entre O Nível De Engajamento Das Equipes De Projeto E O Desempenho Em Uma Empresa Pública Estruturada Por Projetos**. Revista De Gestão E Projetos – Gep, Vol. 7, N. 2, Maio/Agosto, 2016.

Da Silva, E. C. & Lovato., L. A. **FRAMEWORK SCRUM: EFICIÊNCIA EM PROJETOS DE SOFTWARE**, 2015.

De Carvalho, B. V. e Mello, C. H. P. (2012) **Aplicação do método ágil scrum no desenvolvimento de produtos de software em uma pequena empresa de base tecnológica**.

Date, N. M.; Pinochet, C. H. L.; Bueno, P. L. R.; Nemoto, O. M.C.M; **APLICAÇÃO DO MÉTODO ÁGIL SCRUM EM UMA FUNDAÇÃO EDUCACIONAL DO SETOR PÚBLICO**. Revista De Gestão E Projetos – Gep, Vol. 7, N. 2, Maio/Agosto, 2016.

GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 5ª ed. São Paulo: Atlas, 2010.

Jeff Sutherland. **Scrum a arte de fazer o dobro na metade do tempo** – 2ª Edição. 2016.

M. Alnoukari, Z. Alzoabi, S. Hanna, "Applying adaptive software development (ASD) agile modeling on predictive data mining applications: ASD-DM methodology", 2008 International Symposium on Information Technology, pp. 1-6, 2008.

MACHADO, Marcos; MEDINA, Sérgio Gustavo. **SCRUM – Método Ágil: uma mudança cultural na Gestão de Projetos de Desenvolvimento de Software**. Revista Científica Intr@ciência, UNIESP, 2017.

MARCONI, M. de A. LAKATOS, E. M. **Metodologia Científica**. 5ª ed. São Paulo.

Rivas, M. A. & de Souza, E. G. (2014) **ANÁLISE COMPARATIVA DA UTILIZAÇÃO DO MODELO TRADICIONAL (WATERFALL) DE DESENVOLVIMENTO DE PROJETOS E O MODELO ÁGIL (AGILE) EM FÁBRICAS DE SOFTWARE**.

Rodrigues, B. & Parreiras, F. **USO DE INTELIGÊNCIA ANALÍTICA NA ENGENHARIA DE SOFTWARE – UM MAPEAMENTO SISTEMÁTICO DA LITERATURA**. 2016.

RUBIN, Kenneth S.. **Scrum Essencial: Um Guia Prático Para o Mais Popular Processo Ágil**. Rio de Janeiro: Alta Books, 2017. Tradução de Roberto Resende.

Silva, V. B.; Barbosa, M. W. & Carvalho., L. A. C. **Experiências de ensino a distância do gerenciamento ágil de projetos com Scrum e apoio de uma ferramenta para gerência de histórias de usuário.** 2016

SOUZA, Francenila Rodrigues Júnior et al. **O Uso do Scrum Aplicado no Planejamento do Processo de Ensino-Aprendizagem.** III Congresso Internacional das Licenciaturas, 2016.

SUTHERLAND, J.; SCHWABER, K. **The Scrum Guide™.** 2017. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>>. Acesso em: 02.05.2019.

Vinícius Manhães Teles. **Extreme Programming** - 2ª Edição. 2014

## Anexos

### Questionário realizado com os desenvolvedores

12/07/2019

Questionário Desenvolvedor

#### Questionário Desenvolvedor

1. Projeto

---

2. Desenvolvedor Entrevistado

---

3. Você desenvolveu o projeto em equipe?

*Marcar apenas uma oval.*

Sim

Não

4. Se sim, quem era a equipe?

---

---

---

---

---

5. Departamento da Instituição que o sistema foi desenvolvido

---

---

---

---

---

6. Cliente responsável pelo sistema (pegar nome e se possível telefone e email)

---

---

---

---

---

**7. Há quanto tempo você trabalha na organização? (Como desenvolvedor)***Marcar apenas uma oval.*

- Até 1 ano
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- Mais de 10 anos

**8. Quantos projetos você já executou/participou? (Projetos maiores)***Marcar apenas uma oval.*

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- Mais de 10

**9. Formação / Trajetória na organização (se tiver tido mais de um cargo, tentar listar o tempo em cada cargo até ser concursado)**

---

---

---

---

---

**10. Porque você foi escolhido para desenvolver tal projeto?***Marcar apenas uma oval.*

- Disponibilidade
- Experiência
- Proximidade com o cliente
- Outro: \_\_\_\_\_

## 11. Observações sobre essa parte (relate algo que ache interessante que o cliente disse)

---



---



---



---



---

## Planejamento

## 12. Quais foram as maiores dificuldades encontradas no planejamento?

Marcar apenas uma oval.

- Interna
- Externa

## 13. Explique essa maior dificuldade

---



---



---



---



---

## 14. O quão bem você compreendeu a regra de negócios (processo interno do cliente)?

Marcar apenas uma oval.

	1	2	3	4	5	
Pouco	<input type="radio"/>	Muito bem				

## 15. O quão bem o CLIENTE compreendia a própria regra de negócios (processo interno do setor)?

Marcar apenas uma oval.

	1	2	3	4	5	
Pouco	<input type="radio"/>	Muito bem				

## 16. O quão bem a regra de negócios estava estruturada?

Marcar apenas uma oval.

	1	2	3	4	5	
Pouco	<input type="radio"/>	Muito bem				

## 17. O gerente de projetos participou de reuniões do planejamento junto com você?

Marcar apenas uma oval.

