

**UNIVERSIDADE EVANGÉLICA DE GOIÁS – UniEVANGÉLICA**  
**BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**IMPLEMENTAÇÃO DE UM *FRAMEWORK* PARA IDENTIFICAÇÃO DE VALOR DO  
PRODUTO EM FÁBRICA DE *SOFTWARE* ACADÊMICA**

**LEONARDO DUARTE AMORIM**

**ANÁPOLIS**  
**2021**

**Leonardo Duarte Amorim**

**IMPLEMENTAÇÃO DE UM *FRAMEWORK* PARA IDENTIFICAÇÃO DE VALOR DO  
PRODUTO EM 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 Computação da Universidade Evangélica de Goiás – UniEVANGÉLICA.

Orientador(a): Prof. Me. Walquíria Fernandes Marins.

Anápolis  
2021

**LEONARDO DUARTE AMORIM**

**IMPLEMENTAÇÃO DE UM *FRAMEWORK* PARA IDENTIFICAÇÃO DE VALOR DO  
PRODUTO EM 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  
Computação da Universidade Evangélica de Goiás  
– UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 09 de junho de 2021, composta por:

---

Prof. Walquíria Fernandes Marins  
Orientador

---

Prof. Millys Fabrielle Araujo Carvalhaes  
Avaliador

## RESUMO

O objetivo central desse trabalho é identificar um abordagem para descoberta de valor do produto para o cliente em um processo de desenvolvimento de software no ambiente de uma fábrica de *software* acadêmica, dado que diversas metodologias conseguem auxiliar na entrega de valor e não na identificação de valor do produto para o usuário. Através de uma análise bibliográfica sobre o processo base exigido para desenvolvimento de um produto para a engenharia de *software*, sobre as metodologias ágeis já utilizadas no ambiente da fábrica de *software* acadêmica e sobre os *frameworks fit for purpose* e *design thinking* foi possível comparar as similaridades entre os processos de desenvolvimento proposto por cada metodologia ou *framework* e identificar características que auxiliem na identificação de valor do produto.

**Palavras-chave:** Identificar valor do produto. Fábrica de *software* acadêmica. Processo de desenvolvimento de *software*.

## LISTA DE ILUSTRAÇÕES

<b>Figura 1</b> - Processo SCRUM .....	16
<b>Figura 2</b> - Processo OpenUP .....	18
<b>Figura 3</b> - Infográfico framework F4P .....	21

## **LISTA DE ABREVIATURAS E SIGLAS**

FS – Fábrica de Software

FSA – Fábrica de *software* acadêmica

FTT – Fábrica de Tecnologias Turing

F4P – *Fit For Purpose*

DT – *Design Thinking*

ES – Engenharia de *Software*

## SUMÁRIO

Resumo .....	4
Lista de Ilustrações .....	5
Lista de abreviaturas e siglas .....	6
Sumário.....	7
1. Problema.....	8
2. Objetivos .....	10
2.1. Objetivo Geral .....	10
2.2. Objetivos Específicos .....	10
3. Justificativa.....	11
4. Fundamentação Teórica .....	12
4.1. Engenharia de Software.....	13
4.2. Fábrica de Software acadêmica .....	14
4.3. <i>Frameworks</i> ágeis Aplicados na fit .....	15
4.3.1. <i>SCRUM</i> .....	15
4.3.2. <i>OpenUp</i> .....	17
4.4. <i>Frameworks</i> com foco na identificação e entrega de valor do produto.....	19
4.4.1. Design Thinking .....	19
4.4.2. Fit For <i>Purpose</i> .....	20
4.5. Considerações finais .....	23
5. Estudo comparativo entre os <i>frameworks</i> .....	<b>Erro! Indicador não definido.</b>
6. Metodologia .....	24
7. Cronograma.....	26
8. Resultados alcançados.....	28
9. Resultados esperados.....	31
10. Referências Bibliográficas.....	32
11. [Apêndices/Anexos] .....	<b>Erro! Indicador não definido.</b>

# 1. PROBLEMA

O sucesso de uma empresa está atrelado aos “3P’s” que são, respectivamente: as pessoas que compõem a empresa, o produto gerado e os processos de desenvolvimento desse produto. Esta interpretação, evidencia a necessidade da preocupação com as funções e o bem estar de cada colaborador, entender a competitividade do produto no mercado e se ele se adequa às necessidades do cliente. Por fim, compreender se os processos utilizados são claros para todos os colaboradores e eficientes para garantir que as entregas sejam feitas com qualidade dentro do prazo estabelecido para o projeto (LEMONIS, 2015).

Uma fábrica de *software* (FS) também deve se preocupar com as pessoas, processos e produtos, mas, além disso, devem se atentar a como seus processos se adequam a produção massiva de *softwares* com modelos pré-moldados e como é possível agregar valor a diversos produtos gerados com o processo genérico, logo esse cenário, de agregar valor ao produto, se torna mais complexo devido a quantidade de *softwares* produzidos e ao processo abrangente (POCIVI, 2011; TEBET, 2020).

Apesar da complexidade, entender o processo de produção de um produto de software, em que o produto está alinhado a necessidade do cliente é uma atividade essencial, no entanto esse é um dos maiores desafios enfrentados durante todo o ciclo de vida produto, pois gera uma deficiência do negócio podendo antecipar a fase de declínio, fazendo com que o produto seja descontinuado, ou fazer que o produto não seja aceito pelo mercado, não conseguindo recursos suficientes para sua própria manutenção (PERALTA, 2020).

Embora existam metodologias que auxiliam na perseverança do produto do mercado através da melhora na velocidade de entrega de valor para o cliente como *Lean Statup*, *Lean Innovation*, *Lean Product* e *Costumer Development*, nenhuma delas se preocupam integralmente na identificação de valor do produto para atender à necessidade do cliente (PERALTA, 2020). Eric Ries (2012, p.16) reforça essa teoria ao afirmar que a metodologia *Lean Statup* se baseia no processo de manufatura e tem como princípios o reaproveitamento das experiências de cada colaborador, redução dos tamanhos dos lotes de produção, produção do tipo *just in time*, controle de estoque e aceleração do tempo de ciclo.

Logo, cada uma das metodologias possuem características únicas em seu processo que contribuem de forma isolada nas etapas de desenvolvimento de um produto de *software*, sendo assim aplicação de duas ou mais metodologias diferentes no mesmo processo de desenvolvimento de *software* pode gerar dúvidas quanto a sua aplicação e utilização quando

a finalidade é a identificação do valor do produto e não a somente a entrega (PERALTA, 2020; SOMMERVILLE, 2011).

Além da dificuldade em identificar o valor do produto em um projeto único, as fábricas de software trabalham para criar diversos produtos através de projetos que utilizam um processo genérico, adaptando somente alguns artefatos e rotinas para as necessidades de cada projeto (FERNANDES; TEIXEIRA, 2011), sendo assim a ocorrência de múltiplos projetos simultâneos podem dificultar ainda mais o processo de identificação do valor do produto, uma vez que nem sempre o cliente consegue se expressar da maneira correta ou a equipe não consegue absorver a expectativa do cliente (RODRIGUES, 2014).

