

**UNIVERSIDADE NOVE DE JULHO
PROGRAMA DE MESTRADO PROFISSIONAL EM ADMINISTRAÇÃO
GESTÃO DE PROJETOS**

**INTEGRAÇÃO DO *DESIGN THINKING* COM MÉTODOS ÁGEIS EM PROJETOS
DE DESENVOLVIMENTO DE *SOFTWARE***

ROBERTO DE SOUZA GÓES

SÃO PAULO

2017

Roberto de Souza Góes

**INTEGRAÇÃO DO *DESIGN THINKING* COM MÉTODOS ÁGEIS EM PROJETOS
DE DESENVOLVIMENTO DE *SOFTWARE***

**INTEGRATION OF *DESIGN THINKING* AND AGILE METHODS IN
DEVELOPMENT *SOFTWARE* PROJECTS**

Dissertação apresentada ao Programa de Mestrado Profissional em Administração: Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**.

Orientadora: Profa. Dra Rosaria de Fatima Segger Macri Russo

SÃO PAULO

2017

Góes, Roberto de Souza.

Integração do Design Thinking com métodos ágeis em projetos de desenvolvimento de Software. / Roberto de Souza Góes. 2017.

106 f.

Dissertação (mestrado) – Universidade Nove de Julho - UNINOVE, São Paulo, 2017.

Orientador (a): Prof^a. Dr^a. Rosaria de Fatima Segger Macri Russo.

Design Thinking. 2. Métodos ágeis. 3. Desenvolvimento de software.

4. Integração. 5. Gestão de projetos.

Russo, Rosaria de Fatima Segger Macri. II. Título

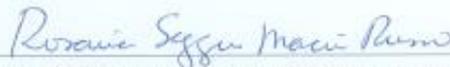
CDU 658.012.2

ROBERTO DE SOUZA GÓES

INTEGRAÇÃO DO DESIGN THINKING COM MÉTODOS ÁGEIS EM PROJETOS
DE DESENVOLVIMENTO DE SOFTWARE

Dissertação apresentada ao Programa de Mestrado Profissional em Administração: Gestão de Projetos da Universidade Nove de Julho – UNINOVE, como requisito parcial para obtenção do grau de **Mestre em Administração**, pela Banca Examinadora, formada por:

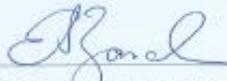
São Paulo, 28 de abril de 2017



Presidente: Profa. Dra. Rosária de Fátima Segger Macri Russo – Orientadora, UNINOVE



Membro: Profa. Dra. Cristiane Drebes Pedron – UNINOVE



Membro: Prof. Dr. Eduardo de Senzi Zancul – POLI/USP

AGRADECIMENTO

Agradeço imensamente a minha orientadora, Professora Doutora Rosaria de Fatima Macri Segger Russo, pela grande dedicação, paciência e motivação.

A Universidade Nove de Julho (Uninove) pela estrutura oferecida, aos professores do curso de gestão de projetos e aos profissionais de apoio. A memória do Prof. Dr. Marcos Roberto Piscopo e Prof. Dr. Ricardo Leonardo Rovai, falecidos durante os dois anos do curso do mestrado.

Agradeço também a minha família, em especial ao meu pai Jose Roberto Góes e minha esposa Dayane Suzi Renzi Góes pelo apoio e motivação. A minha mãe Neide de Souza Góes que além do apoio motivacional, cedeu grande competência técnica e energia nas revisões dos artigos, trabalhos além desta dissertação e das transcrições das entrevistas. O apoio recebido de vocês, jamais conseguirei retribuir.

Em relação à turma do Mestrado Profissional em Administração – Gestão de Projeto (MPA-GP) de 2015, uma turma espetacular, companheira e unida, o mais sincero agradecimento.

Aos amigos Lucas Mendes, André Vidal, Alexandre Abdalla, Vitor Massari e Nelson Abu pela orientação, apoio técnico e consultivo durante minha jornada no curso do mestrado.

A todos, o meu sincero agradecimento.

“Language is Convincing, Seeing is Believing, Touching is Reality”

Alan Kay, Designing Interactions

RESUMO

O *Design Thinking* e os métodos ágeis, são utilizados em um ambiente inovador e dinâmico, com o objetivo de criar produtos e soluções que gerem maior satisfação para o negócio, de modo que traga soluções para os problemas atuais e que também gerem futuros negócios para a empresa. Este estudo foi realizado com o objetivo de verificar como as empresas que adotaram o *Design Thinking* e os métodos ágeis fizeram para integrá-los, desde a criação da solução das iniciativas até o seu desenvolvimento através de *software*. O *Design Thinking* é capaz de auxiliar na geração novas ideias centradas no humano, por meio de técnicas de pesquisa e prototipação, de maneira rápida e multidisciplinar. Os métodos ágeis são utilizados de maneira amplamente divulgada e estudada como método de desenvolvimento de *software*. Esta pesquisa foi realizada por meio de estudo de caso múltiplo do tipo incorporado, com coleta de dados por meio de pesquisa qualitativa, em empresas ou departamentos na área de tecnologia, nas quais essa tecnologia é utilizada como principal ferramenta para gerir e gerar negócios. No estudo foram encontrados três tipos diferentes de integração entre os modelos, de modo sequencial, sendo que o primeiro possuía uma etapa final de definição da solução e o início do desenvolvimento do *software*; o segundo não possui essa característica, o terceiro modelo era a aplicação do *Design Thinking* e dos métodos ágeis de modo conjunto e simultâneo, isso como um único processo. Estes modos diferentes de integração visavam atender as características dos projetos com relação à complexidade e porte, além das incertezas em que as empresas estiveram sujeitas durante o curso do projeto.

PALAVRAS CHAVES: *Design Thinking*, Métodos Ágeis, Desenvolvimento de *Software*, Integração, Gestão de Projetos.

ABSTRACT

Design Thinking and agile methods, when used in an innovative and dynamic environment, aiming the creating products and solutions to support business and to increase satisfaction, bringing solutions to the current problems and that also generate future business for the company. This study was carried out with the objective of verifying how the companies that adopted the *Design Thinking* and the agile methods did to integrate it, from the creation of the solution of the initiatives to be development-using *software*. *Design Thinking* is able to aid in the generation of new ideas centered on the human, through research and prototyping techniques, in a fast and multidisciplinary way. Agile methods are widely used and studied as a method of *software* development. This study was carried out through a multiple case study with data collection for a qualitative research, in companies or in technology departments, where this technology is used as the main tool to manage and generate business. On this research was found three different types of integration between the models, sequentially. The first with a queue between the final solution definition stage and the beginning of *software* development, the second, without a queue between the final solution definition stage and the beginning of *software* development, the third model is the use of *Design Thinking* and agile methods together and simultaneously, as a single process. These different integration models aim to meet the projects needs regarding to complexity and size, as well as the uncertainties in which companies are subject throughout project life cycle.

KEYWORDS: *Design Thinking*, Agile Methods, *Software* development, Integration, Project Management

LISTA DE TABELAS

Tabela 1 - Frameworks e Técnicas Utilizadas.....	79
Tabela 2 - Características do <i>Design Thinking</i> Encontradas	80
Tabela 3 - Priorização e Ideação.....	81
Tabela 4 - Desenvolvimento.....	82

LISTA DE FIGURAS

Figura 1 - Publicações por ano	18
Figura 2 - Comparação entre as linhas de pensamento.	22
Figura 3 - <i>Double Diamond</i>	23
Figura 4 - <i>Design Thinking</i>	25
Figura 5 - Ciclo de Desenvolvimento <i>Scrum</i>	33
Figura 6 - Integração <i>Design Thinking</i> e Desenvolvimento Ágil.....	37
Figura 7 - Etapas da Pesquisa.....	39
Figura 8 - Resumo da Qualificação da Pesquisa	39
Figura 9 - Relação das Proposições de Estudos	41
Figura 10 - Relação Entre Proposições, Objetivos e Questões	41
Figura 11 - Regra de Validação das Proposições	42
Figura 12 - Tipos Básicos de Projetos para Estudo de Caso	42
Figura 13 - Perfil dos Entrevistados	45
Figura 14 - Entrevistados por Tipo de Empresa.....	47
Figura 15 - Entrevistados por Função	47
Figura 16 - Escolaridade dos Entrevistados	48
Figura 17 - Experiência dos Entrevistados em anos.....	48
Figura 18 - Processo de Desenvolvimento - Empresa Grande Porte.....	52
Figura 19 - Tabela de Informações da <i>Users Stories</i>	57
Figura 20 - Kanban de Desenvolvimento	60
Figura 21 - Processo de Desenvolvimento Empresa Médio Porte	61
Figura 22 - Modelo de Desenvolvimento - <i>Software House</i>	67
Figura 23 - Processo de Desenvolvimento - <i>Startup</i>	72

Figura 24 - <i>Product Backlog</i> - <i>Startup</i>	73
Figura 25 - Canvas.....	74
Figura 26 - Matriz SWOT da <i>Startup</i>	75
Figura 27 - Comparação do Processo de Desenvolvimento	84
Figura 28 - Métodos e Técnicas utilizadas pelas empresas	88
Figura 29 - Modelo Integrado - <i>Design Thinking</i> e Métodos Ágeis.....	91

Sumário

1.	INTRODUÇÃO	14
1.1.	PROBLEMA DE PESQUISA	16
1.2.	OBJETIVO	17
1.3.	JUSTIFICATIVA PARA O ESTUDO DO TEMA	17
1.4.	ESTRUTURA DO TRABALHO.....	19
2.	REFERENCIAL TEÓRICO	20
2.1.	<i>DESIGN THINKING</i>	20
2.2.	MÉTODOS ÁGEIS	25
2.3.	SÍNTESE DA FUNDAMENTAÇÃO TEÓRICA E PROPOSIÇÕES.....	33
2.3.1.	<i>DESIGN THINKING</i>	33
2.3.2.	MÉTODOS ÁGEIS	34
2.3.3.	INTEGRAÇÃO DO <i>DESIGN THINKING</i> COM OS MÉTODOS ÁGEIS.	36
3.	METODOLOGIA E TÉCNICAS DE PESQUISA.....	38
3.1.	DELINEAMENTO DA PESQUISA	38
3.2.	MÉTODO DE PESQUISA E UNIDADE DE ANÁLISE.....	42
3.3.	COLETA DE DADOS.....	43
3.4.	ANÁLISE DOS DADOS	48
3.5.	LIMITAÇÕES DA PESQUISA	50
4.	ANÁLISE DOS DADOS	51
4.1.	EMPRESA GRANDE PORTE	51
4.2.	EMPRESA MÉDIO PORTE.....	60
4.3.	<i>SOFTWARE HOUSE</i>.....	66
4.4.	<i>STARTUP</i>	71

4.5.	COMPARAÇÃO ENTRE OS MODELOS DAS EMPRESAS	78
4.5.1.	<i>FRAMEWORKS</i> E TÉCNICAS ÁGEIS UTILIZADAS	78
4.5.2.	UTILIZAÇÃO DO <i>DESIGN THINKING</i>	79
4.5.3.	PRIORIZAÇÃO E IDEAÇÃO DAS INICIATIVAS	80
4.5.4.	DESENVOLVIMENTO DO <i>SOFTWARE</i>	82
4.5.5.	COMPARAÇÃO DOS FLUXOS	83
4.6.	ANÁLISES DAS PROPOSIÇÕES	85
4.7.	NOVO MODELO PROPOSTO	90
5.	CONTRIBUIÇÕES PARA A PRÁTICA	93
6.	CONCLUSÃO.....	94
6.1.	LIMITAÇÕES E CONTRIBUIÇÕES PARA ESTUDOS FUTUROS.....	96
	REFERÊNCIAS	97
	APÊNDICE I – PROTOCOLO DE PESQUISA	104

1. INTRODUÇÃO

Num ambiente competitivo de um mercado globalizado, as organizações necessitam de produtos de qualidade que atendam a demanda dos consumidores e, também, eficiência em seu processo de inovação (Flynn, Dooley, O'Sullivan, & Cormican, 2003). As organizações que possuem capacidade de inovar implementando novas ideias, seja em novos produtos ou melhorando o já existente, ganham destaque sobre seus concorrentes (Baregheh, Rowley, & Sambrook, 2009). Assim, para uma organização se desenvolver, ela depende de sua capacidade de gerar ideias e as produzir de maneira eficaz (Flynn et al., 2003).

As empresas focam em criar novas ideias e as colocar em condições de produção para sua exploração comercial. Na fase de implementação, após as ideias serem aprovadas, o objetivo é transformá-las em produto. Nesta fase, os modelos de gestão a serem utilizados devem suportar abordagens desde as mais simples até as mais complexas, que contenham suporte ao gerenciamento do projeto aderente e ferramentas ágeis de gestão do ciclo de vida do produto e projeto (Westerski, Iglesias, & Nagle, 2011). Entretanto, mesmo dando grande importância a gestão de projetos, os projetos de desenvolvimento de *software* estão sendo executados em larga escala na sociedade, com investimentos ultrapassando a casa dos trilhões de dólares anualmente, e ainda persistem as altas as taxas de insucesso (Serrador & Pinto, 2015; The Standish Group, 2011).

O processo de desenvolvimento de *software* consiste em um conjunto de atividades que são realizadas para projetar, desenvolver, implantar e manter os sistemas de *software*. Com o objetivo de gerir esse conjunto estruturado de atividades, foram criados ao longo dos anos, muitos modelos de desenvolvimento. Estes modelos têm preocupações associadas ao custo, tempo e a qualidade do produto (Kaur & Sengupta, 2011).

Uma vez que as empresas vêm buscando sistemas cada vez mais inovadores e eficientes, as áreas de negócio das empresas tendem a se aproximar e ter mais envolvimento nos processos de desenvolvimento dos produtos de *software*. Com isso, as solicitações de mudança são antecipadas para a fase de desenvolvimento e também em maior volume (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Os métodos ágeis possuem bastante eficácia em ambientes voláteis, nos quais são necessários absorver grande número de mudanças no decorrer do projeto e apresentar resultados de negócio cada vez mais rápido. Nos modelos de desenvolvimento ágil, porém, as tarefas de análise e *design* possuem menos importância (Abrahamsson, Warsta, Siponen, & Ronkainen, 2003; Kaur & Sengupta, 2011).

Os problemas de *design* geralmente são originados a partir da má definição, ou ainda, da inexistência dos requisitos e funcionalidades desejadas pelos clientes. As diferentes opções de construção ou entendimento da necessidade podem não ser articuladas e não terem a transparência desejada a todos os *stakeholders*. Nestas situações são criadas divergências que geram muitas opções de soluções, nem sempre aderente às necessidades. Quando se aplica o conceito de *design*, a equipe do projeto está apreendendo, fazendo algo que pode ser materializado por meio de protótipos ou modelos, como ferramentas para pensar (Brown, 2008; Brown & Katz, 2011; Cross, 2001b).

Quando se inicia o processo de projetar uma nova solução, fundamentalmente inicia-se pelo *design*, pelo seu desenho, pois é uma atividade humana básica. O processo de *design* começa quando se inicia um projeto, com um objetivo esperado (Cahyadi & Prananto, 2015; Cross, 2001b).

O conceito do *Design Thinking* é uma maneira de pensar estrategicamente como *design*, por meio de uma abordagem holística que visa a inovação, utilizando as características de um *designer* para atender as necessidades dos usuários e com base tecnológica, com o

objetivo de criar algo com viabilidade técnica, estratégica ao negócio, gerando valor ao cliente (Brown, 2008; Brown & Katz, 2011; Cahyadi & Prananto, 2015; Dorst & Cross, 2001).

1.1. PROBLEMA DE PESQUISA

Todos os projetos estão sujeitos a incertezas. Em muitos casos são tomadas decisões em condições adversas e não alinhadas com as expectativas de todos os *stakeholders*, com relação ao escopo e aos requisitos, premissas e riscos do projeto. Estas incertezas podem impactar diretamente, positiva ou negativamente, a satisfação dos usuários e clientes do *software* desenvolvido e colocando o sucesso do projeto em risco (Perminova, Gustafsson, & Wikström, 2008; PMBOK, 2013).

O uso do *Design Thinking* serve como processo de levantamento de informações relevantes para a definição dos requisitos, por meio de reuniões e interação entre as partes envolvidas e visão integrada a todo o projeto de modo global (Brown, 2008; Johansson-sköldberg & Woodilla, 2013). Aliando-se a esta abordagem, o modelo de desenvolvimento de *software* que utiliza métodos ágeis, permite um processo de validação durante toda fase de desenvolvimento do projeto e grande chance de absorver mudanças (Thummadi, Shiv, Berente, & Lyytinen, 2011).

Partindo destas premissas, evidencia-se a questão de pesquisa que é: **Como o *Design Thinking* pode ser integrado com os métodos ágeis em projeto de desenvolvimento *software*?**

1.2. OBJETIVO

Normalmente, o *Design Thinking* é utilizado na fase de concepção das ideias para definir as características do produto ou serviço a ser desenvolvido pelo projeto. No caso de desenvolvimento do *software*, o processo pode utilizar métodos ágeis. Portanto pretendeu-se, neste estudo, identificar como o *Design Thinking* pode ser integrado com os métodos ágeis.

Para atender o objetivo principal, foram definidos os seguintes objetivos específicos:

OS_1 - Determinar as características do *Design Thinking* que são utilizadas pelas organizações pesquisadas em projetos de *software*;

OS_2 - Identificar as características ágeis que são utilizadas pelas organizações pesquisadas em projetos de *software*;

OS_3 - Identificar como ocorre a integração entre o *Design Thinking* e os métodos ágeis nas organizações pesquisadas;

1.3. JUSTIFICATIVA PARA O ESTUDO DO TEMA

Como pode ser visto na Figura 1, o estudo do tema *Design Thinking* tem crescido nos últimos anos, conforme dados de estudos realizados em processos de inovação de produtos e solução de problemas no cotidiano das empresas e também das pessoas (Góes, Russo, & Felipe, 2015).

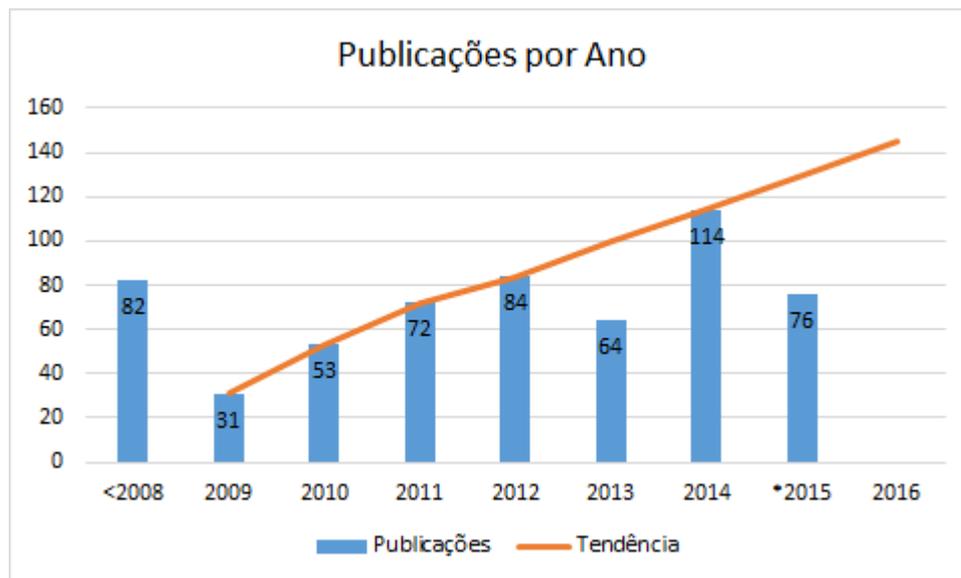


Figura 1 - Publicações por ano
 Fonte: Adaptado de Góes, Russo, & Felipe (2015)

Por outro lado, os modelos de desenvolvimento baseados nos princípios *Lean*, continuam a ganhar espaço no mercado, inclusive ultrapassando suas origens, que é o desenvolvimento de *software* (Dybå & Dingsøyra, 2008; Serrador & Pinto, 2015). Este modelo é servido de técnicas e conceitos que envolvem os *stakeholders* desde o início do projeto (Abrahamsson et al., 2003).

Apesar de ambos os modelos possuírem características semelhantes, os estudos publicados nos principais periódicos demonstram que a integração entre eles é escassa, justificando a importância do estudo de maneira científica, para mostrar a aplicação que já é utilizada pelo mercado. Dos artigos encontrados, destacam-se os artigos “*Pet Empires: Combining Design Thinking, Lean Startup and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment*” que destaca a integração entre os temas em empresas *Startups* (Paula & Ara, 2016), e também o artigo “*An Integrated Framework for Design Thinking and Agile Methods for Digital Transformation*” na qual é descrita a integração entre os modelos por meio de teorias, porém, não foi realizada pesquisa de campo para testar o modelo proposto (Gurusamy, Srinivasaraghavan, & Adikari, 2016).

Nesta era digital, os consumidores são altamente dependentes de soluções tecnológicas em o seu dia-a-dia. Isso leva as empresas a repensar nas novas necessidades dos clientes e realizar, além de colocar no mercado o mais rápido possível, de maneira assertiva, para competir com seus concorrentes comerciais. A maioria das organizações tem sido submetidas à transformação digital de alguma forma, a fim de fornecer um processo rápido de desenvolvimento de *software* para melhor atendimento ao cliente. Porém, este processo tem sido feito por meio de modelos de processos não padronizados e/ou tentativas empíricas de aprendizagem, gerando grandes custos financeiros para as empresas (Gurusamy et al., 2016).

1.4. ESTRUTURA DO TRABALHO

A estrutura deste trabalho é composta por quatro tópicos. O segundo e próximo tópico trata do referencial teórico e é subdividido em três partes: Métodos Ágeis, *Design Thinking* e a integração entre eles. No terceiro tópico, são definidos os procedimentos metodológicos de pesquisa, sendo descrito o delineamento da pesquisa, os procedimentos de coleta e de análise dos dados e as limitações. No quarto tópico será feita a apresentação dos dados colhidos durante o estudo, separado por empresa, e ao final será apresentado um breve resumo comparativo entre os modelos das empresas. Também no quarto tópico, será feita a apresentação e validação das proposições e o modelo de integração atualizado após a coleta das informações. No quinto tópico, serão apresentadas as contribuições do estudo para a utilização nas empresas (utilização prática) e, por fim, no sexto tópico, será feita a conclusão.

2. REFERENCIAL TEÓRICO

Nos subtópicos abaixo serão descritos os constructos no estudo: *Design Thinking* e métodos ágeis. No primeiro item, será tratado o histórico da abordagem do *Design Thinking*, as diferentes abordagens para cada público alvo (empresas de *design*, educadores e pesquisadores) e posteriormente, a definição da ferramenta. No segundo item será descrito o histórico dos métodos ágeis, as principais técnicas e modelos e por fim, um modelo completo que integra as diferentes técnicas. Concluindo, será apresentada uma síntese dos constructos citados, de modo a definir o escopo do estudo e suas proposições.

2.1. DESIGN THINKING

Design Thinking é um conceito aplicado de modo teórico e prático para criação de novos produtos, soluções e também no contexto de gestão, que está fortemente relacionado com a prática (Johansson-sköldberg & Woodilla, 2013). Para este tema existe vasta literatura acadêmica e profissional, publicações em revistas, livros e meios eletrônicos mais diversos. Analisando a literatura acadêmica, foi possível verificar que não existe desenvolvimento sustentável do conceito. Existem discursos acadêmicos de *design* e também discursos em gestão de projetos, porém, sem fazer ligação entre os dois. É como se teóricos de *design* e gestão descrevessem processos completamente independentes um do outro. É possível encontrar artigos sugerindo que o *Design Thinking* é algo passageiro para ser usado no mundo da gestão (Johansson-sköldberg & Woodilla, 2013).

O conceito de *design* pode ser dividido em dois discursos (Johansson-sköldberg & Woodilla, 2013): (i) *Designerly Thinking* é o conhecimento adquirido pelo profissional de mercado por meio de estudos e pesquisas científicas, unindo teoria e prática a partir da

perspectiva de *design*; (ii) *Design Thinking*, neste aspecto o conceito de *design* é levado além das fronteiras do *design* onde se inclui além da arte e arquitetura, também profissionais sem formação acadêmica de *design*, especialmente na gestão. Resumindo, o *Design Thinking* é uma versão simplificada do *Designerly Thinking* ou ainda, uma maneira de integrar as práticas de *design* a gestão.

A abordagem do *Design Thinking* possui três vertentes conforme demonstrado na Figura 2. A primeira é a abordagem proposta pela IDEO por Brown (2008), que possui foco para aplicação do *Design Thinking* em empresas e negócios. As outras duas vertentes utilizam o modelo para fins educacionais e de aprendizagem, sendo que a abordagem de Roger Martin aplica pesquisas acadêmicas de modo a comprovar o fato e as hipóteses da mesma maneira dos autores Richard Boland & Fred Collopy (2004), que se diferenciam apenas pela base acadêmica.

Origem	Público Alvo	Perfil	Conexão Acadêmica	Relação com a Prática
Brown (2008)	Empresa de <i>Design</i> IDEO	Gestores	Experiência em vez de pesquisa Conexões com a inovação	Como a IDEO faz <i>Design Thinking</i> ? Como alguém pode usar o <i>Design Thinking</i> ?
Martin (2007)	Educadores Gestão de Empresas	Casos de sucesso para demonstrar a teoria	Fundamentada na ciência cognitiva, gestão Baseado em teorias de planejamento	Como empresa que utilizam o <i>Design Thinking</i> tem a produção bem-sucedida? Como qualquer empresa pode projetar sem <i>design</i> ?

Origem	Público Alvo	Perfil	Conexão Acadêmica	Relação com a Prática
Boland & Collopy (2004)	Pesquisadores acadêmicos & Educadores.	Ensaio curtos, onde são estabelecidos (Gestão) estudos para aplicar sua perspectiva teórica do <i>design</i>	Fundamentada no indivíduo pesquisadores próprios perspectivas teóricas, inspirado por Gehry de prática arquitetônica ou entre em contato com o projeto	<i>Design Thinking</i> como analogia & alternativa

Figura 2 - Comparação entre as linhas de pensamento.
 Fonte: Adaptado (Johansson-sköldberg & Woodilla, 2013)

Uma das primeiras publicações que utilizou o termo “*Design Thinking*” foi o artigo “*Wicked problems in Design Thinking*” no ano de 1992. A escola “*Koln International School of Design*” (KISD) iniciou a primeira turma de *design* de serviços, que tinha como objetivo aplicar o *Design Thinking* na solução de problemas orientado a serviços. Em 1999, a IDEO incorporou o tema em seu portfólio e, apenas em 2008, a *Harvard Business Review* publicou o artigo mais relevante sobre o tema “*Design Thinking*” do autor Tim Brown, CIO da IDEO (Alt & Pinheiro, 2011).

Muitas marcas bem-sucedidas utilizam ideias inovadoras que são inspiradas na compreensão dos consumidores. Estas empresas utilizam os métodos do *Design Thinking* para inovar e criar valor. O *Design Thinking* é uma ferramenta para imaginar soluções, de modo que as experiências sejam testadas conseguindo atender grandes diferenças de condições socioeconômicas e culturais (Brown, 2008).

A abordagem *Design Thinking* remete-se aos processos cognitivos, que são manifestados na ação de *design* (Cross, 2001a). *Design Thinking* é a abordagem na qual o objetivo é a inovação centrada no ser humano, isto é, a inovação é a compreensão e análise por meio de observação direta, de modo que problemas do cotidiano sejam resolvidos determinando, além da solução, o processo produtivo, comercial e logístico, levando-se em consideração as necessidades e preferências dos consumidores do produto desenvolvido

(Brown, 2008). Esta abordagem deriva também de interações com outras pessoas e com o uso de ferramentas, e não apenas originada dentro de um único *designer* (Boland & Collopy, 2004).

Com o objetivo de identificar como o *Design Thinking* era utilizado pelas empresas, o *Design Council* criou o modelo *Double Diamond* devido às características encontradas de expansão, posterior ao entendimento inicial e de maneira cíclica. É dividido em quatro etapas: descobrir, definir, desenvolver e entregar, conforme destacado na Figura 3 (Alt & Pinheiro, 2011; Council, 2005). Este modelo é aderente ao conceito dos métodos ágeis com relação a empatia, colaboração, experimentação e interação (Alt & Pinheiro, 2011).

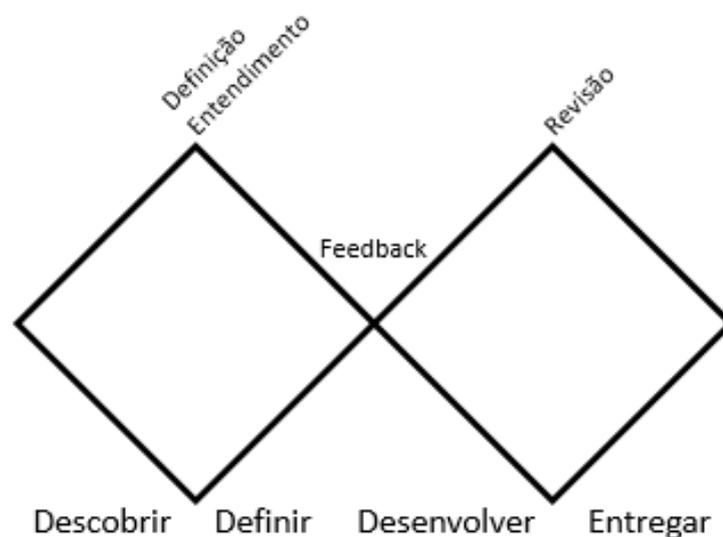


Figura 3 - *Double Diamond*
Fonte: adaptado de (Council, 2005)

O modelo *Design Thinking* proposto pela IDEO, traz uma significativa mudança com relação ao modelo de desenvolvimento tradicional de desenvolvimento de produtos onde, ao invés dos *designers* desenvolverem uma ideia para os consumidores, as empresas optam por solicitar aos *designers* ideias e necessidades de consumidores mais exigentes. Esta é uma mudança na estratégia de abordagem dos consumidores. A primeira tem um papel mais tático,

que gera soluções com valor limitado e a segunda, mais estratégica, leva a novas formas de gerar valor. Além disso, a inovação não está mais focada apenas na criação de novos produtos, mas também na prestação de serviços, como os serviços de TI, exatamente nas atividades centradas no ser humano, onde o *Design Thinking* pode fazer a diferença de forma decisiva (Brown, 2008).

Apesar do significativo investimento em tempo, a equipe envolvida, multidisciplinar, pode discutir diferentes pontos de vista, colhendo *insights* recolhidos com a observação, exploração de potenciais soluções por meio de *brainstorming* e prototipagem. A prototipagem não necessariamente deve ser cara e complexa, podendo utilizar desenhos e materiais de baixo custo e até mesmo rudimentares, que estejam disponíveis. Também não se deve desprender grandes volumes de tempo na construção destes protótipos, mas sim, deve-se ter como objetivo, gerar uma ideia do conceito para iniciar as discussões e coleta de *feedback* (Brown, 2008).

Acredita-se que a solução de um problema é gerada por “um membro dotado de mente brilhante, capaz de proezas inimagináveis [...]” (Brown, 2008). Já uma equipe de *designers* não é um avanço repentino, mas um avanço gradual, resultante de muito trabalho, sustentado por um processo criativo de descoberta, ciclos interativos de protótipos, testes e refinamento, sempre centrado no ser humano (Brown, 2008).

Conforme demonstrado na Figura 4, os projetos de *design* passam por três fases de desenvolvimento: (i) inspiração, que é a motivação ou necessidade de uma solução, (ii) ideação, que é o processo de criar e testar ideias que nos levem a solução e, por fim, (iii) implementação, que é o processo para divulgação da solução e a abertura para o mercado. Os projetos passam em forma de *looping* sobre essas fases, até que a solução esteja madura, refinada, e que sejam disponibilizadas para a utilização ou comercialização (Brown, 2008).



Figura 4 - *Design Thinking*
Fonte: Adaptado de (Brown, 2008)

Em geral, o *Design Thinking* na gestão é menos pragmático e possui poucas contribuições, se comparado ao *Designerly Thinking*. Porém, é muito mais jovem e tem crescido rapidamente, uma vez que pode ser o caminho para os gestores entenderem melhor o projeto, de modo mais simples do que a maneira como a gestão de projeto é constituída, por meio da plataforma tradicional de gestão. O modelo permite uma abordagem em que é permitido e compreensível que os *designers* e gestores sejam incentivados a compartilhar o conhecimento e juntos, sugerirem a maneira de progredir com o projeto (Johansson-sköldberg & Woodilla, 2013).

2.2. MÉTODOS ÁGEIS

No início dos anos 70 foi criado o modelo de desenvolvimento de *software* cascata (*waterfall*), no qual as fases de desenvolvimento (análise de requisitos, desenvolvimento, testes e manutenção) são sequenciais (Royce, 1970). Esse modelo não é ideal em ambientes voláteis e dinâmicos, pois mudanças são geradas por inúmeros fatores, dificultando o

atendimento dos objetivos previamente definidos no projeto com o cliente (Dybå & Dingsøyra, 2008).

O método ágil originou-se em meados da década de 1980, mais precisamente no ano de 1986, com a abordagem *Scrum* de Nonakata e Takeushi (Abrahamsson et al., 2002). Esse método tem como base o conceito “*Lean*” que foi amplamente divulgado os anos 1980 e 1990 (Abrahamsson et al., 2002). Esse modelo de produção possui características de gerenciamento de projetos mais simples. Foi muito disseminado nos EUA, com o objetivo de tornar a indústria automobilística americana mais competitiva com o uso desse método de produção utilizado pela Toyota no Japão (Hasle, Bojesen, Jensen, & Bramming, 2012).

O método ágil é descrito no documento chamado “*Scrum Guide*”, que teve sua última atualização em 2016 (Schwaber & Sutherland, 2016). Os métodos ágeis não se resumem apenas ao *Scrum*, mas também a outros modelos como *Crystal Clear*, *Extreme Programming (XP)*, *Adaptive Software Development (ASD)*, *Feature Driven Development (FDD)* e *Dynamic Systems Development Method (DSDM)*.

Todos esses modelos de desenvolvimento de *software* ágil são caracterizados por serem incrementais, interativos, simples e adaptáveis (Schwaber & Sutherland, 2013), fazendo com que os modelos tenham características diferentes e aplicações distintas. Porém, todos possuem o princípio de geração de valor, baseado na afirmação: trazer mais valor ao cliente, o mais rápido possível, e com qualidade, mantendo flexibilidade para absorver as mudanças necessárias para o sucesso do projeto. (Abrahamsson et al., 2003; Boehm & Turner, 2003, 2005; Michels & Ferreira, 2013; Schwaber & Sutherland, 2013; Serrador & Pinto, 2015).

Este modelo difere dos métodos ágeis principalmente com relação a organização das suas atividades e a frequência das entregas (Thummadi et al., 2011). Nos métodos ágeis, o projeto depende fortemente de interação com o cliente durante toda a sua execução, com a

respectiva validação do produto, diferente do modelo cascata, no qual a validação é feita apenas na entrega final (Thummadi et al., 2011). Esse processo exige uma negociação com o cliente sobre vários aspectos como escopo e prazo de entrega.

Diante das constantes mudanças existentes nos projetos de desenvolvimento de *software* e, aliada à necessidade de se desenvolver projetos cada vez mais aderentes às necessidades de negócios dos clientes, a utilização dos métodos ágeis vem ganhando espaço no mercado (Abrahamsson & Koskela, 2004; Abrahamsson et al., 2002; Boehm & Turner, 2003).

O processo de desenvolvimento de *software* cobre a sequência de passos a serem seguidos pela empresa, desde a concepção da ideia até a comercialização ou utilização do produto desenvolvido, que é chamado de ciclo de vida do *software*. Os processos utilizados devem ser eficientes com relação a tempo e consumo de recursos. Para isso, é de fundamental importância o gerenciamento de projetos, visto como um fator relevante pelos métodos ágeis, de modo que permita a correta execução das atividades de desenvolvimento (Abrahamsson et al., 2003).

No início dos anos 2000, um grupo de pessoas chamado de “A Agile Alliance” se reuniu para discutir como poderia melhorar o processo de desenvolvimento de *software*. Entre eles estavam Jeff Sutherland, Ken Schwaber e Mike Beedle, que são os autores do modelo de desenvolvimento *Scrum*. Este encontro resultou em um manifesto chamado de “Manifesto Ágil” o qual descreve as intenções de criar um modelo de desenvolvimento de *software*, influenciado pela indústria japonesa e com princípios enxutos (“Manifesto for Agile Software Development,” 2015; Schwaber & Sutherland, 2013).

Também nesta mesma época, início dos anos 2000, os projetos de desenvolvimento de *software* utilizavam o modelo tradicional chamado de “*Waterfall*”, criado no ano de 1970, que consiste em segmentar cada fase do projeto em etapas independentes e padronizadas,

permitindo que o projeto inteiro seja planejado de maneira organizada e programada, mantendo a rastreabilidade entre todos os artefatos e garantindo assim a previsibilidade do projeto. Porém, as características deste modelo dificultam ou impedem que mudanças sejam feitas durante a evolução do projeto (Kruchten, 2013; Royce, 1970). Este processo rígido, que inclui retrabalho excessivo, falta de flexibilidade e insatisfação do cliente, pode resultar em problemas sérios, culminando num projeto sem sucesso ou mal sucedido (Serrador & Pinto, 2015).

Em 2002, os métodos de desenvolvimento ágeis, mesmo que ainda incipiente, geravam curiosidade nos pesquisadores e engenheiros de *software* do mundo inteiro e, nesta época os estudos sobre os métodos ágeis, eram escassos (Abrahamsson et al., 2003). Atualmente, o cenário mudou e já existem muitos estudos mostrando os resultados gerados, quando são aplicados os métodos ágeis.

Um dos aspectos mais importantes da pesquisa e da utilização dos métodos ágeis se faz necessário com forte atuação prática, de modo que é necessário um suporte empírico para evidenciar e fundamentar a utilização de cada um dos processos (Abrahamsson et al., 2003).

O Desenvolvimento de *software* adaptativo (*ASD - Adaptative Software Development*) é um modelo de desenvolvimento de *software*, que permite que o paradigma da adaptação à realidade do projeto e características da empresa seja superado, de modo que a criatividade e a adaptação sejam ferramentas para a solução de problemas em projetos complexos e em ambientes caóticos (Abrahamsson et al., 2003).

A modelagem ágil também traz outra inovação com relação ao modelo tradicional. Ao invés de dar grande importância à comunicação escrita, por meio das especificações técnicas e funcionais, no modelo ágil é utilizada uma comunicação direta entre os desenvolvedores e o cliente final, levando em consideração, inclusive, o aspecto cultural, onde o objetivo é incentivar os desenvolvedores a criar soluções por meio de modelos simples, que gerem valor

ao usuário e que sejam suficientemente completas, de modo que apoiem as necessidades de *design* e não abafem a criatividade de cada indivíduo (Abrahamsson et al., 2003).

O modelo *Crystal Clear* é um conjunto de métodos utilizados de maneira combinada para cada tipo de projeto, ou para características específicas de cada projeto. Este conjunto de métodos tem como objetivo definir, por meio de cores, o peso de cada um dos itens a ser desenvolvido, definindo sua priorização ou ordem de desenvolvimento. O modelo também permite a integração com técnicas do modelo XP e *Scrum* (Abrahamsson et al., 2003).

O método de desenvolvimento de sistemas dinâmico (*DSDM - Dynamic Systems Development Method*) teve sua origem no Reino Unido no início dos anos 90, e trouxe outra inovação em relação ao método tradicional: propõe que, ao invés de fixar o escopo e, posteriormente, os recursos e o tempo necessário para implementação, parte-se do tempo e recurso fixos, ajustando a quantidade de escopo que é possível absorver com os recursos e tempo disponíveis. Este é o primeiro modelo de desenvolvimento reconhecidamente ágil (Abrahamsson et al., 2003).

Outro importante conjunto de ferramentas é conhecido como *Extreme Programming* ou apenas XP. O objetivo deste método é garantir que o processo de desenvolvimento produza requisitos de qualidade. É baseado em práticas individuais e devem ser alinhadas para o trabalho coletivo. Estão entre as principais características: o método de programação em par, interações recorrentes em curto espaço de tempo com *feedback* rápido, ajustes que são feitos de maneira rápida, integração contínua de código e testes. Esse modelo pode ser combinado com o modelo ISD (*Internet Software development*) que também busca o processo de desenvolvimento, onde a entrega constante e rápida, na velocidade em que são desenvolvidos, são entregues para o usuário (Abrahamsson et al., 2003). Em 2004 foi realizado um estudo sobre o papel do cliente, onde foi descoberto que o maior tempo, cerca de 43% do esforço, foi gasto com reuniões de planejamento, testes de aceitação e reuniões de retrospectiva ao

final do ciclo, o que demonstra a vontade de reduzir os processos formais que até então eram usados (Serrador & Pinto, 2015).

O desenvolvimento baseado em funcionalidades é proposto pelo método FDD (*Feature Driven Development*). Neste método o processo de desenvolvimento é guiado por funcionalidades críticas ao negócio, focando as fases de concepção e construção do *software*. O modelo também é orientado a interações curtas enfatizando a qualidade do produto, permitindo entregas constantes e tangíveis ao negócio (Abrahamsson et al., 2003).

Concluindo a conceituação dos métodos ágeis, aparece o *framework Scrum*. Este é o *framework* mais importante dos métodos ágeis, sendo responsável por estruturar os demais modelos citados de maneira coesa e com objetivo único. Este *framework* foi criado por Jeff Sutherland e Ken Schwaber. O modelo é uma abordagem empírica, baseado nos pilares dos métodos ágeis (flexibilidade, adaptabilidade e produtividade). É um modelo simples de entender e complexo de ser utilizado, onde o processo de desenvolvimento é flexível e permite que o time de desenvolvimento defina o processo de implementação. Seus processos são atividades de gestão, que tem como objetivo buscar sempre a melhoria contínua, visando aumentar a produtividade, reduzindo as ineficiências e impedimentos (Abrahamsson et al., 2003; Schwaber & Sutherland, 2016).

Atualmente, métodos ágeis continuam em alta no mercado (Serrador & Pinto, 2015), inclusive, em julho de 2016 foi lançada a nova versão do “*Scrum Guide*”. Neste documento estão descritas as definições do *Scrum*, seus papéis, eventos, artefatos e processos.

O *Scrum* é sustentado por três pilares de controle de processo:

- **Transparência:** As informações relevantes aos responsáveis pelo resultado devem estar disponíveis e visíveis, seguindo um padrão claro e comum, para que os observadores entendam e interpretem o que está sendo demonstrado.

- **Inspeção:** As pessoas envolvidas no processo frequentemente devem inspecionar os processos envolvidos, detectar variações e aprender com erros e acertos. Esta característica também é vista por outros modelos gestão e guias de referência de gestão de projetos (PMBOK, 2013).
- **Adaptação:** Os aspectos do processo e do produto, após passar por inspeções e apresentarem desvios fora dos aceitáveis, devem ser ajustados o mais breve possível.

Os times *Scrum* são auto-organizáveis, uma vez que o próprio time define a melhor maneira de completar o seu trabalho. Também devem ser multifuncionais, ou seja, todos do time devem possuir competências necessárias para completar o trabalho, sem depender dos outros. Os times *Scrum* são compostos apenas por três papéis: Time de Desenvolvimento, *Product Owner* e *Scrum Master*.

- ***Product Owner*:** é responsável por definir e priorizar o que será feito dentro de cada *Sprint*, de modo que o valor do produto criado seja o maior possível e tenha o melhor retorno sobre o investimento.
- **Time de desenvolvimento:** é composto por profissionais que efetivamente irão realizar as tarefas. São os desenvolvedores, analistas e testadores. O tamanho ideal dos times deve ser superior a 3 (três) e menor que 9 (nove) integrantes. Se os times forem menores que isso, não terão condições de entregar um incremento de produto dentro da *Sprint* e, no caso de times com mais integrantes que o recomendado, será necessária muita coordenação entre tarefas e pessoas.
- ***Scrum Master*:** é o profissional responsável para garantir que o processo seja cumprido e entendido. Ele não possui papel de gestor do projeto e deve atuar

como apoiador do time de desenvolvimento e também orientando o *Product Owner*.

Os eventos prescritos no *framework* têm o objetivo de criar uma rotina, de maneira organizada e planejada, com dias, horários e duração fixos. Cada evento é um *time-box*, o que garante que os eventos não ocupem tempo demais e também não deixem de ser realizados. A *Sprint*: é o evento mais importante do *Scrum*, tem duração fixa de no máximo 1 (um) mês e sua principal entrega é um produto ou versão utilizável. Cada *Sprint* se inicia imediatamente após o término da outra. É composta por outros eventos, também com tempo fixo e com procedimento definido como reunião de planejamento, reunião diária, trabalho de desenvolvimento, revisão do produto e retrospectiva da *Sprint*.

Os artefatos do *Scrum* representam o trabalho ou o valor para sustentar as características de transparência, oportunidade de inspeção e adaptação do modelo.

- ***Backlog do Produto***: é uma lista ordenada de todas as características que o produto deve possuir, dentre elas: funções, requisitos, melhorias e correções.
- ***Backlog da Sprint***: é o conjunto de itens priorizados do *backlog* do produto escolhido para ser desenvolvido dentro do *time-box* da *Sprint*.

Na Figura 5 é demonstrado o resumo dos principais eventos e características do *Scrum*. Os conjuntos de características apresentados pelo *Scrum Guide* descritas acima, são imutáveis, embora seja possível a utilização de apenas partes do modelo e agregando outras, modelos ágeis, técnicas e práticas.

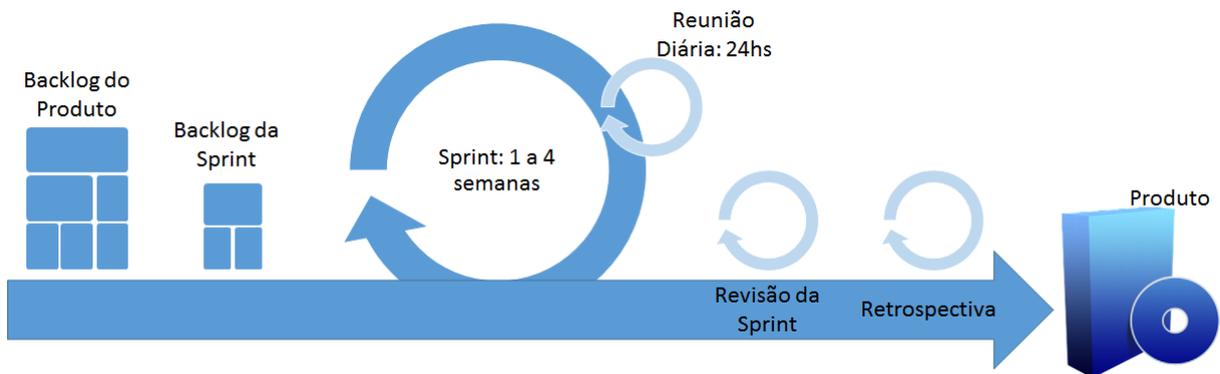


Figura 5 - Ciclo de Desenvolvimento *Scrum*
 Fonte: Elaborado pelo Autor

2.3. SÍNTESE DA FUNDAMENTAÇÃO TEÓRICA E PROPOSIÇÕES

Neste item serão apresentados os principais pontos que serão explorados conforme a fundamentação teórica e também as proposições de estudos (Yin, 2015), com o objetivo de definir o escopo da pesquisa.

2.3.1. *Design Thinking*

O *Design Thinking* é um processo centrado no humano, por isso deve reconhecer os interesses dos *stakeholders* interagindo uns com os outros, de modo que as ideias sejam construídas por meio da capacidade intuitiva e diferentes formas de se expressar (Clune & Lockrey, 2014). Este modelo é baseado na sensibilidade do time do projeto e em métodos de *design*, fazendo com que os *stakeholders* participem desde as etapas iniciais. Dessa maneira, eles se tornam parte do processo do levantamento de necessidades, de modo que o resultado seja mais claro e objetivo, sem impor limites à criatividade e/ou à imaginação (Brown, 2008; Brown & Katz, 2011; Clune & Lockrey, 2014). Assim surge a primeira proposição do estudo:

P1. Os *stakeholders* das empresas participam desde a fase inicial do projeto.

Como parte fundamental do processo, o resultado da etapa inicial é um protótipo que auxilia a transformação das ideias em objetos ou evidências concretas de seus desejos. Este protótipo é o insumo principal e deve ser submetido à avaliação dos usuários finais para experimentações, fornecendo *feedback* e realização de ajustes conforme necessário (Bonini & Sbragia, 2011; Brown, 2008; Council, 2005).

Com base nessas argumentações, foram avaliadas as seguintes proposições de estudo.

P2. A interação da área de negócio com a área técnica, no grupo que participa do processo do *Design Thinking* ocorre com frequência, para coleta de *feedback*.

P3. O protótipo criado no processo de *Design Thinking* é submetido a *feedbacks* dos usuários finais, antes da integração com os métodos ágeis e após o desenvolvimento.

2.3.2. Métodos Ágeis

Durante a última década, a abordagem de desenvolvimento que utiliza os métodos ágeis tem recebido muita atenção de profissionais e também de pesquisadores acadêmicos, devido às suas características aderentes a um ambiente inovador e dinâmico. O objetivo principal da abordagem é satisfazer o cliente, fornecendo um produto valioso desde a primeira entrega e de maneira contínua (Fox, Sillito, & Maurer, 2008). Essa satisfação é realizada por meio de pequenas entregas, contendo um conjunto de funcionalidades que agreguem valor ao cliente o mais rápido possível, de maneira incremental e interativa. No entanto, essas entregas rápidas podem não gerar o valor necessário para satisfazer as necessidades dos usuários finais, devido ao retrabalho e redefinições estruturais no produto, que ocorrem devido à ausência de

arquitetura de projeto, pesquisa e, geralmente, devido à falta ou inexistência de documentação (Cao, Mohan, Xu, & Ramesh, 2009).

Métodos ágeis como o *Scrum* e o XP defendem um processo de desenvolvimento interativo, com a comunicação intensa entre o time técnico e os clientes ou usuários finais, de modo que o produto lançado seja o mais aderente possível às necessidades dos usuários finais. Estes métodos também argumentam que o processo de codificação para melhorar a qualidade e também a arquitetura, geram enormes desperdícios (Beck, 2000; Kautz, Johanson, & Uldahl, 2014).

Pode-se concluir que as principais características do método ágil são: ser incremental, ser interativo, ser simples e ser adaptável (Schwaber & Sutherland, 2013). Aplicar todas essas características no desenvolvimento de um projeto pode ser difícil para as organizações. Problemas de desencontros dos membros do time e cliente, sobrecarga de atividades e priorização das tarefas relacionadas à execução do projeto em relação às tarefas do processo de gestão, estão entre os principais itens ofensores da característica de interação e incremental (Carvalho & Mello, 2012). A definição dos métodos ágeis de serem adaptáveis, não está ligada a estabilização do processo e sim à possibilidade de fazer ajustes no modo de operação e também, complementados com outros frameworks (Pressman, 2005). Com base nesta afirmação, foi definida a seguinte proposição:

P4: A aplicação do método ágil no desenvolvimento de *software* possui como característica principal o fato de ser adaptável.

O *Product Backlog* é uma lista de iniciativas que deverá ser desenvolvida dentro das *Sprints*. O produto resultante ao final de cada *Sprint* deve possuir as características

necessárias para que possa potencialmente ser entregue ao cliente (Cohn, 2005). Com base nesta argumentação, foi definida a proposição:

P5: O incremento de produto entregue ao final da interação ou *Sprint*, é um incremento viável para implantação e utilização dos usuários.

2.3.3. Integração do *Design Thinking* com os métodos Ágeis.

A integração entre os dois modelos tem como objetivo validar a natureza do *design* no contexto dos negócios (Martin, 2009). Já Brown (2009) define que o conceito de *Design Thinking* é “trazer os princípios dos *designers*, abordagens, métodos e ferramentas para resolução de problemas”. A proposta deste modelo é empregar práticas que ajudem os *stakeholders* a pensar em pressupostos fundamentais na maneira como são desenvolvidas as soluções para os problemas (Boland & Collopy, 2004).

Quando comparado com o *Design Thinking*, os métodos ágeis mostram muita simetria como a característica de aprendizagem, o processo incremental e interativo e a contínua comunicação com a equipe (Lindberg, Meinel, & Wagner, 2011).

Existem razões para que os métodos ágeis e o *Design Thinking* sejam utilizados em conjunto. Os desenvolvedores de *software*, que utilizam os métodos ágeis, são voltados às características técnicas, enquanto que a característica do *Design Thinking* é a relativa empatia para identificar as necessidades do usuário (Kowark, Häger, Gehrler, & Krüger, 2013).

P6. A integração do *Design Thinking* e dos Métodos ágeis ocorre de modo sequencial.

Conforme descrito na Figura 6, o processo é iniciado a partir de uma ou mais iniciativas. Após a priorização das iniciativas, elas passam pela abordagem de *Design Thinking* para refinamento das ideias, criação de soluções que atendam as expectativas dos usuários e também como viés técnico viável. As soluções são enviadas para o processo de desenvolvimento por meio de histórias de usuários e protótipos, ou ainda podem ser criadas novas iniciativas que deverão iniciar o processo novamente. Na sequência, após nova rotina de priorização, entram em outro ciclo de desenvolvimento, desta vez utilizando os métodos ágeis para transformar a solução criada pelo processo anterior (*Design Thinking*) em produto ou solução funcional. Durante o processo de desenvolvimento existem novas coletas de *feedback*, e estes novos itens podem se tornar novas iniciativas.

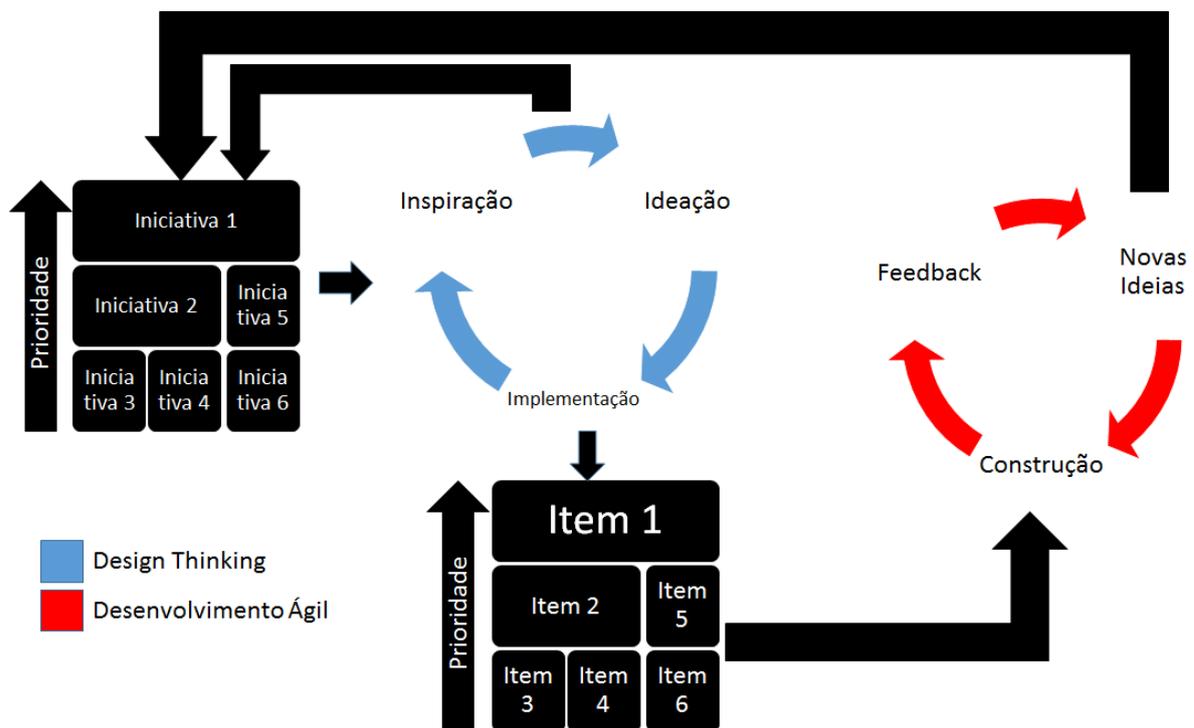


Figura 6 - Integração *Design Thinking* e Desenvolvimento Ágil
Fonte: Elaborado pelo Autor

3. METODOLOGIA E TÉCNICAS DE PESQUISA

Neste capítulo serão descritos os procedimentos metodológicos utilizados para o desenvolvimento da pesquisa, abordando como o *Design Thinking* pode ser integrado com os métodos ágeis.

3.1. DELINEAMENTO DA PESQUISA

O principal objetivo deste estudo foi identificar como ocorre a integração entre o *Design Thinking* e os métodos ágeis.

O desenvolvimento deste estudo foi feito em etapas conforme Figura 7:

- Levantamento bibliográfico dos temas *Design Thinking* e Métodos Ágeis;
- Desenvolvimento de proposições da pesquisa que serão foco de análise no estudo dos casos;
- Escolha do método de pesquisa;
- Escolha das características da pesquisa e definição das características da unidade de análise;
- Desenvolvimento dos instrumentos de coleta de dados;
- Definição da análise e interpretação dos dados;
- Apresentação dos resultados.

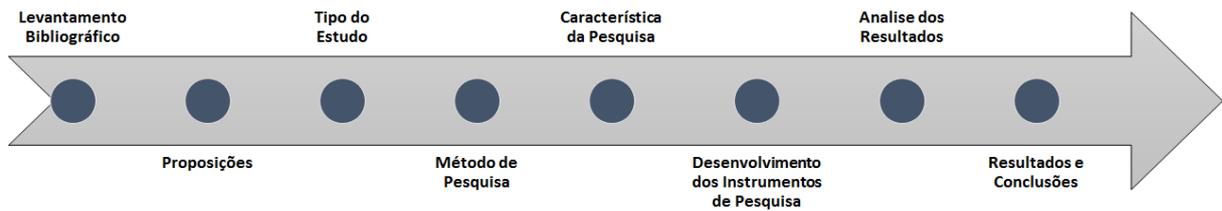


Figura 7 - