

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA  
BACHARELADO EM ENGENHARIA DE SOFTWARE**

**BRUNO D'LUCCA SILVA CARVALHO  
MARCELLA CANEDO TRISTÃO**

**APLICAÇÃO DAS TÉCNICAS DE LEAN INCEPTION E MVP NO PROCESSO DE UMA  
FÁBRICA DE SOFTWARE ACADÊMICA**

**ANÁPOLIS - GO  
2021-01**

**BRUNO D'LUCCA SILVA CARVALHO  
MARCELLA CANEDO TRISTÃO**

**APLICAÇÃO DAS TÉCNICAS DE LEAN INCEPTION E MVP NO PROCESSO DE UMA  
FÁBRICA DE SOFTWARE ACADÊMICA**

Trabalho de Conclusão de Curso I apresentado como requisito parcial para a conclusão da disciplina de Trabalho de Conclusão de Curso I do curso de Bacharelado em Engenharia de Software do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Prof. Me. Luciana Nishi.

**ANÁPOLIS - GO  
2021-01**

**BRUNO D'LUCCA SILVA CARVALHO**  
**MARCELLA CANEDO TRISTÃO**

**APLICAÇÃO DAS TÉCNICAS DE LEAN INCEPTION E MVP NO PROCESSO DE UMA  
FÁBRICA DE SOFTWARE ACADÊMICA**

Trabalho de Conclusão de Curso I apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Software do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em [dia] de [mês] de 2021, composta por:

---

Prof. Luciana Nishi  
Orientador

---

Prof. [nome do professor]

---

Prof. [nome do professor]

## Resumo

O desenvolvimento de produtos de *software* vem crescendo constantemente nas últimas décadas, tendo um aumento exponencial nos últimos anos. Mas ainda hoje é possível observar alguns dos problemas que perduram desde a crise do *software*, como exemplo a falta de definição de escopo, inconformidade dos produtos com as expectativas do cliente e entregas não frequentes, situação que também foi observada no ambiente da Fábrica de Tecnologias Turing (FTT). Tendo esse cenário como base, estabeleceu-se um estudo de caso para entender os problemas que causam isso dentro do ambiente da FTT para assim elaborar uma proposta de processo incluindo práticas da *Lean Inception* e MVP com o objetivo de mitigá-los. Espera-se que a aplicação deste processo possa contribuir para a produção de produtos de *software* com melhores níveis de conformidade e entregas de valor contínuas dentro do ambiente de desenvolvimento de *software* acadêmico.

**Palavras-chave:** Processo de desenvolvimento de *Software*; *Lean Inception*; MVP.

## Sumário

1.	<b>PROBLEMA</b> .....	6
2.	<b>OBJETIVOS</b> .....	7
2.1	<b>Objetivo Geral</b> .....	7
2.2	<b>Objetivos Específicos</b> .....	7
3.	<b>JUSTIFICATIVA</b> .....	8
4.	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	9
4.1	<b>Fábrica de Tecnologias Turing</b> .....	9
4.2	<i>Framework Scrum</i> .....	9
4.3	<i>OpenUP</i> .....	11
4.3.1	<b>DEFINIÇÃO DE PAPÉIS</b> .....	11
4.4	<b>Metodologia Híbrida</b> .....	12
4.5	<i>Minimum Viable Product (MVP)</i> .....	13
4.6	<i>Lean Inception</i> .....	14
5.	<b>METODOLOGIA</b> .....	16
6.	<b>CRONOGRAMA</b> .....	17
7.	<b>RESULTADOS ALCANÇADOS</b> .....	18
8.	<b>RESULTADOS ESPERADOS</b> .....	20
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	21

## **Lista de Figuras**

Figura 1 - Processo da Fábrica de Tecnologias Turing .....	13
--	----

## 1. Problema

A velocidade e o dinamismo que novas tecnologias trazem ao mundo, tornaram necessárias novas técnicas na concepção e desenvolvimento de modelos de negócios que aceleram o processo, desde a contextualização da ideia até a chegada do produto no mercado (Ries, 2012).

Nobel (2011) ressalta que grande parte das empresas de *software* falham em seus projetos pois acabam desperdiçando tempo e dinheiro no desenvolvimento do produto de *software* errado, ou seja, que não atende às reais necessidades dos *stakeholders* e estes só percebem tardiamente como realmente o produto deveria se comportar para atender as demandas dos clientes.