- Durante o início
- Durante todo o projeto
- Não participou

18. **Você acha essencial a presença do gerente de projetos nas reuniões de planejamento?**

*Marcar apenas uma oval.*

- Sim  
 Não

19. **Porque?**

---

---

---

---

---

20. **Como foi feita a validação do planejamento com o cliente?**

*Marcar apenas uma oval.*

- Online  
 Presencial

21. **O cliente validou o planejamento em primeira instância?**

*Marcar apenas uma oval.*

- Sim  
 Não

22. **Se sim, houve mudanças após a validação?**

*Marcar apenas uma oval.*

- Sim  
 Não

23. **O cliente respondeu ao chamado GLPI rapidamente? (Marcar apenas se tiver acontecido o chamado)**

*Marcar apenas uma oval.*

- Sim  
 Não

24. **Caso tenha demorado, você ficou ocioso?**

---

25. **Você só começou a executar após a validação?**

*Marcar apenas uma oval.*

- Sim  
 Não

**26. Observações**

---

---

---

---

---

**Prototipagem****27. Você realizou a prototipagem?**

*Marcar apenas uma oval.*

- Sim  
 Não

**28. Se sim, quanto tempo gastou?**

*Marcar apenas uma oval.*

- Até 1 semana  
 Até 2 semanas  
 Até 3 semanas  
 Até 4 semanas  
 Mais de 4 semanas

**29. A prototipagem ajuda o cliente a entender de forma mais clara o projeto?**

*Marcar apenas uma oval.*

- Sim  
 Não

**30. O que você acha de utilizar a prototipagem?**

---

---

---

---

---

**31. Observações**

---

---

---

---

---

**Execução**

**32. Você tinha facilidade em contatar os clientes durante o projeto?***Marcar apenas uma oval.*

- Sim  
 Não

**33. Qual era a facilidade em reunir com os clientes?***Marcar apenas uma oval.*

	1	2	3	4	5	
Muito Ruim	<input type="radio"/>	Muito Bom				

**34. Você tinha facilidade em contatar as partes interessadas (usuários) durante o projeto?***Marcar apenas uma oval.*

- Sim  
 Não  
 Talvez

**35. Você realizou entregas incrementais? Quantas?***Marcar apenas uma oval.*

- Nenhuma  
 1 a 3  
 4 a 6  
 7 a 10  
 Mais de 10

**36. Teve que alterar algo a pedido do cliente depois das entregas incrementais?***Marcar apenas uma oval.*

- Sim  
 Não

**37. Porque ocorreu essa alteração?***Marcar apenas uma oval.*

- A pedido do cliente  
 Erro no sistema  
 Outro: \_\_\_\_\_

**38. Se outro, especifique**

---

---

---

---

---

**39. O cronograma do projeto foi respeitado?***Marcar apenas uma oval.*

- Sim  
 Não

**40. O projeto foi entregue no tempo determinado?***Marcar apenas uma oval.*

- Sim  
 Não

**41. Se não, porque?***Marque todas que se aplicam.*

- Greve  
 Dificuldade de entendimento sobre a regra de negócios  
 Mudança da equipe  
 Dificuldade em contatar os clientes  
 Outro: \_\_\_\_\_

**42. Após finalizado o projeto, foi possível colocá-lo em produção?***Marcar apenas uma oval.*

- Sim  
 Não

**43. Para a entrega do software (inauguração), quem compareceu na reunião?***Marque todas que se aplicam.*

- Desenvolvedor  
 Gerente de Projetos  
 Direção  
 Cliente  
 Outro: \_\_\_\_\_

**44. Foi realizado feedback com o cliente para saber a satisfação do cliente com o software desenvolvido?***Marcar apenas uma oval.*

- Sim  
 Não  
 Não sei

**45. Você acha que o sistema atendeu às necessidades do cliente?***Marcar apenas uma oval.*

- Sim  
 Não  
 Talvez

**46. Observações**

---

---

---

---

---

**Testes****47. Quem realizou os testes do sistema?**

*Marque todas que se aplicam.*

- Próprio Desenvolvedor
- Cliente
- Outros desenvolvedores do NTI
- Pessoas externas

**48. Você acredita que seria interessante outra pessoa realizar também?**

*Marcar apenas uma oval.*

- Sim
- Não
- Talvez

**49. Porque você acha interessante outra pessoa realizar também?**

---

---

---

---

---

**50. Teve que alterar algo no sistema após os testes?**

*Marcar apenas uma oval.*

- Sim
- Não

**51. Se sim, porque teve que alterar?**

*Marcar apenas uma oval.*

- Erro no sistema
- Pedido do cliente
- Outro: \_\_\_\_\_

**52. Observações**

---

---

---

---

---

**Treinamento****53. Você executou treinamento?**

*Marcar apenas uma oval.*

- Sim  
 Não

**54. Para quem?**

*Marcar apenas uma oval.*

- Pessoas diretamente interessadas  
 Cliente  
 Outro: \_\_\_\_\_

**55. Quanto tempo você gastou?**

---

---

---

---

---

**56. Foi efetivo?**

*Marcar apenas uma oval.*

- Sim  
 Não  
 Talvez

**57. Você já teve que repetir o treinamento?**

*Marcar apenas uma oval.*

- Sim  
 Não

58. O que você acha se o treinamento fosse automatizado?

---

---

---

---

---

59. O que você acha se o treinamento fosse realizado por uma pessoa que não desenvolveu o sistema?

---

---

---

---

---

60. Observações

---

---

---

---

---

## Execução em grupo

61. Você já desenvolveu software em equipe dentro ou fora da Instituição?

*Marcar apenas uma oval.*

- Sim  
 Não

62. Se sim, como foi essa experiência?

---

---

---

---

---

63. Você conhece as metodologias ágeis para gerenciamento de projetos?

*Marcar apenas uma oval.*

- Sim  
 Não

64. O que você acha dessa metodologia?

---

---

---

---

---

65. Observações gerais

---

---

---

---

---



# Questionário realizado com os clientes

09/07/2019

Questionário Cliente

## Questionário Cliente

### 1. Projeto

\_\_\_\_\_

### 2. Entrevistado

\_\_\_\_\_

### 3. De onde surgiu a demanda do projeto? Qual foi a motivação?

*Marque todas que se aplicam.*

- Necessidade
- Legislação / Portaria
- Melhoria no setor
- Outro: \_\_\_\_\_

### 4. O Software desenvolvido foi eficiente? Por quê?

*Marque todas que se aplicam.*

- Sim
- Não
- Outro: \_\_\_\_\_

### 5. Quanto tempo demorou desde o pedido até o software ficar pronto?

*Marque todas que se aplicam.*

- até 2 meses
- de 2 meses a 6 meses
- de 6 meses a 1 ano
- 1 ano a 2 anos
- mais de 2 anos
- Outro: \_\_\_\_\_

**6. Quantas reuniões foram executadas para o planejamento?***Marque todas que se aplicam.*

- Nenhuma
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- Mais de 10
- Outro: \_\_\_\_\_

**7. Elas foram eficientes? Quem comparecia?***Marque todas que se aplicam.*

- Sim
- Não
- Outro: \_\_\_\_\_

**8. As etapas do desenvolvimento do software estavam claras para você?***Marque todas que se aplicam.*

- Sim
- Não
- Talvez
- Outro: \_\_\_\_\_

**9. O protótipo do software foi essencial para você entender como ele realmente seria?***Marque todas que se aplicam.*

- Sim
- Não
- Talvez
- Não houve Protótipo
- Outro: \_\_\_\_\_

**10. Os requisitos (funcionalidades/necessidades) do projeto foram atendidos?***Marque todas que se aplicam.*

- Sim
- Não
- Talvez
- Outro: \_\_\_\_\_

**11. Quais documentos foram entregues para você? Estavam bem explicados? O que poderia ter a mais?**

Marque todas que se aplicam.