Dada a dificuldade na identificação de valor do produto é questionado se há alguma metodologia ou *framework* com o foco na identificação de valor do produto. Caso exista, seria possível adaptar ao processo produtivo de desenvolvimento de software? Quais métricas seria possível implementar para avaliar a adequação do produto à necessidade do cliente? Quais seriam os impactos da aplicação no processo já utilizado? Se não, como as metodologias já existem podem colaborar para construção de um novo modelo focado na identificação de valor?

## **2. OBJETIVOS**

### **2.1. OBJETIVO GERAL**

Identificar um abordagem para descoberta de valor do produto para o cliente em um processo de desenvolvimento de software.

### **2.2. OBJETIVOS ESPECÍFICOS**

- Comparar o processo base da ES com o processo das metodologias de desenvolvimento de *software*.
- Análise teórica sobre os frameworks e metodologias com a finalidade de identificar características que auxiliam na descoberta de valor do produto.
- Selecionar um *framework* para aplicação no processo de desenvolvimento.
- Mapear a situação atual do ambiente de aplicação do *framework* em relação ao processo de desenvolvimento.
- Infundir o conhecimento básico sobre o *framework* escolhido para a equipe.
- Mapear o conhecimento da equipe em relação a processo utilizado e ao *framework* apresentado.
- Identificar novas métricas ou classificar métricas existentes de acordo com *framework* escolhido.
- Aplicar o *framework* no processo de desenvolvimento.
- Mapear e avaliar os resultados da aplicação do *framework*.

### 3. JUSTIFICATIVA

A ubiquidade de *software* na vida cotidiana é algo indiscutível, uma vez que está presente até nas atividades diárias mais simples sem precisar de altos investimentos, dado este fato, afirma-se que 100% da população mundial é afetada diretamente ou indiretamente por tecnologias de *software* sendo que grande parte dessas pessoas tem contato direto com essas tecnologias e, mesmo com a crescente presença de *software* na vida humana, não é dado a devida atenção para desenvolver produtos que forneçam uma boa experiência para o usuário (TORRES, 2015).

De acordo com Salviano (2006), “a comunidade tem procurado entender a natureza das atividades de *software* e sua relação com as outras disciplinas da engenharia, e desenvolver métodos, técnicas, modelos e ferramentas para a engenharia de *software*”, ademais Torres (2015) confirma que o processo de desenvolvimento de *software* teve uma grande evolução, no entanto defende a necessidade de uma área específica para gerenciar o produto com o intuito de desenvolver algo que atenda as expectativas e necessidade do usuário.

Portanto, entender o propósito do cliente em relação ao produto é essencial para sucesso de qualquer negócio (PERALTA, 2020), para David J. Anderson e Alexei Zheglov (2018, p. 38) a falta de capacidade para entender o propósito é o mesmo que “voar às cegas”, essa deficiência leva os consumidores a procurarem seus concorrentes e, por não conseguir identificar o porquê eles tomaram essa decisão, as estratégias elaboradas serão falhas e os investimentos feitos de maneira errada.

Logo, “muitas empresas, no intuito de fechar um venda, gastam centenas de milhares de reais para encantar o cliente, quando muitas vezes o cliente esperava apenas o atendimento de sua necessidade.”, por essa razão entender a necessidade do cliente e colocá-lo em destaque antes de iniciar qualquer projeto pode gerar economias de tempo e dinheiro, aumentar a capacidade produtiva da equipe e satisfação do cliente (RODRIGUES, 2014, p.74).

## 4. FUNDAMENTAÇÃO TEÓRICA

Um produto pode ser definido de diversas formas para diferentes propósitos, no entanto Eric Ries (2012, p.25), em seu livro *A startup enxuta*, apresenta uma definição muito completa e abrangente, logo ele usa a definição de produto como:

“... aquela que abrange qualquer fonte de valor para as pessoas que se tornam clientes. Qualquer coisa que os clientes vivenciam da interação com uma empresa deve ser considerada parte do produto daquela empresa. Isso é verdade em relação a uma quitanda, um site de comércio eletrônico, um serviço de consultoria e uma entidade de serviço sem fins lucrativos.”

Após entender a definição de um produto e relacioná-la com o contexto de desenvolvimento de *software* surge o conceito de produto digital que para Joaquim Torres (2020, p.34) é “qualquer software que tenha usuários”, para ele é possível classificar um produto como digital observando tipo de público que ele atende e pela forma que ele é entregue aos consumidores finais.

Produtos digitais podem estar presentes em diversos tipos de empreendimento, mas é necessário entender a natureza de atuação da empresa para explicitar a função do produto no negócio, sendo assim há a natureza **digital** em que o produto de *software* é a parte mais importante do negócio, sem ele o negócio não existiria, há a natureza **tradicional** em que o produto de *software* potencializa o crescimento de algo que já existia e não representa a parte principal do negócio e, por último, há a natureza **tradicional nascida digital** em que o produto foi concebido como um *software*, porém ele poderia existir fisicamente, neste caso também não representa a parte principal do negócio (TORRES, 2020).

Compreender o que é um produto e sua função em um negócio é essencial, no entanto, para entender seu comportamento em relação a um negócio, é necessário conceituar o que é valor de produto para o cliente. Sendo assim, Mello e Leão (2008) conceituam o valor de produto para o cliente de três<sup>1</sup> formas distintas, porém para esse trabalho o valor do produto para o cliente deve ser considerado como “uma relação de custo e benefício, ou seja,

---

<sup>1</sup> Disponível em: <http://www.spell.org.br/documentos/ver/31086/uma-revisao-critica-sobre-o-uso-de-diferentes-conceitos-de-valor-no-marketing/i/pt-br>.

a diferença entre os valores que o cliente ganha (benefícios funcionais e subjetivos) comprando e usando um produto, e os custos (dinheiro, esforço, custo físico ou psíquico) para obter este produto.”

Apesar das diferentes naturezas de atuação de um produto nas empresa o desenvolvimento de um produto deve seguir um processo e, conseqüentemente, esse processo deve ser gerenciado para a garantir que o produto gerado seja manutenível com o decorrer do tempo, confiável em relação aos dados utilizados e gerados por ele, eficiente no que se propôs executar e aceitável pelo seu público alvo (SOMMERVILLE, 2011).

Processos são indispensáveis para desenvolvimento de um produto, no entanto uma dificuldade encontrada em relação a assimilação de processos é que as pessoas se familiarizam mais com processos físicos e os processos de desenvolvimento de um produto são mais intelectuais e organizacionais, consistindo em uma sequência de passos ou atividades para conceber, projetar e comercializar o produto, outras dificuldades encontradas são a rigidez, ausência ou pluralidade de processos.(EPPINGER; ULRICH, 2016).

#### **4.1. ENGENHARIA DE SOFTWARE**

Entender os processos dos *frameworks* é importante, no entanto é indispensável compreender a disciplina que estuda todas as ações necessárias para o desenvolvimento de software, a Engenharia de *Software* (ES), nela podemos compreender as relações entre os processos básicos de desenvolvimento de *software* e como eles podem ser aplicados no cotidiano de uma equipe (SOMMERVILLE, 2011).