Os diversos problemas ocasionados pela definição errônea de um processo de desenvolvimento de um produto também são encontrados nas fábricas de *software*, grande parte destes problemas provém da falta de uma metodologia organizacional bem definida voltada para o desenvolvimento do produto focando principalmente nas fases iniciais de desenvolvimento do projeto, o que acaba ocasionando o desenvolvimento de um produto errado o qual não atende a demanda do cliente (Maximiliano, 2010).

De acordo com Nagarajan (2020), uma das falhas que acontecem frequentemente no desenvolvimento de sistemas de *software* é a falta de compreensão das reais necessidades do negócio. Para a iniciação de um projeto é necessário haver um mapeamento preciso de recursos e funções para suprir as necessidades do negócio, ou seja, a definição da visão de um projeto, essa a qual pode ocasionar o seu sucesso se bem definida ou seu fracasso caso for mal concebida.

Considerando as informações apresentadas anteriormente, como os processos de definição de visão, de escopo e das entregas dos produtos, poderiam ser otimizados dentro de um ambiente acadêmico de desenvolvimento de *software*?

## 2. Objetivos

### 2.1 Objetivo Geral

Agilizar e prover mais assertividade na definição de visão, escopo e entregas do produto, aumentando a quantidade de entregas e sua cadência.

### 2.2 Objetivos Específicos

- Analisar o ambiente da Fábrica de Tecnologias Turing e seus projetos vigentes;
- Analisar práticas que auxiliem na definição do produto;
- Analisar e coletar informações sobre o processo atual da Fábrica de Tecnologias Turing;
- Identificar pontos de melhoria no processo, com foco na definição de visão, escopo e entregas dos produtos;
- Propor melhorias no processo utilizando práticas da *Lean Inception* e MVP;
- Aplicação do subprocesso de definição de visão e MVP ao processo atual da FTT;
- Analisar os resultados da aplicação do processo proposto.

### 3. Justificativa

O mercado tem tido um índice de crescimento acelerado, portanto é necessário que o desenvolvimento de empresas e produtos seja cada vez mais rápido para atender todas as demandas e necessidades dos clientes. Consoante a Tachizawa (2006) algumas características presentes neste novo mercado são as mudanças rápidas nas exigências do consumidor em relação a produtos e serviços, demanda crescente do consumidor por produtos e serviços de qualidade, alteração no poder de compra da população, escassez de insumos produtivos e recursos críticos, alterações tecnológicas crescentes, escassez de determinadas habilidades e alterações no ritmo e na natureza das mudanças sociais.

A metodologia do *Lean Startup* foi idealizada por Ries (2012), que aborda técnicas de gestão para que startups possam implantar uma cultura de aprendizagem validada, desde ideias de desenvolvimento de produto ou serviço, testes de hipóteses baseados em um MVP e finalmente, o lançamento do produto no mercado, com objetivo de evitar realizar os processos de forma equivocada, não fracassar e ter sucesso nas operações desenvolvidas, poupando tempo e dinheiro.

O autor também comenta sobre a importância de construir o mais rápido possível o Produto Mínimo Viável, para que este seja tratado com maior ênfase na sequência referencial, sendo assim este produto “teste”, tenha a capacidade de medir seus efeitos através da perspectiva dos clientes, e com a utilização de métricas, e com o resultado destas aprimorar o produto teste para que ele fique a nível da perspectiva do usuário.

No ambiente da FTT, é utilizada uma metodologia híbrida, composta pelas metodologias *Scrum* e *OpenUp*, utilizando práticas, métodos e artefatos que melhor se adequam ao modelo de trabalho da FTT. Porém, mesmo com uma metodologia e um processo bem estruturados, pode-se perceber que há algumas dificuldades quanto à definição da visão, escopo, e de entregas dos projetos, e assim a fase de iniciação dos projetos tem a sua duração estendida, as entregas são grandes e espaçadas ao invés de pequenas e frequentes, o que dificulta o aprendizado contínuo do negócio do cliente para melhor adaptação do *software* nas entregas seguintes.

Com base nas informações apresentadas, este trabalho propõe-se a investigar e analisar o ambiente da FTT, o processo utilizado e o fluxo de trabalho dos seus membros, visando identificar os pontos de melhoria e realizar um estudo sobre outras metodologias e técnicas para identificar práticas que poderiam melhorar os pontos identificados.

## 4. Fundamentação Teórica

Este capítulo aborda os conceitos relacionados a uma fábrica de software e sobre as metodologias ágeis utilizadas nos processos das mesmas. A seção 4.1 apresenta a unidade dos cursos bacharelados em computação (FTT) e as seguintes abordam sobre as metodologias de desenvolvimento de software ágil.