- Nenhum
- Documento de Visão
- Documento de oficialização de demanda
- Outro: \_\_\_\_\_

**12. Você tinha contato direto com o desenvolvedor?**

Marque todas que se aplicam.

- Sim
- Não
- Talvez
- Outro: \_\_\_\_\_

**13. O cronograma do projeto foi respeitado? O software foi entregue no tempo planejado?**

Marque todas que se aplicam.

- Sim
- Não
- Talvez
- Outro: \_\_\_\_\_

**14. Durante o desenvolvimento você sugeriu mudanças? Dentre essas sugestões, você voltou atrás de alguma? (mudou a sugestão). Teve alguma sugestão que não havia sido planejada no projeto?**

Marque todas que se aplicam.

- Sim, sugeri
- Não
- Sim, voltei atrás
- Não
- Outro: \_\_\_\_\_

**15. Houve treinamento?**

Marque todas que se aplicam.

- Sim
- Não
- Outro: \_\_\_\_\_

**16. Foi efetivo?**

Marque todas que se aplicam.

- Sim
- Não
- Talvez
- Outro: \_\_\_\_\_

17. O quão satisfeito você está com a atual gestão da (organização)?  
*Marcar apenas uma oval.*

	1	2	3	4	5	
Muito Insatisfeito	<input type="radio"/>	Muito Satisfeito				

18. Você teria alguma sugestão de melhoria quanto ao processo de desenvolvimento?

---

---

---

---

---

# Mapeamento do Processo de Desenvolvimento de Softwares

## **1. Introdução**

A (ORGANIZAÇÃO) tem como objetivo normatizar, construir e disponibilizar o mapeamento do Processo de Desenvolvimento de Softwares através da metodologia ágil. Baseado no padrão SISP, este guia serve como modelo referência de um documento orientado, para possíveis aplicações de métodos ágeis, mantendo a autonomia dos projetos, mas podendo ser adaptado de acordo com as particularidades de cada um.

O uso desta metodologia permitirá padronizar o processo de desenvolvimento de softwares do (ORGANIZAÇÃO), alinhando os processos de trabalho e criando a documentação adequada. Fator que permitirá melhorias na governança, refletindo no aumento da eficácia, eficiência, efetividade, produtividade e menor dependência dos indivíduos, aumentando a retenção do conhecimento na organização.

O (ORGANIZAÇÃO) fazia uso da metodologia tradicional para execução dos projetos, porém, para execução do mesmo é necessário o levantamento prévio de todos os requisitos do produto e esta metodologia não permite encontrar gargalos no projeto durante o desenvolvimento, bem como a mudança do cronograma. Portanto, a demanda da criação do processo de desenvolvimento de sistemas através da metodologia ágil, especificamente a metodologia Scrum, surgiu a partir do gerente de projetos, uma vez que, com a metodologia tradicional era gasto muito tempo para elaborar o projeto e, muitas vezes, o planejamento não era efetivo para a execução.

### **1.1 Metodologia Scrum**

O Scrum é uma metodologia ágil para gerenciamento de projetos que propõe que os projetos sejam divididos em ciclos, permitindo uma abordagem de forma iterativa e possibilitando entregas incrementais. Essa metodologia possibilita o aumento da produtividade e agilidade do projeto, além de um maior contato com o cliente. O modelo adotado permite que os projetos sejam realizados em equipes autogerenciáveis, no caso do (ORGANIZAÇÃO), composto geralmente por uma equipe de 3 a 4 pessoas.

Os ciclos do Scrum são chamados de sprints e possuem duração de tempo pré-determinada no planejamento para que o cliente possa estar ciente do tempo aproximado em que o produto ficará pronto. No (ORGANIZAÇÃO), geralmente, cada sprint possui 14 dias. Porém, diferente da metodologia tradicional, o planejamento através da metodologia ágil é realizada de forma sucinta, já que não necessita do levantamento de todos os requisitos do produto. Isso possibilita que ocorram mudanças no planejamento, tanto da duração, quanto das atividades a serem executadas. A estimativa do projeto é de caráter informativo para o cliente, que deve estar ciente que o cronograma estará sujeito a alterações.

Ao final de cada sprint é realizada uma reunião de retrospectiva entre equipe e cliente com objetivo de fazer uma análise sobre a última sprint e alinhar a expectativa do produto final. Além disso, essa entrega incremental permite transparência da evolução do projeto e interação entre equipe e cliente.

## **2. Definição dos Papéis**

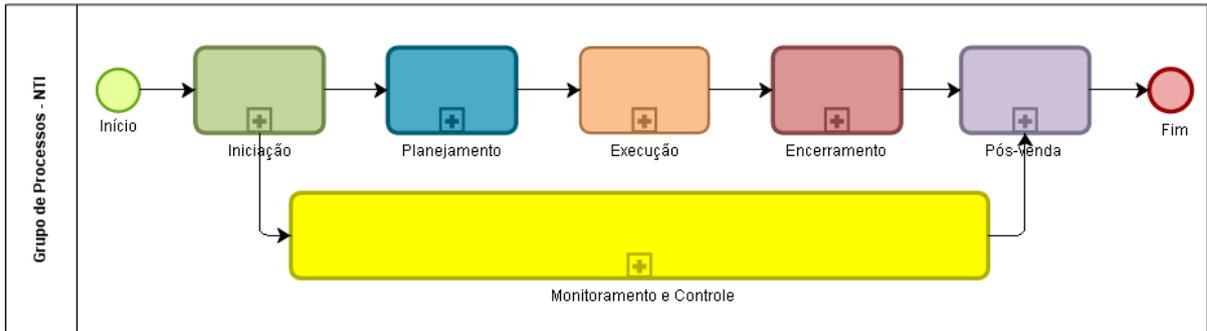
Para facilitar o entendimento do processo e normatizar os envolvidos no processo, segue abaixo o quadro que descreve os papéis de cada envolvido na execução de Projetos de Desenvolvimento de Softwares.