A IEEE, citada por Pressman (2011, p.39), define a ES como “a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de *software*”, sendo assim Sommerville (2011, p.05) complementa essa definição quando afirma que ES “não se preocupa apenas com os processos técnicos do desenvolvimento de *software*. Ela também inclui atividades como gerenciamento de projeto de software e desenvolvimento de ferramentas, métodos e teorias para apoiar a produção de *software*”.

A ES define que os processos de desenvolvimento de *software* devem possuir quatro atividades essenciais, sendo elas a especificação de requisitos, projeto e implementação de *software*, validação de software e evolução de software. Todas essas etapas são essenciais

para garantir que o produto gerado possa atingir os requisitos mínimos de manutenibilidade, confiabilidade, eficiência e aceitação por parte do usuário final (SOMMERVILLE, 2011).

A etapa de especificação de requisitos para a ES é o momento do processo em que os colaboradores são responsáveis por entender as necessidades do produto, nela é criado um documento detalhando os aspectos do sistema que será construído e essa etapa deve ser executada antes ou em paralelo com a etapa de projeto e implementação de software (PRESSMAN, 2011; SOMMERVILLE, 2011).

A etapa de projeto e implementação de *software* para Somerville (2011) é a fase em que busca definir o planejamento de como será executado o projeto, desenvolvendo artefatos como, por exemplo, documento arquitetural e/ou *roadmaps*, além de desenvolver o que foi levantado pela etapa de especificação de requisitos, gerando assim os incrementos funcionais.

Com os incrementos gerados é possível realizar a validação dos requisitos, isto é, verificar se o que foi especificado está de acordo com o que foi implementado e, a última etapa, deve ser a de evolução do *software* em que o objetivo é a entrega e o acompanhamento do produto gerado com a finalidade de encontrar pontos de melhoria, caso sejam encontrados pontos de melhoria todo o processo deve ser reiniciado, dando início a um processo cíclico e não linear (SOMMERVILLE, 2011).

Indiscutivelmente, manter um processo de desenvolvimento de *software* é trabalhoso, por essa razão um dos elementos essenciais para nortear a evolução constante deste processo são as métricas, que possibilita realizar uma análise quantitativa sobre pontos de atenção ou pontos exemplo para manter a “saúde” do processo, no entanto métricas de *software* são complexas de serem aplicadas devido à grande quantidade de incertezas durante um projeto (PRESSMAN, 2011).

## **4.2. FÁBRICA DE SOFTWARE ACADÊMICA**

Para Fernandes e Teixeira (2011) FS é um ambiente voltado para a produção múltipla de produtos de softwares com diferentes escopos e natureza, tendo como base um processo estruturado, controlado e melhorado de forma contínua seguindo as boas práticas da engenharia industrial. No entanto, o processo de uma FS não deve tratar uma pessoa como uma máquina automatizada, uma vez que o processo de uma FS é mais criativo e envolvendo talento das pessoas que o compõe (BORGES; MACHADO; CARVALHO, 2011).

Neste contexto de um processo criativo e envolvendo talento das pessoas dentro de uma FS, surge o conceito de Fábrica de Software Acadêmica (FSA), em que a finalidade é capacitar os alunos de instituições de ensino superior para o mercado de trabalho através de desafios, palestras, pesquisas, eventos, cursos e sempre promovendo a interdisciplinaridade (BORGES; MACHADO; CARVALHO, 2011).

Sendo assim, em conjunto com os Cursos de Bacharelados em Computação, a UniEvangélica fundou em 2006 a FTT, uma FSA com foco em desenvolvimento de *softwares* personalizados para a própria instituição ou para instituições parceiras a UniEvangélica e, desde 2006, forma profissionais mais qualificados para o mercado de trabalho, uma vez que o próprio grupo gestor da FTT incentiva a adoção das tecnologias e ferramentas mais atuais do mercado (SOUSA et al., 2018).

### **4.3. FRAMEWORKS ÁGEIS APLICADOS NA FTT**

Fernandes e Teixeira (2004), citados por Pocivi (2011), relataram que uma FS deve possuir frameworks que contemplem tanto o processo produtivo como o processo gerencial. Logo, a FTT utiliza um processo híbrido baseado nos frameworks ágeis OpenUp e Scrum, sendo o OpenUp utilizado para nortear o processo produtivo de desenvolvimento de software iterativo e o Scrum como processo gerencial (SOUSA et al., 2018).

#### **4.3.1. SCRUM**

O *framework Scrum* foi citado pela primeira vez em 1986 por Hirotaka Takeuchi e Ikujiro Nonaka e foi aperfeiçoada por Jeff Sutherland e Ken Schwaber até 1995, ano o qual publicaram o artigo “*Scrum and the perfect storm*”<sup>2</sup> e, apesar de seu ano de publicação ser anterior à data do manifesto ágil, o SCRUM ainda é considerado um *framework* ágil de desenvolvimento de projetos de *software* (PHAM; PHAM, 2012).

O processo *Scrum* deve ser empírico incitando a melhoria contínua através da observação da realidade ou experiência dos colaboradores e também aplica conceitos da

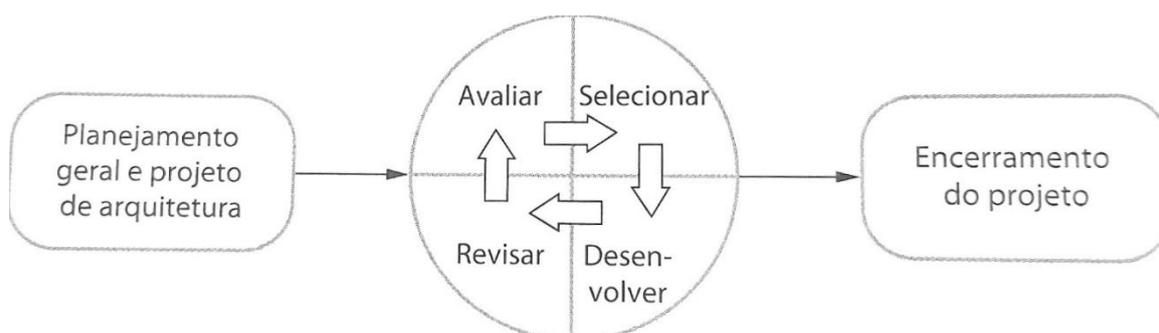
---

<sup>2</sup> Disponível em: <http://www.controlchaos.com/my-articles>. Acessado em: 23 de mar. De 2021.

metodologia *lean thinking*<sup>3</sup>. Além de ser um processo empírico e aplicar a metodologia *lean thinking*, o *Scrum* se baseia em três pilares: Transparência, Adaptação e Inspeção, gerando valores como compromisso, foco, abertura, respeito e coragem (SCHWABER; SUTHERLAND, 2020).

O SCRUM é dividido em três fases, conforme a Figura 1, sendo na primeira em que são definidos o planejamento geral do projeto e arquitetura do software, a segunda são os ciclos Sprint, em que cada ciclo é desenvolvido um incremento utilizável do software seguindo sempre o ciclo iterativo e incremental de avaliar, selecionar, desenvolver e revisar, já a terceira, e última fase, é a fase onde encerra o projeto entregando manuais de sistema, documentação exigida, quadros de ajuda e avalia a lições aprendidas com o projeto (SOMMERVILLE, 2011).

**Figura 1 - Processo SCRUM**



Fonte: (SOMMERVILLE, 2011).

Além da estrutura inicial do processo pré-definida, o *SCRUM* reconhece os colaboradores como *Scrum team* e os divide em papéis que são: 1) *Product Owner*: responsável maximizar o valor do produto junto a equipe; 2) *Scrum Master*: que é responsável por manter o bom andamento do projeto conforme o guia do *Scrum*; 3) *Developers*: são os responsáveis por criar qualquer incremento funcional na *sprint* (SCHWABER; SUTHERLAND, 2020).

Com a equipe definida será possível realizar uma *Sprint* (um ciclo iterativo e incremental de desenvolvimento de software) que além do tempo de desenvolvimento é composta pelos eventos: 1) *Sprint planning*: evento em que se define a meta da *Sprint*, logo

---

<sup>3</sup> Uma metodologia criada com o foco em evitar desperdícios e agregar valor ao produto, inicialmente aplicada em ambiente fabris e que, ao longo dos anos, foi adaptada para outras áreas (WOMACK; JONES, 2004).

nesse momento todo o *scrum team* e os convidados pelo *product owner* (PO) discutem quais itens do *product backlog* (PBL) devem ser inseridos na *sprint backlog*; 2) *Daily scrum*: evento em que o *scrum team* relata qual o progresso em relação a meta da *sprint* auxiliando assim para a elaboração do um plano de ação para o dia; 3) *Sprint review*: evento em que os *steakeholdres*<sup>4</sup> realizam a inspeção de todos os artefatos gerados durante a *sprint*; 4) *Sprint retrospective*: evento em que são identificados os pontos de atenção em relação a processos, ferramentas (SCHWABER; SUTHERLAND, 2020).

Para Schwaber (2004) e Schwaber e Beedle (2001) SCRUM é um metodologia ágil com o foco gerenciamento de desenvolvimento de software iterativo (*apud*. SOMMERVILLE, 2011), logo é uma metodologia que não se aplica a todos os contextos de desenvolvimento de software e há a necessidade de adaptar os processos de acordo com ambiente (PHAM; PHAM, 2012)

Por esse motivo a Fábrica de Tecnologias Turing (FTT), uma FSA mantida pela Associação Educativa Evangélica e coordenada pelos cursos bacharelados em computação, além de utilizar a metodologia *Scrum* para o processo gerencial, utiliza-se também o *OpenUp* para o desenvolvimento de *software* iterativo e incremental (POCIVI, 2011; SOUSA et al., 2018).

### 4.3.2. OPENUP

Pocivi (2011, p.24) defini o *OpenUp* como

“Um processo de desenvolvimento com abordagem ágil, voltado para equipes pequenas. Ele foi criado pela IBM, reflete as boas práticas contidas no RUP e pode ser extensível e otimizado para as necessidades de cada projeto ou ambiente.”.

A metodologia tem como características base ser mínima, utilizando somente os artefatos necessários para o desenvolvimento de software, completa, abrangendo todos os passos do processo de desenvolvimento ágil de *software*, e extensível, para se adaptar à

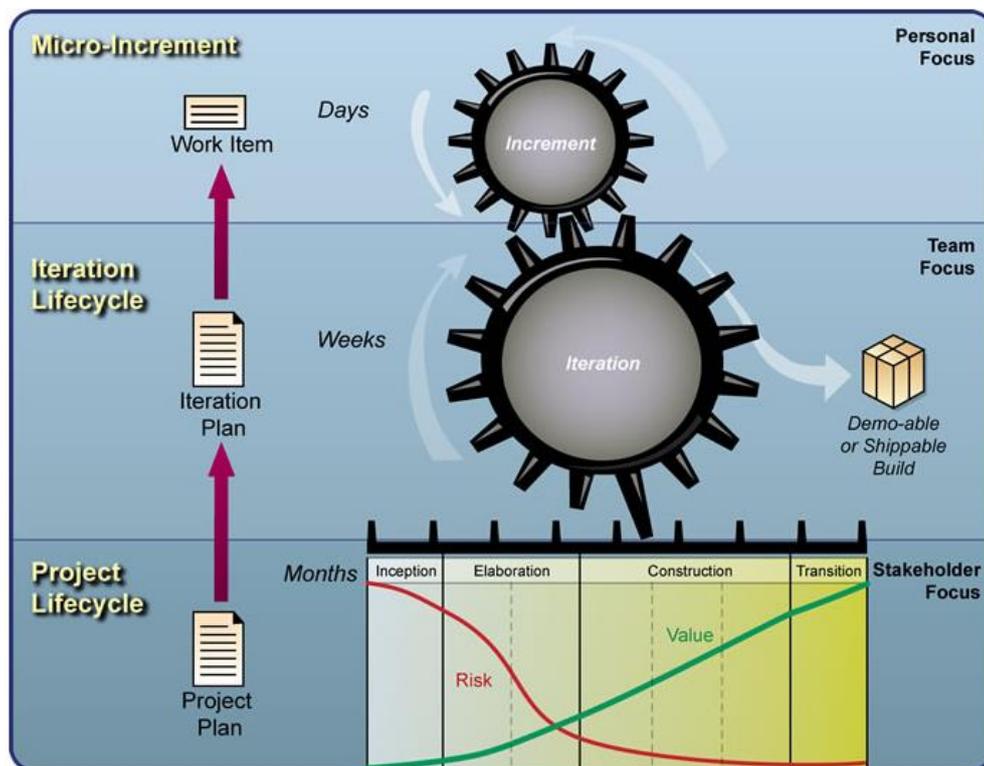
---

<sup>4</sup> Saiba mais sobre *steakholders* em: <https://robsoncamargo.com.br/blog/O-que-sao-stakeholders-Saiba-tudo-sobre-eles-e-sua-importancia>. Acessado em: 01 de maio de 2021.

realidade de cada projeto, podendo também ser utilizada em conjunto com outras metodologias (IBM, 2006).

O *OpenUP* pode ser hierarquizado em três níveis, sendo o mais alto o ciclo de vida do projeto, o mediano o ciclo de vida da iteração e o mais baixo é o micro incremento, observe a Figura 2. No micro incremento o foco é realizar pequenos pacotes de trabalhos que vão colaborar diretamente para o resultado, no ciclo de vida da iteração o objetivo é entregar um incremento utilizável derivado das ações do micro incremento e ciclo de vida do projeto que visa a entrega completa do produto gerado pelo projeto (IBM, 2006).

Figura 2 - Processo OpenUP



Fonte: (IBM, 2006).

Segundo a IBM (2006, *online*), o ciclo de vida do projeto para o *OpenUp* é dividido em quatro fases:

- *Inception*: essa é a primeira fase do projeto e uma característica notável nesse fase é o risco de falha do projeto muito alto e um valor baixíssimo, nessa etapa é identificado o que fazer no projeto, quais são os requisitos de maior valor; são exibidas uma possível solução para os requisitos e a estimativa do projeto.

- *Elaboration*: na segunda fase do projeto nota-se que o risco do projeto decai e o valor tem uma alta já devido a definições da primeira fase e também porque nesta etapa um dos objetivos é mitigar os riscos do projeto e, além disso, detalhar melhor os requisitos e definir o projeto de arquitetura do software;
- *Construction*: nesta fase o produto é construído baseado nas informações adquiridas nas fases anteriores e é importante destacar que essa fase o produto é desenvolvido iterativamente e de forma incremental. Nesta etapa também vemos os riscos do projeto quase estáveis e a etapa que o valor mais aumenta;
- *Transition*: na última fase do projeto em que o objetivo é realizar a entrega do produto gerado e entender e documentar as lições aprendidas com o projeto realizado.

#### **4.4. FRAMEWORKS COM FOCO NA IDENTIFICAÇÃO E ENTREGA DE VALOR DO PRODUTO**

A utilização de um processo híbrido para desenvolvimento de software em uma FSA se tornou essencial no contexto da FTT, uma vez que o *Scrum* abrange a área de processo gerencial e o *OpenUP* o processo produtivo (SOUSA et al., 2018). No entanto nenhuma das duas metodologias tem o foco na identificação ou entrega de valor do produto, pois, como afirma Pocivi (2011, p.77), “O Scrum e o OpenUp foram selecionados, principalmente, por permitirem executar, do princípio ao fim, todas as fases e as respectivas atividades do ciclo de vida de desenvolvimento de sistemas, em pequenos espaços de tempo.”.

Sendo assim, essas metodologias tem o foco na agilidade de entrega do produto, por essa razão é necessário complementar o processo quando o objetivo é a identificação e/ou entrega de valor para o cliente.

##### **4.4.1. DESIGN THINKING**

O *Design Thinking* (DT) é um *framework* o qual tem o objetivo de criar soluções inovadoras, principalmente porque foi criado se baseando na visão que os designers teriam no momento de concepção de um novo produto, assim o processo de concepção de um novo

produto para o DT deve ser realizado por uma equipe multidisciplinar para ampliar a possibilidade de novas ideias e garantindo a pluralidade de visões (VIANNA et al., 2012).

Empatia, colaboração e experimentação são os pilares do DT devendo estar presentes ao longo de todo o processo sugerido para a concepção e manutenção do produto que será desenvolvido. O pilar da **empatia**, considerado o mais importante, é o que permite os profissionais se colocar no lugar do cliente para compreender melhor suas necessidades e propor soluções, o pilar da **colaboração** é aquele que coloca em destaque a importância do trabalho em equipe e na necessidade de envolvimento de todos no processo e, por último, o pilar da **experimentação** que tem o objetivo de experimentar ideias e cenários diferentes para fomentar a criatividade nas soluções de problemas (UPIS, 2019).

O DT é composto por cinco etapas, que são: 1) Empatia: etapa com o foco de entender as dificuldades que determinado cliente ou usuário enfrenta; 2) Definição: nesta etapa define-se a necessidade e as motivações do usuário final; 3) Idealização: é a etapa a qual todas ideias de solução para problemas do usuário final são aceitas com a finalidade de expandir a possibilidade de ação, utiliza como ferramenta o *Brainstorm*<sup>5</sup> ; 4) Prototipação: este é o momento de se criar algo utilizável para usuário; 5) Testar: nesta etapa são coletados os *feedbacks* dos usuários com a finalidade de reiniciar o ciclo e manter o produto em constante evolução (SOUZA, 2016).

Em uma pesquisa realizada na empresa Positivo Informática por Bonini e Sbragia (2011) relata que 89% dos colaboradores concordam que as soluções geradas são mais criativas, 81% acreditam ser mais eficaz a utilização do DT no processo de criação de produtos, 78% acreditam diminuir os riscos e 74% afirmam geração de retornos financeiros. Esses números comprovam a eficácia ao implementar esse *framework* em empresas de tecnologia.

#### **4.4.2. FIT FOR PURPOSE**

David Anderson e Alexei Zheglov, criaram, em 2017, o *framework Fit for purpose* (F4P) para cobrir a lacuna de eficácia dos negócios em relação a produto e orientar os