### 4.1 Fábrica de Tecnologias Turing

Consoante a Fernandes (2014), uma Fábrica de *Software* é constituída de um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, visando a geração de produtos de *software* conforme os requerimentos documentados de todas as partes interessadas, da forma mais produtiva e econômica.

A Fábrica de Tecnologias Turing (FTT), unidade dos cursos bacharelados em computação, é uma fábrica de *software* acadêmica com sua estruturação dividida em três principais núcleos (Fábrica de *software*, Núcleo de capacitação e Núcleo de pesquisa), seu principal foco é formar acadêmicos com uma visão mercadológica, isto é, com um perfil humano, criativo e reflexivo, com experiência e conhecimentos necessários para o mercado de trabalho (UNIEVANGÉLICA, 2017).

### 4.2 *Framework Scrum*

O *Scrum* é um *framework* que auxilia na gestão de equipes e projetos ágeis. Ele não define um regras ou atividades engessadas que obrigatoriamente devem ser seguidas à risca para o sucesso de um projeto, mas provê uma estrutura básica na qual podem ser incorporadas vários métodos e técnicas diferentes que adaptem melhor a realidade do ambiente em que estão sendo aplicadas (SCHWABBER e SUTHERLAND, 2020).

Ainda de acordo com o autor, a teoria do *Scrum* se baseia no empirismo, conceito afirma que o conhecimento vem das experiências e vivências dos indivíduos fazendo assim com que as decisões sejam tomadas de acordo com o que já foi observado e vivenciado, e no *Lean Thinking* (Pensamento Enxuto), que foca em reduzir o desperdício e focar naquilo que é essencial.

Segundo Schwabber e Sutherland (2020), o *Scrum* tem três pilares fundamentais, sendo eles a transparência, inspeção e adaptação. A transparência diz respeito à disponibilidade e

esclarecimento de toda informação relacionada aos projetos a todos os envolvidos. Em inspeção a inspeção, todos os artefatos devem ser verificados frequentemente, para identificar quaisquer possíveis inconformidades. A adaptação diz respeito às mudanças que devem ser feitas sempre que variações fora do limite do aceitável forem identificadas.

Os artefatos do *Scrum* incluem o *Product Backlog*, que é uma lista priorizada dos itens que devem ser desenvolvidos para construir o produto e representa a meta do produto, o *Sprint Backlog*, que é uma lista priorizada dos itens do *Product Backlog* que serão desenvolvidos em uma *Sprint*, e por fim, a Definição de Pronto, que deve estar clara a todos e serve como base para verificar se a completude do incremento (SCHWABBER e SUTHERLAND, 2020).

O *Scrum* possui três papéis. O *Scrum Master*, responsável por assegurar que todos da equipe entendam, respeitem e pratiquem todos os pilares, valores, artefatos e eventos do *Scrum*. O *Product Owner*, que é responsável por entender e negociar as funcionalidades com o cliente, priorizar o *Product Backlog* do produto e esclarecer quaisquer dúvidas sobre o objetivo do produto e suas funcionalidades para a equipe. Os *Developers* são aqueles que participam diretamente da construção dos incrementos do produto (SCHWABBER e SUTHERLAND, 2020).

Além disso, o *Scrum* possui cinco eventos que compõem o seu processo. A *Sprint* é um período de um mês ou menos em que um incremento do produto será desenvolvido. Após o início do projeto, as *Sprints* se iniciam ao término da anterior, sem pausa entre elas. A *Sprint* é um evento “*container*”, dentro do qual todos os outros eventos do *Scrum* acontecem. A *Sprint Planning* é o evento que dá início a *Sprint*. Nesta reunião são planejadas todas as atividades para a *Sprint* de forma colaborativa por toda a equipe *Scrum* (SCHWABBER e SUTHERLAND, 2020).

O produto da *Sprint Planning* é o *Sprint Backlog*. A *Daily Scrum* é uma reunião que acontece em todos os dias úteis da *Sprint*, e tem como objetivo inspecionar o progresso da equipe em relação ao *Sprint Backlog*, adaptá-lo se necessário, e serve para que os desenvolvedores ajustem o seu plano de trabalho, não sendo o único momento em que isso pode ser feito. Essas reuniões têm como objetivo melhorar a comunicação da equipe, identificar impedimentos e tomar decisões de forma rápida, eliminando a necessidade de outras reuniões. A *Sprint Review* é o penúltimo evento, e é nessa reunião que o resultado do trabalho da *Sprint* é apresentado aos *stakeholders* (partes interessadas, como cliente e usuários finais) e inspecionado. Após a revisão do trabalho realizado, a equipe discute para decidir o que fazer em seguida. A *Sprint Retrospective* é o último evento, e nesta reunião a equipe *Scrum* analisa a *Sprint* com foco nos indivíduos, suas interações e o processo em geral, para identificar desvios

e decidir quais ações serão tomadas para eliminá-los na próxima *Sprint* (SCHWABBER e SUTHERLAND, 2020).