<b>Papel</b>	<b>Descrição</b>
Comitê de Tecnologia da Informação (CTI)	É um grupo formado por representantes de cada setor da Universidade, com o objetivo de definir os critérios para a seleção dos projetos, bem como a priorização destes no PDTI.
Gerente de Projetos	Responsável pela interface inicial entre projetos e as partes interessadas. Define a equipe responsável pelo planejamento e execução e faz o acompanhamento dos projetos, ou seja, o controle e monitoramento para medir o desempenho do projeto e da equipe.
Dono do Produto (PO)	É considerado o principal responsável por definir as diretrizes do projeto. Representando o cliente dentro da equipe, atua como ponto de contato com a equipe de desenvolvimento do produto, além de coletar as Histórias de Usuário, registrá-las no Backlog do Produto e garantir que as prioridades estabelecidas pelo cliente sejam respeitadas.
Scrum Master	Atua com o objetivo de aumentar a eficiência e a qualidade do trabalho da equipe assegurando que as técnicas do Scrum sejam aplicadas de forma eficiente.
Equipe de desenvolvimento	Equipe autogerenciável que executa o projeto, além de priorizar e estimar o tempo para cada ação necessária para o produto final.
Cliente	Pessoa que representa a organização que adquire o produto. Nem sempre esta irá utilizar o sistema. É responsável por informar os requisitos do projeto, aprovar o planejamento, acompanhar a execução através de reuniões com a equipe em todo sprint.
Partes interessadas	São os próprios usuários do sistema, que nem sempre atuam como cliente. São responsáveis por dar informações detalhadas do processo quando a equipe solicitar.
Equipe pós-venda	Equipe responsável para realizar feedback do produto diretamente com o cliente. Garantindo a satisfação e documentando o parecer do mesmo.

### 3. Grupo de Processos

O Processo de Desenvolvimento de Softwares do (ORGANIZAÇÃO) é formado pelas seguintes atividades que ocorrem ao longo de todo o projeto. Em geral, a saída de um processo torna-se a entrada de outro, ou uma entrega do projeto.

O processo foi baseado no padrão do SISP e adaptado para a realidade do (ORGANIZAÇÃO) e, com o intuito de melhorar de forma contínua os projetos, foi criada a fase de pós-venda. Nessa fase será realizado o acompanhamento após a implantação do software para entender a satisfação do cliente com o projeto entregue.



Para melhor entendimento, cada fase do processo encontra-se sucintamente descrita na tabela abaixo. O número de processos diz respeito à quantidade de processos contidos no grupo e os artefatos são os documentos que devem ser criados e preenchidos em cada fase.

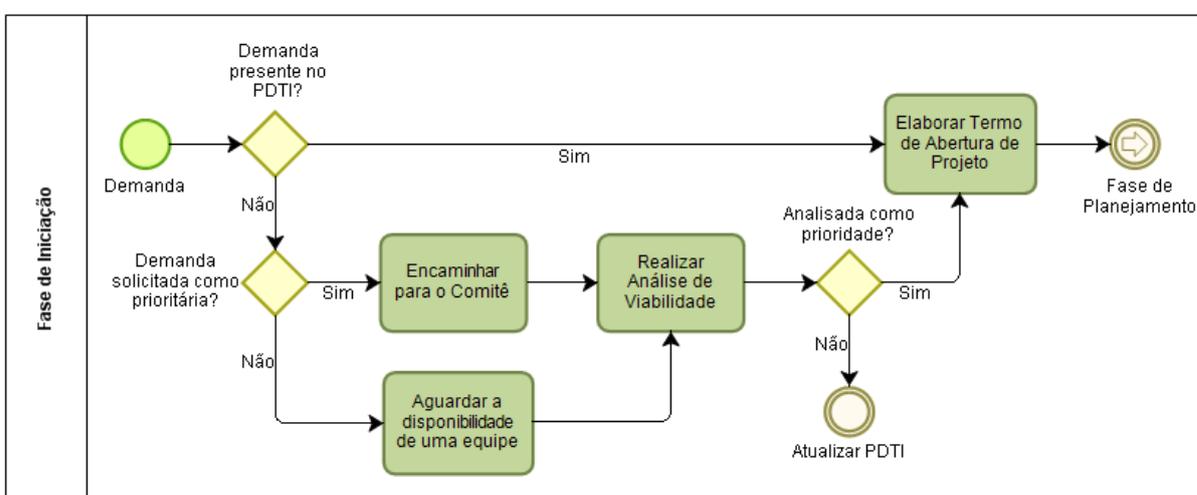
Grupo de processo	Nº de processos	Nº de artefatos	Descrição
 Iniciação	4	3	São os processos realizados para definir um novo projeto à partir da prioridade presente no PDTI. É formado pela oficialização da demanda, análise de viabilidade e termo de abertura do projeto.
 Planejamento	7	2	São os processos realizados para planejar as ações do projeto através do levantamento dos requisitos de forma sucinta. Neste grupo começa-se a interação entre a equipe e as partes interessadas.
 Execução	4	3	São os processos realizados para executar o projeto através da metodologia Scrum. A execução é feita através de Sprints.
 Encerramento	2	2	São os processos para finalizar as atividades do projeto, visando finalizar formalmente e implementar o projeto.
 Monitoramento e Controle	2		São os processos realizados para observar e acompanhar a execução do projeto, bem como, medir o desempenho do projeto e da equipe.

 Pós-venda	2	1	São os processos realizados para ide(organização)ficar a satisfação das partes interessadas quanto ao projeto executado.
--	---	---	--

#### 4. Detalhamento das fases de cada grupo

Essa fase tem como objetivo detalhar cada atividade dos grupos de processo.

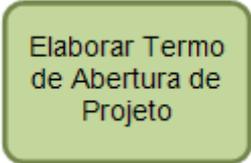
##### 4.1 Iniciação

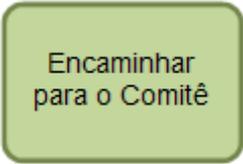


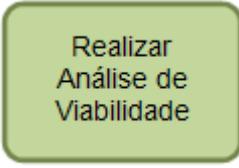
Para a iniciação do processo é preciso ter o conhecimento das demandas da Universidade. Para isso, todas as e(organização)dades devem enviar para o Comitê as suas demandas documentadas no formato padrão do DOD - Documento de Oficialização de Demanda.

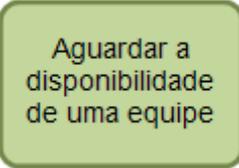
No ano de 2019 o Comitê irá construir o PDTI - Plano Diretor de Tecnologia da Informação e este irá conter todas as demandas previamente documentadas em ordem de prioridade para serem executadas. Portanto, este grupo de processo considera o PDTI construído e a atuação do Comitê.

O processo sempre terá início quando uma equipe estiver disponível, ou no caso especial, quando chegar uma demanda prioritária que não se encontra no PDTI. Neste caso, o início do processo terá duas possibilidades. A primeira, caso o cliente afirme que a demanda é realmente prioritária, o início será “encaminhar para o Comitê”, que irá avaliar a demanda e terá autonomia para pausar um projeto em execução. A segunda possibilidade irá acontecer quando a demanda for prioritária, mas não a ponto de ter que parar um projeto em execução. Portanto, ela vai para a fase de “aguardar disponibilidade de uma equipe” para que quando alguma equipe estiver disponível possa realizar a análise de viabilidade para avaliar a prioridade da demanda.

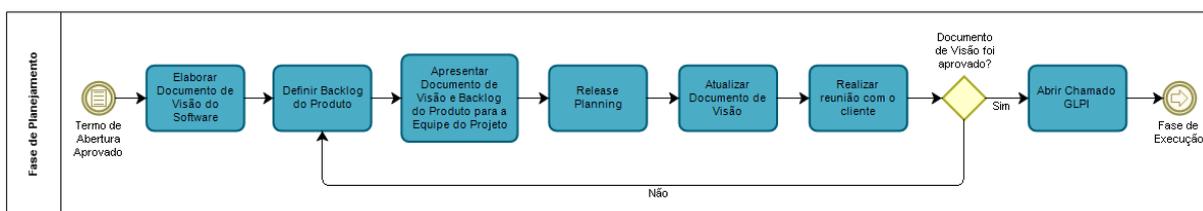
<b>1. Elaborar Termo de Abertura de Projeto (TAP)</b>	
	<b>Objetivo:</b> Documentar a abertura do projeto para iniciar a fase de planejamento.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Oficialização de Demanda (DOD);</li> <li>• Plano Diretor de Tecnologia da Informação (PDTI).</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Verificar se a demanda recebida está alinhada com os objetivos estratégicos;</li> <li>• Descrever demanda de forma sucinta;</li> <li>• Estimar os custos do projeto;</li> <li>• Ide(organização)ficar os riscos do projeto;</li> <li>• Aprovar o Termo de Abertura de Projeto (TAP).</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Termo de Abertura do Projeto (TAP).</li> </ul>	

<b>2. Encaminhar para o Comitê</b>	
	<b>Objetivo:</b> Decidir se uma equipe deverá parar suas atividades para atender uma demanda prioritária.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• Gerente de Projetos;</li> <li>• Cliente.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Oficialização de Demanda (DOD);</li> <li>• Solicitação para análise do Comitê.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Analisar a demanda a ser incluída;</li> <li>• Analisar a motivação e justificativa;</li> <li>• Selecionar o projeto a ser pausado;</li> <li>• Selecionar equipe disponível.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Análise de Viabilidade.</li> </ul>	

<b>3. Realizar Análise de Viabilidade</b>	
	<p><b>Objetivo:</b> Equipe realiza Análise de Viabilidade à partir de especificações técnicas do projeto para complementar a decisão do Comitê.</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>• Equipe de desenvolvimento.</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Solicitação do Comitê;</li> <li>• Documento de Oficialização de Demanda (DOD).</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>• Realizar Análise de Viabilidade;</li> <li>• Estimar os custos do projeto;</li> <li>• Ide(organização)ficar os riscos do projeto;</li> <li>• Decidir se o projeto é viável;</li> <li>• Decidir se o projeto é mesmo prioritário.</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>• Termo de Abertura do Projeto (TAP) - Caso dado início ao projeto;</li> <li>• PDTI atualizado.</li> </ul>	

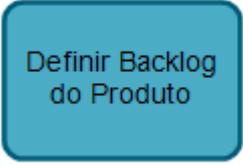
<b>4. Aguardar disponibilidade de equipe</b>	
	<p><b>Objetivo:</b> Aguardar um projeto finalizar para que a equipe disponível realize a Análise de Viabilidade.</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>• Cliente.</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Documento de oficialização de demanda (DOD).</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>• Demanda aguarda a disponibilidade de uma equipe.</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>• Análise de Viabilidade.</li> </ul>	

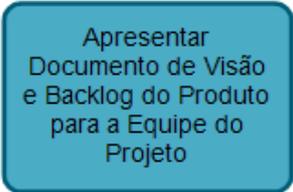
## 4.2 Planejamento

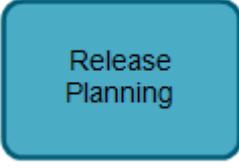


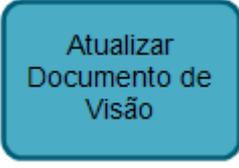
O planejamento do projeto com a metodologia Scrum se dá de forma mais sucinta, ou seja, essa fase deve ser rápida sem haver muitos impedimentos. Após o Product Owner definir as tarefas necessárias para a realização do projeto, a equipe classifica cada tarefa com o número de pontos. Estes pontos indicam a complexidade e a estimativa de tempo necessária para a execução de cada uma. Através destes pontos, a mesma equipe consegue estimar o número de Sprints que o projeto irá precisar. O número de Sprints é apenas uma estimativa, pois eles poderão sofrer alterações ao longo do desenvolvimento.