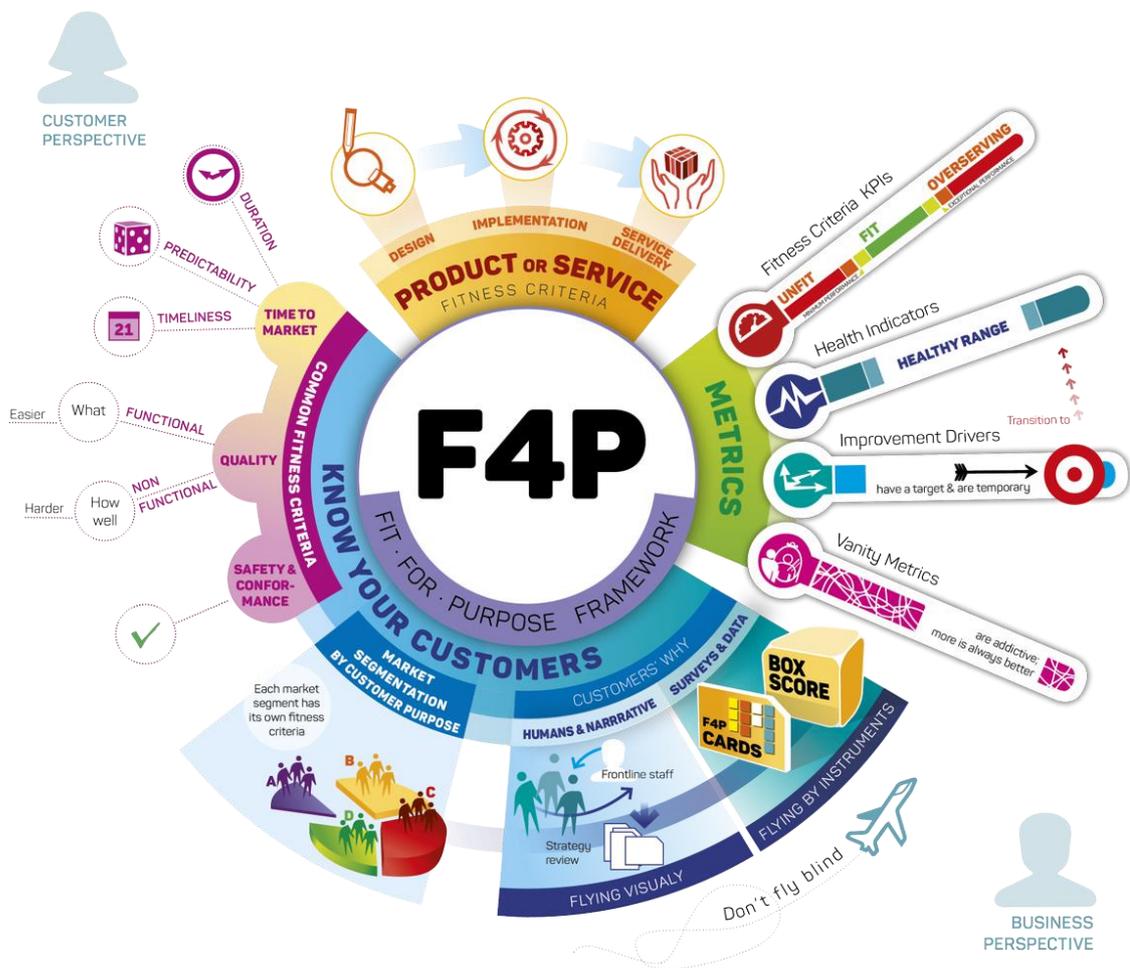
---

<sup>5</sup> “Traduzida para ‘Tempestade de ideias’, tem como objetivo principal capturar muitas ideias de maneira rápida e bruta (não lapidada) e funciona muito bem em equipes” (MELO; ABELHEIRA, 2015).

negócios sobre o propósito do cliente entregando um produto que se adeque a sua necessidade (ANDERSON; ZHEGLOV, 2018).

Como observado na Figura 3 o *framework* é dividido em três grandes áreas, que são o *product or service* (produto ou serviço, representado pela parte laranja), *metrics* (métricas, representado pela parte verde) e *know your customer* (conheça seu cliente, representado pela parte azul).

**Figura 3 - Infográfico framework F4P**



Fonte: (ANDERSON; ZHEGLOV, 2018).

Para garantir que um produto ou serviço se adeque a necessidade do cliente e, como consequência, consiga manter um negócio de sucesso é necessário que os gestores desse consigam entender e balancear três momentos do processo de desenvolvimento, que são: 1) **Design** (Design): momento com objetivo de entender quais as funcionalidades daquele produto

se propõe a resolver; 2) **Implementation** (Implementação): neste momento será possível focar em como o produto será, se aproximando ao máximo do que foi idealizado no *Design*; 3) **Service Delivery** (Entrega de Serviço): etapa com o foco na experiência ao utilizar o produto e, também, coleta de *feedbacks* (DUBOVSKI, 2019).

Além de entender e conseguir dar a mesma importância ao design, implementação e entrega de serviço é necessário ainda garantir que o produto ou serviço esteja atingindo o *Fitness Criteria* do usuário final (DUBOVSKI, 2019). Para Anderson e Zheglov (2018) as métricas são essenciais para entender o propósito do cliente e manter o negócio estável, logo eles dividiram as métricas em quatro categorias:

- *Fitness criteria KPIs* (métricas chaves de adequação): são as métricas que direcionam o objetivo do negócio e, normalmente, são os mesmos critérios que o consumidor usa para selecionar seu produto ou serviço. Para auxiliar nesse processo de identificação dessas métricas é aconselhável o uso de KPIs<sup>6</sup>;
- *Health indicators* (métricas de saúde): são as métricas essenciais para a empresa desenvolvedora, mas não tão relevantes para o usuário final, sendo assim elas garantem que o negócio está “saudável”;
- *Improvement drivers* (métricas de melhoria): são as métricas geradas a partir de alguma anomalia no negócio que gere risco para a experiência do usuário, por tanto deve ser acompanhada para aquela anomalia desapareça e volte ao estado normal;
- *Vanity metrics* (métricas de vaidade): são as métricas sem relevância para nenhuma das partes envolvidas no negócio e são as que geram a sensação positiva, mas seu uso deve ser consciente, pois ela pode apresentar falsos positivos.

Com o intuito de conhecer o usuário final do produto o F4P propõe identificar se os critérios de adequação do usuários ao produto estão sendo atingidos, qual a segmentação do mercado aquele usuário pertence e se ele está dentro do público alvo e estabelecer uma relação com o usuário constante, mas com atenção em mesclar interação humana com

---

<sup>6</sup> Para mais informações acesse <https://rockcontent.com/br/blog/kpi/>.

formulários de pesquisa para não gerar uma situação de experiência negativa (DUBOVSKI, 2019).

#### **4.5. CONSIDERAÇÕES FINAIS**

A definição sobre o que é um produto de software e suas classificações faz-se necessário para entender sobre qual perspectiva esse trabalho tratará um produto, já o conceito de Fábrica de *Software* e os *frameworks* utilizados por ela explicita a base dos processos produtivos e gerenciais já utilizados e, por último, os *frameworks* voltados para a identificação de valor do produto para o usuário final limitam o escopo, dado a grande quantidade de *frameworks* ágeis existentes no mercado.

Faz-se necessário para este trabalho compreender também a aplicação da disciplina ES no contexto de processo e qualidade, uma vez que o processo base definido pela ES será comparado aos processos dos *frameworks* citados para tomadas de decisão ao longo do trabalho e também compreender a importâncias das métricas na área de qualidade, pois elas serão utilizadas para o acompanhamento das alterações sugeridas.

## 5. METODOLOGIA

No primeiro momento, foi realizada uma análise bibliográfica com foco em identificar características do processo base definido pela Engenharia de *Software* nas metodologias *Scrum*, *OpenUp*, *Design Thinking* e *Fit for Purpose*. Essa comparação permitiu identificar pontos positivos e negativos de cada metodologia e *framework* em relação a identificação de valor de um produto para o cliente.

Em seguida, como uma fábrica desenvolve diversos projetos, será implementada uma pesquisa descritiva, para analisar os projetos que se enquadrariam para aplicação do *framework*, nessa pesquisa serão levadas em consideração as informações relativas a nível de proximidade com o cliente, complexidade do projeto, prioridade, estrutura da equipe, cultura e o *Net Promote Score* (NPS) da equipe e cliente no momento da pesquisa.

No intuito de apresentar o *framework*, agregar conhecimentos para os integrantes e levantar hipóteses de impedimentos que possam ser gerados será feita uma apresentação de conteúdo, em método expositivo e com o auxílio de slides, em seguida, uma dinâmica de competição de grupos em que todos os integrantes participarão e, por último, um momento aberto para questionamentos, sugestões ou críticas.

Após a realização da apresentação e da dinâmica, será disponibilizado um formulário pelo google forms e divulgado para os integrantes responderem. Após respondido, o questionário servirá como base para mapear, através de gráficos, o conhecimento dos integrantes acerca dos projetos em andamento, processos, metodologias e métricas já utilizadas e, também, acerca do *framework* apresentado.

Com o mapeamento feito será aplicado *framework* escolhido em conjunto com a coordenação da FS, além disso também serão implementadas métricas, que serão definidas de acordo com os pontos de maior vulnerabilidade do processo para o desenvolvimento de software e com a finalidade de acompanhar a evolução do processo de desenvolvimento no ambiente de FS. Para isso, essas métricas serão classificadas em grupos para facilitar o acompanhamento.

Para a finalização do projeto de pesquisa serão mapeados todos os resultados obtidos através das métricas implementadas e, além disso, da disponibilização outro formulário pelo google forms para a identificação da satisfação da equipe e do cliente no processo de mudança durante a aplicação do no *framework*. Para medir a satisfação, será comparado o NPS pré implementação do *framework* e pós implementação e, em relação ao processo, serão

comparadas as métricas levantadas no começo do projeto e a variação dos resultados delas durante a aplicação com intuito de medir a velocidade de entrega e qualidade do software.

## 6. CRONOGRAMA

ATIVIDADE	2021										2021									
	Fev		Mar		Abr		Maio		Jun		Jul		Ago		Setr		Out		Nov	
	1ª Quinzena	2ª Quinzena																		
Análise sobre os conceitos de produto, fábrica de software e engenharia de software.	X																			
Análise bibliográfica do metodologias utilizadas na FTT		X	X																	
Análise bibliográfica do frameworks de identificação de valor do produto				X	X															
Comparativo entre os frameworks, metodologias e processo base da ES						X	X	X												
Análise dos projetos									X											
Formulação da dinâmica e questionário									X	X										
Aplicação da dinâmica e questionário											X									
Estudo das métricas já utilizadas												X								
Estudo de novas métricas para as lacunas identificadas													X	X						
Definição do modelo de identificação de necessidade do usuário															X	X	X			



## 7. RESULTADOS ALCANÇADOS

A análise teórica realizada com a finalidade de conceituar produto, fábrica de software e processo base de engenharia de software e também análise bibliográfica das metodologias utilizadas pela FSA e dos outros dois *frameworks* permitiram a conclusão do objetivo de analisar teoricamente os frameworks e metodologias com a finalidade de identificar características que auxiliam na descoberta de valor do produto.

Com esse resultado foi possível realizar a comparação entre os processos dos *frameworks* e metodologias com o processo base da ES com o processo das metodologias de desenvolvimento de *software*, onde podemos observar quais metodologias cumpriam as etapas base de um processo de desenvolvimento de um produto de software.

**Figura 4** - Comparação do processo base de ES com os processos dos frameworks e metodologias

Processo base definido pela Engenharia de Software			
Especificação de Requisitos	Projeto e Implementação de software	Validação de software	Evolução de software
Planejamento geral e projeto de arquitetura			
Ciclo iterativo e incremental (avaliar, selecionar, desenvolver e revisar)			Entrega do projeto
Inception			
	Elaboration	Construction	Transition
Empatia	Definição	Idealização	Prototipação
			Testar
Design	Implementation	Service Delivery	
Legenda: <span style="color: yellow;">■</span> Scrum <span style="color: green;">■</span> OpenUP <span style="color: red;">■</span> Design Thinking <span style="color: purple;">■</span> Fir for Purpose			

Fonte: Autoria própria.

Para Sommerville (2011), as etapas de especificação de requisitos, projeto e implementação de software, validação de software e evolução de software devem estar presentes em todo processo de desenvolvimento de software. Baseado nesta afirmação e na

necessidade de entender o cenário do problema levantado para essa pesquisa foi criada a Figura 4 que representa uma comparação entre as etapas do processo base definido pela ES com as etapas dos *frameworks* DT e F4P e das metodologias *Scrum* e *OpenUp*.

A etapa de especificação de requisitos definida pelo processo base de ES se relaciona diretamente com etapas de planejamento geral e projeto de arquitetura (*Scrum*), ciclo iterativo (*Scrum* – avaliar), *inception* (*OpenUp*), *elaboration* (*OpenUp*), empatia (DT), definição (DT) e *design* (F4P). Essa relação existe, pois todas essas fases citadas são direcionadas a gerar um artefato que detalhe aspectos do produto que será desenvolvido, sendo assim esses artefatos podem ser compreendidos como uma especificação de requisitos, já que para Pressman (2011) uma especificação de requisitos do sistema “pode ser um documento por escrito, um conjunto de modelos gráficos, um modelo matemático formal, um conjunto de cenários de uso, um protótipo ou qualquer combinação dos fatores citados.”.

Já a etapa de projeto e implementação de *software* que é a fase responsável por planejar e executar o planejado pode ser comparada com as fases de ciclo iterativo (*Scrum* – selecionar e desenvolver), *elaboration* (*OpenUp*), *construction* (*OpenUp*), idealização (DT), prototipação (DT) e *implementation* (F4P).

A etapa de validação de *software* tem o objetivo de validar se o produto gerado pela etapa de planejamento e implementação de *software* atende o que foi definido na etapa de especificação de requisitos e as etapas das metodologias e *frameworks* que correspondem a essa definição são: ciclo iterativo (*Scrum* – revisar), *construction* (*OpenUp*), testar (DT) e *service delivery* (F4P).

A última etapa para o desenvolvimento de *software* definida para ER é a etapa de evolução do software que tem o objetivo de colher *feedbacks* e apontar pontos de melhoria, dessa forma as etapas das metodologias e *frameworks* que correspondem a essa definição são: entrega de projeto (*Scrum*), *transition* (*OpenUp*), testar (DT) e *service delivery* (F4P).

Dado a comparação entre as etapas do processo base definido pela ES com as etapas dos *frameworks* DT e F4P e das metodologias *Scrum* e *OpenUp*, pode-se afirmar que o DT, F4P, *Scrum* e *OpenUp* percorrem todo o processo base da ES e que para uma possível aplicação dos *frameworks* DT ou F4P no processo de desenvolvimento de uma fábrica de software acadêmica que utilize as metodologias *Scrum* e *OpenUp*, não há a necessidade de alteração das etapas do processo já utilizado, porém há a necessidade de alteração de artefatos gerados, métricas e rotinas.

Sendo assim, a decisão sobre qual *framework* aplicar quando o intuito é a identificação de valor do cliente não deve considerar se o *framework* abrange todas as etapas mínimas exigidas pela Engenharia de *Software* para o desenvolvimento de produtos de *software*, mas deve considerar as características sobre como cada *framework* pode auxiliar no processo de identificação de valor do produto para o cliente.

Logo, para esse projeto de pesquisa, *Fit for Purpose* se sobressai, pois, além de se preocupar com a identificação de valor do produto, o F4P também classifica suas métricas em grupos que podem auxiliar avaliação dos resultados, podendo facilitar em futuras obtenção de resultados.

## 8. RESULTADOS ESPERADOS

Além dos resultados já obtidos é esperado que esse trabalho ainda apresente a aplicação do framework selecionado no ambiente de fábrica de software. No entanto, antes de realizar a aplicação é necessário realizar uma pesquisa com o intuito de mapear a situação real da FSA à qual será aplicado o *framework*, além disso é preciso mapear o nível de conhecimento dos colaboradores em relação a *framework* escolhido.

Para reforçar e garantir o aprendizado dos colaboradores sobre o *framework* espera-se que seja aplicada uma dinâmica seguida de um questionário. Esse questionário e está dinâmica deverá compreender o quanto os colaboradores conhecer dos processos utilizados na FSA, métricas aplicadas nesse processo, conhecimento sobre o *framework* apresentado e satisfação dos colaboradores em relação a esse processo.

Após a dinâmica aplicada e as informações do questionário compiladas em forma de gráficos deve-se dar início a aplicação do *framework* proposto no processo de desenvolvimento da FSA, sendo assim será levantado quais alterações foram necessárias para se adequar ao processo e também serão implementadas métricas para acompanhamento da evolução do processo.

Ao concluir a aplicação deverá ser realizada uma análise dos dados gerados durante o momento em que foi aplicado o *framework* com a finalidade de entender os impactos dessa aplicação levantando os pontos positivos e negativos, além de analisar se conseguir cumprir o objetivo principal que é a melhora no processo de identificação do valor do produto.

## 9. REFERÊNCIAS BIBLIOGRÁFICAS

- ANDERSON, D. J.; ZHEGLOV, A. **Fit For Purpose: How Modern Business Find, Satisfy & Keep Customers**. 2. ed. Seattle, WA: Blue Hole Press, 2018.
- BONINI, L. A.; SBRAGIA, R. O Modelo de Design Thinking como Indutor da Inovação nas Empresas: Um Estudo Empírico. **Revista de Gestão de Projetos**, jun. 2011.
- BORGES, K. S.; MACHADO, R. P.; CARVALHO, T. P. DE. **Programa de Extensão “Fábrica de Software Acadêmica”:** contribuindo para a formação profissional na área de informática. Porto Alegre: IFRS, 2011.
- DUBOVSKI, M. **Fit for Purpose e a importância de conhecer o propósito do cliente**. Disponível em: <<https://medium.com/@dubovski/fit-for-purpose-e-a-importancia-de-conhecer-o-propósito-do-cliente-abdfd29b9baf>>. Acesso em: 2 maio. 2021.
- EPPINGER, S. D.; ULRICH, K. T. **Product Design and Development**. 6. ed. Nova York, NY: McGraw-Hill Education, 2016.
- FERNANDES, A. A.; TEIXEIRA, D. DE S. **Fábrica de Software: implantação e gestão de operações**. 1. ed. São Paulo, SP: Editora Atlas, 2011.
- IBM. **OpenUP**. Disponível em: <[https://download.eclipse.org/technology/epf/OpenUP/published/openup\\_published\\_1.5.1.5\\_20121212/openup/index.htm](https://download.eclipse.org/technology/epf/OpenUP/published/openup_published_1.5.1.5_20121212/openup/index.htm)>. Acesso em: 2 maio. 2021.
- LEMONIS, M. A. **Confira a entrevista exclusiva com Marcus Lemonis**. Disponível em: <<https://history.uol.com.br/noticias/confira-entrevista-exclusiva-com-marcus-lemonis>>. Acesso em: 11 dez. 2020.
- MELLO, S. C. B.; LEÃO, A. L. M. DE S. Uma Revisão Crítica sobre o Uso de Diferentes Conceitos de Valor no Marketing. **RAC-Eletrônica**, abr. 2008.
- MELO, A.; ABELHEIRA, R. Design Thinking & Thinking Design: Metodologia, ferramentas e uma reflexões sobre o tema. In: **Design Thinking & Thinking Design: Metodologia, ferramentas e uma reflexões sobre o tema**. 1. ed. São Paulo, SP: Novatec, 2015.
- PERALTA, C. B. DA L. **Mensuração de valor em produtos inovadores na perspectiva do cliente**. Porto Alegre: UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL ESCOLA DE ENGENHARIA PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO, 2020.
- PHAM, A.; PHAM, P.-V. **Scrum em ação: Gerenciamento e desenvolvimento ágil de projetos de software**. 1. ed. São Paulo, SP: Novatec, 2012.
- POCIVI, V. C. B. **Um estudo para melhorado processo de ensino e aprendizagem de engenharia de software em cursos de graduação**. Recife: C.E.S.A.R – CENTRO DE ESTUDOS E SISTEMAS AVANÇADOS DO RECIFE, 2011.
- PRESSMAN, R. S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH Editora, 2011.
- RIES, E. **A startup enxuta**. 1. ed. São Paulo, SP: Leya Editora, 2012.
- RODRIGUES, E. **21 Erros Clássicos da Gestão de Projetos - Eli Rodrigues - Google Livros**. 1. ed. Rio de Janeiro, RJ: Brasport, 2014.
- SALVIANO, C. F. **Uma Proposta Orientada a Perfis de Capacidade de Processo para Evolução da Melhoria de Processo de Software**. [s.l: s.n.].
- SCHWABER, K.; SUTHERLAND, J. **Guia do SCRUM**. Disponível em: <[https://www.scrum.org/Portals/0/Documents/Scrum Guides/Scrum\\_Guide.pdf](https://www.scrum.org/Portals/0/Documents/Scrum Guides/Scrum_Guide.pdf)>. Acesso em: 1 maio. 2021.
- SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo, SP: Pearson Education,

2011.

SOUSA, F. C. et al. **Preparaçãodos Acadêmicos dos Cursos de Computação da UniEVANGÉLICA para o Mercado de Trabalho através de uma Fábrica de Software Acadêmica.** *Revista Ada Lovelace*, dez. 2018.

SOUZA, M. **O que é e para que serve o Design Thinking.** Disponível em: <[https://matheusdesouza.com/2016/06/22/o-que-e-e-para-que-serve-o-design-thinking/#:~:text=Design Thinking é uma ferramenta,que os indivíduos resolvam problemas.>](https://matheusdesouza.com/2016/06/22/o-que-e-e-para-que-serve-o-design-thinking/#:~:text=Design%20Thinking%20%C3%A9%20uma%20ferramenta,que%20os%20indiv%C3%ADduos%20resolvam%20problemas.>). Acesso em: 2 maio. 2021.

TEBET, I. **Fábrica de software: o que é, como funciona e os benefícios para sua empresa.** Disponível em: <<https://www.objective.com.br/insights/o-que-uma-fabrica-de-software-pode-fazer-por-sua-empresa/>>. Acesso em: 11 dez. 2020.

TORRES, J. **Gestão de produtos: como aumentar as chances de sucesso do seu software.** 3ª ed. São: Casa do Código, 2015.

TORRES, J. **Liderança de produtos digitais: a ciência e a arte da gestão de time de produto.** São Paulo, SP: Casa do Código, 2020.

UPIS. **Design Thinking: o que é, etapas e como aplicar.** Disponível em: <<https://upis.br/blog/design-thinking/>>. Acesso em: 2 maio. 2021.

VIANNA, M. et al. **Design Thinking: Inovação em negócios.** 1. ed. Rio de Janeiro, RJ: MJV Press, 2012.

WOMACK, J.; JONES, D. **A mentalidade enxuta nas empresas.** 6. ed. Rio de Janeiro, RJ: Editora Campos, 2004.