### 4.3 *OpenUP*

*OpenUp* é um processo de desenvolvimento de *software* minimamente suficiente, o que significa que apenas o que é necessário está incluso no processo. No entanto, este processo é completo no sentido de que pode ser utilizado como processo interno para construir um sistema. É um processo ágil e aplica abordagens iterativas e incrementais com um ciclo de vida estruturado, valorizando a colaboração da equipe e os benefícios para as partes interessadas em relação a entregas e formalidades (ECLIPSE FOUNDATION, 2019).

#### 4.3.1 Definição de Papéis

De acordo com BALDUÍNO (2007), a habilidade essencial necessária aos pequenos times co-aloçados é representada pelos seguintes papéis do *OpenUp*:

- **Stakeholder:** Representa todas as partes interessadas, suas necessidades devem ser satisfeitas pelo projeto desenvolvido. Este papel pode ser executado por pessoas que serão afetadas materialmente pelo resultado do desenvolvimento do *software*;
- **Analista:** Este representa o cliente e o usuário final no que se diz a respeito da obtenção de entradas dos *stakeholders* para entender o problema a ser resolvido e pela captura das prioridades dos requisitos;
- **Arquiteto:** É o componente da equipe responsável por projetar a arquitetura, que inclui as tomadas de decisões chave que definem o design e implementação do projeto;
- **Desenvolvedor:** Componente da equipe responsável por desenvolver parte do sistema de acordo com a arquitetura definida;
- **Testador:** É responsável por realizar as atividades referentes aos testes, bem como o registro de entrada e saída coletados a partir dos resultados;
- **Gerente de projetos:** Lidera o planejamento do projeto mantendo o time focado em atingir os objetivos do projeto;
- **Qualquer papel:** É um papel que pode ser representado por qualquer integrante dentro do time.

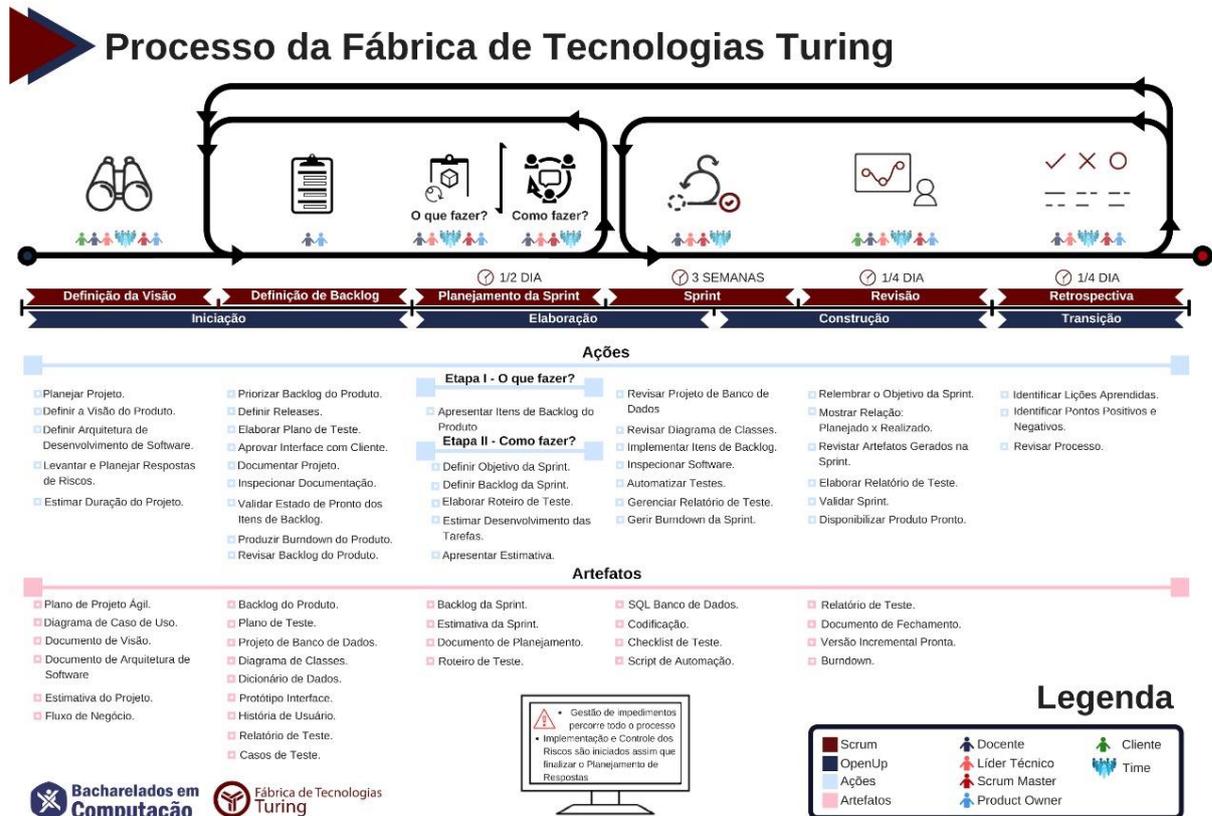
## 4.4 Metodologia Híbrida

Segundo Pressman e Maxim (2016), a base da engenharia de *software* é a camada de processo. O processo define uma metodologia que deve ser estabelecida para a entrega efetiva da tecnologia de engenharia de *software*. O processo de *software* provê uma base para o controle da gestão de projetos de *software* e estabelece o contexto no qual os métodos técnicos são aplicados, artefatos são produzidos (modelos, documentos, dados, relatórios, formulários, etc.), objetivos são estabelecidos, qualidade é garantida e as mudanças são gerenciadas apropriadamente.

Ainda segundo os autores, processo de software é um conjunto de atividades, ações e tarefas realizadas na criação de algum artefato. É uma abordagem adaptativa que permite que as pessoas façam o trabalho da melhor maneira, escolhendo o conjunto apropriado de ações para que o *software* seja entregue no prazo e com qualidade suficiente para satisfazer as pessoas que estão patrocinando o desenvolvimento e aquelas que vão utilizá-lo posteriormente.

A Fábrica de Tecnologias Turing trabalha com a metodologia de desenvolvimento híbrido, composto pelo Scrum com foco no gerenciamento das equipes e de processos, e o *OpenUp* que aplica o desenvolvimento de *software* de modo iterativo e incremental ao processo.

Figura 1 - Processo da Fábrica de Tecnologias Turing



Fonte: Fabrica de Tecnologias Turing 2019

O *Scrum* no âmbito da FTT é o *framework* responsável pelo processo de produção de sistemas garantindo a transparência, inspeção e adaptação entre todas as partes interessadas. Os projetos são afetados por restrições de tempo, custo, escopo, qualidade, recursos, capacidade de organização, dentre outras limitações (SBOK GUIDE, 2016).

O *OpenUP* é um processo iterativo de desenvolvimento de *software* mínimo, completo e extensível, tratando da organização do trabalho pessoal, das equipes e das partes interessadas no visando integrar valor incremental às partes interessadas de maneira passível (ECLIPSE FOUNDATION, 2019).

#### 4.5 Minimum Viable Product (MVP)

A sociedade contemporânea dispõe de um conjunto comprovado de técnicas visando administrar grandes empresas e conhecimento em melhores práticas para construir projetos reais. De acordo com Caroli (2018), MVP é uma maneira de desenvolver produtos entregando o mínimo necessário para sua utilização, diminuindo gastos e permitindo que o produto seja

validado e incrementado. Cada incremento também é um MVP, sendo assim, traz uma evolução em relação à versão anterior juntamente com os ajustes advindos do *feedback* obtido.

O MVP é uma versão do produto que permite produzir um artefato contendo o mínimo de esforço e tempo empregado. De certa forma, criar um MVP exige que seja realizado um trabalho extra: ter a capacidade de medir o impacto do mesmo. Não é adequado construir um protótipo para que ele seja avaliado por engenheiros e designers em função da qualidade interna, também é necessário inseri-lo diante dos possíveis clientes para avaliar se o protótipo feito atenderá as necessidades (RIES, Eric, 2012, p.58).

Pode-se perceber uma semelhança entre o *Scrum* e o MVP no quesito de entregas de valor incrementais, e que esse conceito de entregas mínimas, evitando o desperdício investindo em recurso para o produto sem haver tanto conhecimento sobre o negócio e possibilitando validar o que foi produzido com essas entregas, aprender com base nisso e assim evoluir o produto de forma incremental, entra em consonância com os princípios da teoria do *Scrum*, o empirismo e o *Lean Thinking*.