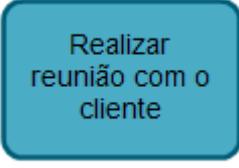
1. Elaborar Documento de Visão do Software (DVS)	
	<p><b>Objetivo:</b> Ide(organização)ficar os requisitos básicos e definir o escopo do produto para elaboração do Documento de Visão do Software (DVS).</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>• PO</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Termo de Abertura do Projeto (TAP);</li> <li>• Análise de Viabilidade.</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>• Realizar reunião de história do usuário;</li> <li>• Definir backlog do produto;</li> <li>• Definir aquisições e contratações;</li> <li>• Validar as informações levantadas.</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS);</li> </ul>	

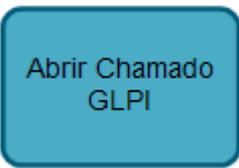
<b>2. Definir Backlog do Produto</b>	
	<b>Objetivo:</b> Planejar e listar as tarefas necessárias para a execução do produto.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS).</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Estimar o tamanho do projeto;</li> <li>• Planejar de forma sucinta o projeto;</li> <li>• Listar as tarefas necessárias para execução do produto.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Backlog - Lista de todas as tarefas necessárias para execução do produto.</li> </ul>	

<b>3. Apresentar Doc. de Visão e Backlog do Produto para a Equipe do Projeto</b>	
	<b>Objetivo:</b> Apresentar o projeto para a equipe de desenvolvimento.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO</li> <li>• Scrum Master</li> <li>• Equipe de Desenvolvimento</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS)</li> <li>• Backlog do Produto</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Reunir com a equipe de desenvolvimento;</li> <li>• Apresentar o projeto e todas as tarefas necessárias.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS) atualizado, caso necessário.</li> </ul>	

<b>4. Release Planning</b>	
	<b>Objetivo:</b> Estimar a complexidade e o tempo necessário para realizar cada tarefa do backlog.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• Equipe de Desenvolvimento.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS);</li> <li>• Backlog do Produto.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Estimar a complexidade de cada tarefa;</li> <li>• Estimar o número de “pontos” necessário para realizar cada tarefa;</li> <li>• Alocar as tarefas em Sprints;</li> <li>• Definir o número de Sprints necessários inicialmente para execução do projeto.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS) atualizado;</li> <li>• Backlog do Produto dividido em Sprints.</li> </ul>	

<b>5. Atualizar Documento de Visão</b>	
	<b>Objetivo:</b> Adicionar no DVS a estimativa de tempo do projeto, baseada no número de Sprints.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS);</li> <li>• Backlog atualizado.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Adicionar o tempo estimado de duração do projeto no DVS;</li> <li>• Criar a lista de tarefas no OpenProject;</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS) atualizado;</li> <li>• OpenProject.</li> </ul>	

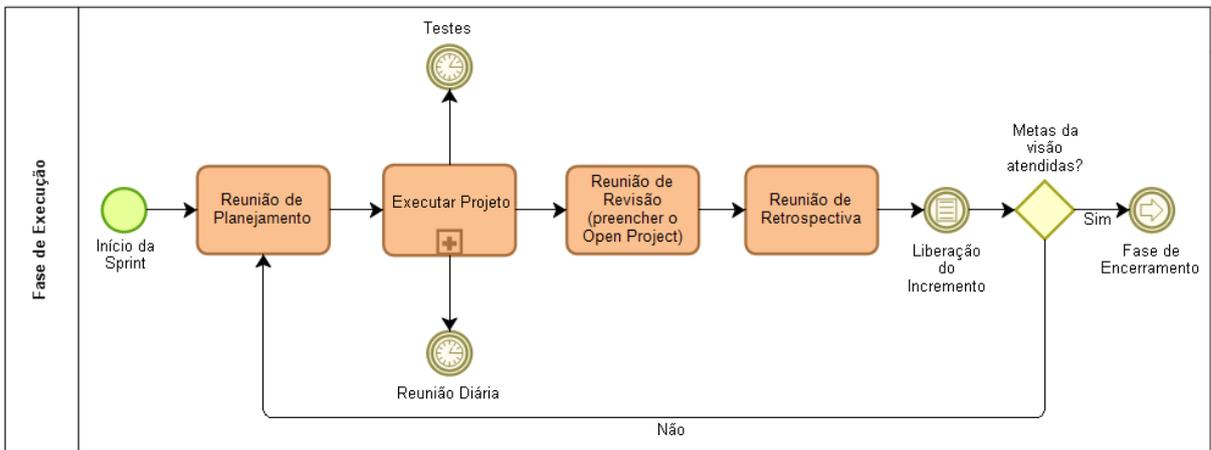
<b>6. Realizar reunião com o cliente</b>	
	<b>Objetivo:</b> Atualizar o cliente quanto às tarefas e estimativa de duração do projeto.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO;</li> <li>• Cliente.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS) atualizado;</li> <li>• OpenProject.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Reunir com o cliente;</li> <li>• Apresentar o cronograma e a lista de tarefas;</li> <li>• Alinhar a expectativa do cliente e da equipe.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Abertura de chamado GLPI.</li> </ul>	

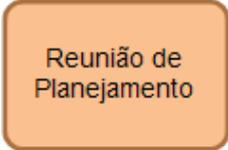
<b>7. Abrir Chamado GLPI</b>	
	<b>Objetivo:</b> Iniciar a execução do projeto.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO;</li> <li>• Cliente.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Documento de Visão do Software (DVS) atualizado.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• PO abre um chamado GLPI para o cliente com o DVS anexado;</li> <li>• Cliente aprova o chamado.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• GLPI aprovado;</li> <li>• Início de uma sprint.</li> </ul>	

### 4.3 Execução

A execução do projeto é realizada de acordo com a duração de cada sprint. Durante sua execução são realizadas reuniões diárias com o objetivo de identificar os impedimentos, avanços e performance do processo. Ao mesmo tempo testes são feitos para validar o sistema, analisando os erros e corrigindo-os. Enquanto todas as metas de visão não forem atendidas há necessidade de uma nova sprint para sua realização. Ao final de cada Sprint deve ser entregue todos os incrementos para o cliente, mostrando a evolução ao longo do projeto.

Todas as tarefas necessárias para implementar o software são divididas durante as Sprints, inclusive a criação do manual do software e a criação de vídeos ou documentos para o treinamento.



1. Reunião de Planejamento	
	<b>Objetivo:</b> Planejar o que será feito na Sprint.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• PO;</li> <li>• Scrum Master;</li> <li>• Equipe de Desenvolvimento.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Backlog do Produto;</li> <li>• OpenProject;</li> <li>• Chamado GLPI aprovado.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Reunião entre todos os membros do projeto;</li> <li>• Detalhar e planejar as atividades da Sprint;</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• OpenProject atualizado.</li> </ul>	

<b>2. Executar Projeto</b>	
	<b>Objetivo:</b> Executar a sprint.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>● Equipe de Desenvolvimento;</li> <li>● PO;</li> <li>● Scrum Master.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>● OpenProject;</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>● Mobilizar e instruir a equipe do projeto;</li> <li>● Realizar tarefas definidas pelo Backlog;</li> <li>● Realizar reuniões diárias;</li> <li>● Executar mudanças aprovadas;</li> <li>● Gerenciar os custos do projeto;</li> <li>● Gerar informações sobre o projeto;</li> <li>● Reunir as entregas geradas como resultado da execução das tarefas;</li> <li>● Criar documentos de treinamento (videoaula)</li> <li>● Realizar testes no software.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>● Liberação do incremento do software;</li> <li>● Cronograma do OpenProject atualizado;</li> </ul>	

### 3. Reunião de Revisão

Reunião de  
Revisão

**Objetivo:** Verificar com o cliente todo processo já realizado, ide(organização)ficando possíveis alterações e performance do projeto.

**Responsáveis:**

- PO;
- Scrum Master;
- Equipe de Desenvolvimento;
- Cliente.

**Entradas:**

- Liberação do incremento do software;

**Descrição das atividades:**

- Gerenciar as expectativas e os relacionamentos das partes interessadas;
- Gerar e comunicar as informações, de acordo com as especificações previstas no backlog;
- Documentar as alterações necessárias
- Incrementar o software.
- Repassar treinamentos do software;

**Saídas:**

- Entregas;
- Documento de treinamento;
- Relatório de Acompanhamento do Projeto (RAP).

#### 4. Reunião de Retrospectiva

Reunião de Retrospectiva

**Objetivo:** Documentar as lições aprendidas na Sprint e analisar o que pode ser melhorado para a próxima.

**Responsáveis:**

- PO;
- Scrum Master;
- Equipe de Desenvolvimento.

**Entradas:**

- Reunião de Revisão;
- Entregas;
- Planilha de Lições Aprendidas.

**Descrição das atividades:**

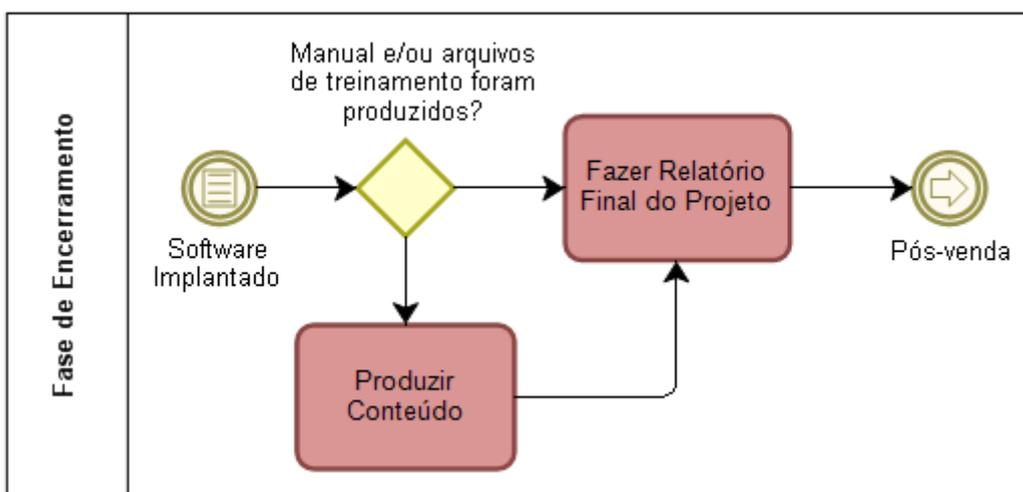
- Iniciar a reunião com a equipe do projeto na identificação de eventos positivos e negativos ocorridos durante o projeto;
- Cadastrar as ocorrências na Planilha de Lições Aprendidas;

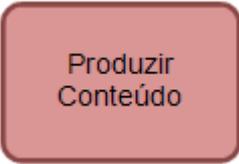
**Saídas:**

- Planilha de lições aprendidas.
- Software implantado;

#### 4.4 Encerramento

A fase de encerramento tem como objetivo, depois do software implantado, disponibilizar conteúdos de treinamentos para o cliente e partes interessadas. A solução proposta para esse processo é realizar uma sprint apenas para produção de videoaulas e disponibilizar ao final do projeto toda documentação necessária. Porém, caso não seja realizado em uma sprint, a equipe deve produzir o conteúdo durante a fase de encerramento. Ao final dessa fase, será produzido um Relatório Final, onde serão reunidas as informações essenciais do projeto para(organização)ndo uma cogestão eficiente.



<b>1. Produzir Conteúdo</b>	
	<p><b>Objetivo:</b> Construir conteúdo de treinamento de software, caso não realizado durante uma sprint.</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>• Equipe de Desenvolvimento.</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>• Software implantado;</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>• Elaborar manual do usuário;</li> <li>• Elaborar manual do sistema;</li> <li>• Documentar todo manual;</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>• Manual do software elaborado.</li> </ul>	

## 2. Fazer Relatório Final do Projeto

Fazer Relatório  
Final do Projeto

**Objetivo:** Documentar todas as tarefas, impedimentos e lições aprendidas durante o projeto para cogestão.

### Responsáveis:

- PO;
- Scrum Master;
- Equipe de Desenvolvimento.

### Entradas:

- Planilha de lições aprendidas;
- Relatório de Acompanhamento do Projeto;
- OpenProject.

### Descrição das atividades:

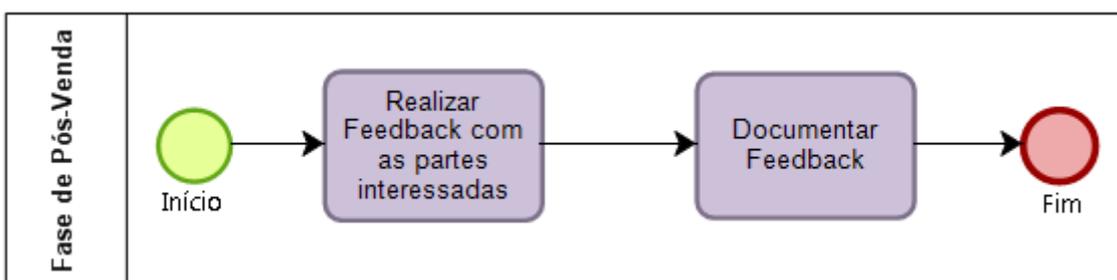
- Verificar se todos os serviços ou produtos contratados foram recebidos definitivamente, de acordo com os requisitos;
- Observar os termos e condições contratuais quanto a procedimentos específicos para encerramento;
- Preencher o Termo de Encerramento do Projeto (TEP);
- Reconhecer a equipe pela finalização do projeto;
- Consolidar todas as lições aprendidas na Base de Conhecimento de Lições Aprendidas e publicá-las de acordo com a política de comunicação da organização;
- Informar o encerramento dos contratos às partes interessadas.

### Saídas:

- Planilha de lições aprendidas atualizada;
- Termo de Encerramento do Projeto (TEP);
- Projeto encerrado.

## 4.5 Pós-venda

Na fase de pós-venda foi proposta a realização de um feedback com as partes interessadas a fim de avaliar o projeto desde seu planejamento até a entrega do software em funcionamento. A partir do feedback pode-se analisar se o software realmente está em atividade e em qual grau de qualidade e satisfação ele atendeu as partes interessadas. Além de disponibilizar, se necessário, possíveis mudanças no software durante a documentação do feedback.



<b>1. Realizar Feedback com as partes interessadas</b>	
	<b>Objetivo:</b> Mensurar a satisfação do cliente, conferir se o software está em produção e verificar possíveis alterações.
<b>Responsáveis:</b> <ul style="list-style-type: none"> <li>• Equipe pós-venda;</li> <li>• Cliente.</li> </ul>	
<b>Entradas:</b> <ul style="list-style-type: none"> <li>• Questionário de Feedback.</li> </ul>	
<b>Descrição das atividades:</b> <ul style="list-style-type: none"> <li>• Realizar reunião com o cliente;</li> <li>• Verificar a utilização do software;</li> <li>• Verificar possíveis alterações do software;</li> <li>• Aplicar questionário de satisfação.</li> </ul>	
<b>Saídas:</b> <ul style="list-style-type: none"> <li>• Questionário de feedback preenchido.</li> </ul>	

## 2. Documentar Feedback

Documentar  
Feedback

**Objetivo:** Documentar o feedback para cogestão e conhecimento do Gerente de Projetos.

### Responsáveis:

- Equipe pós-venda;
- Gerente de Projetos.

### Entradas:

- Questionário de feedback preenchido.

### Descrição das atividades:

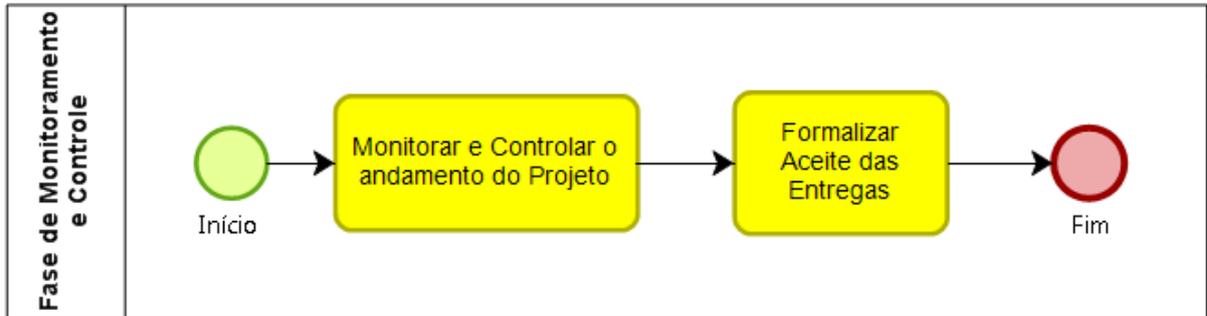
- Repassar para o Gerente de Projetos o feedback realizado;
- Anexar o questionário no OpenProject;
- Realizar alterações no software.

### Saídas:

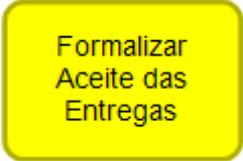
- OpenProject atualizado;
- Software atualizado;

## 4.6 Monitoramento e Controle

A fase de monitoramento tem como objetivo controlar todo o projeto desde seu planejamento até a fase de pós-venda para verificar se os processos de entrada e saída condizem com o que foi proposto na elaboração do projeto. Além de gerenciar as tarefas e artefatos proporcionando o melhor resultado possível para a execução do projeto.



<b>1. Monitorar e Controlar o andamento do Projeto</b>	
	<p><b>Objetivo:</b> Acompanhar a execução do projeto identificando possíveis desvios em relação ao que foi planejado.</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>● Gerente de Projetos</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>● Cronograma;</li> <li>● Análise de Viabilidade.</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>● Monitorar e controlar cronograma;</li> <li>● Monitorar e controlar custos;</li> <li>● Monitorar e controlar qualidade;</li> <li>● Monitorar os riscos;</li> <li>● Identificar, analisar e acompanhar novos riscos e elaborar seus respectivos planos de respostas na Planilha de Riscos;</li> <li>● Gerar Relatório de Acompanhamento do Projeto (RAP), comparando o desempenho real com o planejado;</li> <li>● Determinar as ações corretivas ou preventivas necessárias.</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>● Relatório de Acompanhamento do Projeto (RAP);</li> <li>● OpenProject atualizado.</li> </ul>	
<b>2. Formalizar Aceite das Entregas</b>	

 <p>Formalizar Aceite das Entregas</p>	<p><b>Objetivo:</b> Realizar formalmente o aceite de cada entrega do Projeto.</p>
<p><b>Responsáveis:</b></p> <ul style="list-style-type: none"> <li>● Gerente de Projetos</li> </ul>	
<p><b>Entradas:</b></p> <ul style="list-style-type: none"> <li>● OpenProject;</li> </ul>	
<p><b>Descrição das atividades:</b></p> <ul style="list-style-type: none"> <li>● Mobilizar e instruir a equipe do projeto;</li> <li>● Realizar tarefas definidas pelo Backlog;</li> <li>● Realizar reuniões diárias;</li> <li>● Executar mudanças aprovadas;</li> <li>● Gerenciar os custos do projeto;</li> <li>● Gerar informações sobre o projeto;</li> <li>● Reunir as entregas geradas como resultado da execução das tarefas;</li> <li>● Criar documentos de treinamento (videoaula)</li> <li>● Realizar testes no software.</li> </ul>	
<p><b>Saídas:</b></p> <ul style="list-style-type: none"> <li>● Liberação do incremento do software;</li> <li>● Cronograma do OpenProject atualizado.</li> </ul>	