#### **4.6 *Lean Inception***

Segundo Carolli (2018), a *Lean Inception* é utilizada quando o time precisa desenvolver um MVP e criar um produto de forma iterativa e incremental. Sendo assim a *Lean Inception* propõe-se que em uma única semana de trabalho colaborativo a equipe como um todo entenda os objetivos do produto, os principais usuários e o escopo funcional de alto nível tal que a duração do projeto possa ser estimada de forma mais eficiente e uma estratégia de lançamento incremental de MVP's possa ser identificada.

“A *Lean Inception* propõe um processo colaborativo de descoberta e esclarecimento no qual as pessoas envolvidas trabalhem juntas em uma sequência de atividades para compreender opções e elaborar o MVP.” (CAROLI, Paulo, 2018, p.50).

O autor também ressalta que o *workshop Lean Inception* direciona os participantes a entender e planejar a entrega incremental do produto, focando o MVP, objetivando a entrega da funcionalidade para que assim o *feedback* das partes interessadas seja coletado.

“A metodologia *Scrum* não tem a parte de desenvolvimento de pessoas e estímulo à criatividade embora seja eficaz na parte de entregas no prazo, porém o *Lean Inception*, por outro lado, promove diversas técnicas, como desenvolvimento de pessoas, trabalho de estimulação da criatividade, por meio de ciclos passando por fases onde há a participação de todos os envolvidos no projeto, cada fase tem uma forma diferente de *brainstorm* até chegar a um produto mínimo viável.” (BRAGA, 2020).

Ainda de acordo com Braga et al. (2020), com base em seus estudos, é comprovado que se tratando da criação de um produto mínimo viável para o usuário, a metodologia *Lean Inception* tem um resultado mais rápido, principalmente pelo fato de que a ideia é desenvolver o MVP o mais rápido possível e agregar valor ao produto ao longo do tempo ao produto de acordo com as solicitações e necessidades do usuário.

## 5. Metodologia

A natureza desta pesquisa melhor se adequou à abordagem descritiva pois esta visa o estudo das características de um fato e possui seu foco no envolvimento de técnicas padronizadas de análise como a elaboração de questionários e observação sistemática (GIL, 2010).

Consoante a pesquisa escolhida, pretende-se realizar um estudo das abordagens e do atual processo implementado na Fábrica de Tecnologias Turing com o objetivo consolidar uma base de conhecimento ampla para que seja realizada a integração de um subprocesso referente a definição da visão de *software* e visando a agilidade nos entregáveis do projeto. Em vista disso será feita uma comparação entre o atual processo e os demais já existentes a fim de complementar um processo de desenvolvimento de *software* ágil.

No primeiro momento, para analisar a viabilidade da implementação de um subprocesso, pretende-se realizar uma análise do funcionamento da empresa, o processo referente a cada equipe, indicadores de produtividade e relatórios das últimas releases realizadas através da observação participante, método de pesquisa em que se assume uma posição totalmente ativa e envolve-se com o fenômeno analisado, neste trabalho, como participante total (DUARTE, 2021). Propõe-se também a realização de reuniões e entrevistas para entender melhor quais são os principais desafios enfrentados pelos membros da equipe, juntamente com as todas as partes interessadas no projeto.

O segundo passo com os dados das análises realizadas em mãos consiste em estudar a metodologia *Lean Inception* e os conceitos de MVP, de modo a verificar se suas práticas e técnicas podem ser parcialmente ou totalmente incorporadas ao processo vigente da FTT como um subprocesso de definição de visão incluindo a *Lean Inception* e o MVP na sua metodologia híbrida.

## 6. Cronograma

ATIVIDADE	2021										2021									
	Fev		Mar		Abr		Mai		Jun		Jul		Ago		Setr		Out		Nov	
	1 a	2 a																		
Estudos sobre Lean Inception	X																			
Estudos sobre os processos da FTT	X																			
Adequações e revisão		X																		
Estruturação da proposta de processo				X	X	X														
Análise do Ambiente FTT	X	X																		
Estudo dos processos da FTT	X	X	X																	
Adequação de métodos e processos vigentes em relação ao processo do Lean Inception							X	X												
Implantação do processo do Lean Inception									X	X	X									
Análise da resposta do ambiente									X	X										
Adequações											X	X								
Implementação das adequações												X								
Análise da resposta do ambiente													X	X						
Considerações finais															X					

## 7. Resultados Alcançados

Segundo as observações diretas realizadas por parte dos próprios autores do trabalho, uma vez em que ambos ocupavam o cargo de Product Owner dos projetos analisados neste trabalho, percebeu-se que há dificuldade da definição de visão, de escopo e das entregas dos produtos de software, o acaba ocasionando a ausência de entregas constantes fazendo com que a obtenção de feedback do cliente ocorra com pouca frequência.

Ao executar a análise do ambiente da FTT e realizando uma pesquisa nos dados históricos observou-se que haviam dois projetos sendo desenvolvidos, os sistemas SeIntegra e Virtoo. O projeto SeIntegra teve seu início como um “Piloto”, onde os alunos dos primeiros períodos, sob orientação do líder técnico e dos professores dos cursos bacharelados em computação, deveriam tomar todas as decisões acerca do projeto a fim de aproximá-los da realidade do mercado de trabalho. O projeto Piloto teve início em agosto de 2018, e em novembro teve seu escopo aumentado e passou a se chamar SeIntegra.

Através da análise de dados obtidos dos repositórios e pela vivência dos autores com os projetos durante o período de observação, a primeira sprint se iniciou somente 5 meses depois do início do projeto, tendo um período de definição de visão bastante prolongado. O primeiro contato dos stakeholders com o SeIntegra foi em uma apresentação no início do ano de 2020, na qual foi possível fazer a coleta de *feedback*. A primeira entrega do SeIntegra ocorreu no dia 22/09/2020, mais de dois anos após o início do projeto. Alguns fatores que contribuíram com isso foram:

- Inexperiência dos alunos, pois todos estavam tendo a sua primeira experiência em uma fábrica de *software*;
- Falta de conhecimento de práticas e processos de software, uma vez que os alunos participantes do projeto estavam nos primeiros períodos do curso;
- Falta de domínio dos métodos e *frameworks* utilizados na FTT, como Scrum, *OpenUP*, UML, BPMN 2.0.

Todos esses fatores ocasionaram na necessidade de estudo por parte dos alunos, sendo assim, toda a fase de iniciação do processo ocorreu em paralelo à uma fase de estudos e capacitação dos alunos, justificando a duração prolongada da fase de iniciação do projeto. Os alunos eram constantemente orientados pelo líder técnico da FTT e professores do curso e estimulados a buscar conhecimento e encontrar as melhores soluções para a resolução dos problemas, porém, todo o seu trabalho era supervisionado e revisado pelos orientadores.

O sistema Virtoo, é um projeto desenvolvido para um cliente internacional. Além dos desafios que existem no ambiente da FTT como a falta de experiência e alta rotatividade dos integrantes, há também dificuldade quanto à obtenção de *feedback* dos clientes. O período entre solicitação e a recepção de informações do cliente é grande, o que têm ocasionado em entregas que não atendem às suas demandas, diminuição do ritmo de trabalho ou até mesmo a paralisação das atividades do projeto. No entanto, pode-se observar um aumento de *feedback* no período em que há uma entrega do sistema.

A concepção do projeto Virtoo se deu início em meados de agosto de 2015, onde por intermédio do Núcleo de Assuntos Internacionais (NAI), onde o diretor do setor administrativo e financeiro do Instituto Politécnico Superior VIDA (ISPVIDA) realizou uma visita ao Centro Universitário Unievangélica, e nesta ocasião foram apresentados as necessidades referentes à gestão acadêmica e financeira dos institutos ISPVIDA e o Instituto Superior de Teologia em Lubango (ISTEL), ambos mantidos pela Aliança Evangélica da Angola. E deste encontro vislumbrou-se a possibilidade de uma parceria para o desenvolvimento de um *software* de gestão acadêmica e financeira para ambas instituições (UNIEVANGÉLICA, 2018).

Para a elicitação de requisitos referentes à estruturação do projeto foi realizada uma visita técnica às duas instituições, no período de 25/04/2016 até 28/04/2016. O projeto, que recebeu o nome de Virtoo, teve seu início em agosto de 2016 e seu planejamento foi estimado para ser realizado em módulos de entregas incrementais, as quais deveriam ser realizadas em período de 6 a 9 meses (UNIEVANGÉLICA, 2018).

O início do processo de observação se deu junto ao início do projeto SeIntegra, momento em que o projeto Virtoo já estava na fase de desenvolvimento. Sendo assim, não foi possível obter informações detalhadas sobre como se deu a fase de iniciação do Virtoo através de observação, apenas através de alguns relatos de membros da FTT ainda ativos e de alguns documentos disponibilizados pela própria instituição.

## 8. Resultados Esperados

Espera-se que este trabalho possa contribuir com a FTT através da criação e integração de um subprocesso focado na fase de iniciação ao processo de software atual, estabelecer práticas que auxiliam e aceleram a definição de visão e escopo de projetos, diminuir a fase de iniciação e possibilitar que a produção de artefatos como documentos de especificação de requisitos, diagramas UML, e mesmo código, tenha seu início mais cedo no projeto.

Implementar o conceito de MVP na definição das entregas, para que o primeiro contato dos clientes com o produto de *software* ocorra em um período menor do que a média atual e cada incremento seja planejado de modo a agregar o máximo de valor com o mínimo de esforço da equipe. Com isso, possibilitar a obtenção de feedback contínuo, para que a equipe possa aprender mais sobre o negócio e o produto e adaptá-lo de forma iterativa e incremental ao longo das releases do projeto. Também se espera que o produto desta pesquisa possa servir como base para trabalhos futuros.

## Referências Bibliográficas

- BALDUINO, Ricardo. **Introduction to OpenUP (Open Unified Process)**. 2007. Disponível em: <<http://www.eclipse.org/epf/general/OpenUP.pdf>>
- BRAGA, Igor; NOGUEIRA, Marcelo; SANTOS, Nuno; MACHADO, Ricardo J. **Does the Lean Inception Methodology Contribute to the Software Project Initiation Phase?**. Computational Science and Its Applications – ICCSA 2020: 20th International Conference, Cagliari, Itália, ano 2020, p. 741-752, 4 jul. 2020. *E-book* (1026 p.).
- CAROLI, Paulo; **Lean Inception: Como alinhar pessoas e construir o produto certo**. São Paulo: Editora Caroli, 2018.
- DUARTE, Vânia; **Tipos de observação segundo critérios específicos**. 2021. Disponível em <<https://monografias.brasilecola.uol.com.br/regras-abnt/tipos-observacao-segundo-criterios-especificos.htm>>.
- ECLIPSE Foundation; **Enabling Open Innovation & Collaboration**. OpenUp Vision. 2020 Disponível em: <[https://www.eclipse.org/epf/openup\\_component/openup\\_vision.php](https://www.eclipse.org/epf/openup_component/openup_vision.php)>.
- FERNANDES, A. Aguinaldo. **Afinal, O Que É Uma Fábrica de Software?** Apresentação Power Point. 2005. Disponível em: <<https://slideplayer.com.br/slide/12849126/>>
- GIL, A.C; **Como elaborar projetos de pesquisa**. 5ed. São Paulo: Atlas, 2010. P. 1-176
- NAGARAJAN, Amith; **14 Common reasons software projects fail (And How To Avoid Them)**. Forbes Technology Council. 2020. Disponível em: <<https://www.forbes.com/sites/forbestechcouncil/2020/03/31/14-common-reasons-software-projects-fail-and-how-to-avoid-them/?sh=3b41ba8b798c>>
- PRESSMAN, Roger S.; MAXIM, Bruce R. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Londres: AMGH, 2016. 968 p. ISBN 978-8580555332.
- RIES, Eric; **A Startup Enxuta: Como os Empreendedores Atuais Utilizam a Invocação Contínua para Criar Empresas Extremamente Bem-sucedidas**. Edição. São Paulo: Lua de Papel, 2012. p. 1-210.
- SCHWABER, Ken; SUTHERLAND, Jeff. **The scrum guide: The Definitive Guide to Scrum: The Rules of the Game**. 2020 p. 1-14. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>>.
- SCRUMSTUDY. **Um guia para o conhecimento scrum (Guia SBOK): Um guia completo para entregar projetos utilizando scrum**. 3. ed. p 1 - 403.
- TACHIZAWA, Takeshi. **Gestão ambiental e responsabilidade social corporativa**. São Paulo: Atlas, 4. ed., 2006.
- UNIEVANGELICA, C. U. **Engenharia de Software: Diferenciais**. 2017. Disponível em:<<http://www.unievangelica.edu.br/curso.engenharia.software/diferenciais>>.

UNIEVANGELICA, C. U. **Projeto Pedagógico de Curso: Engenharia de Software.** 2018, Disponível em: <<http://www.unievangelica.edu.br/files/images/aaappcs/PPC%20Engenharia%20de%20Software.pdf>>

ZELKWITZ, M.V., Wallace, D. **Experimental models for validating computer technology.** IEEE Comput. 1998.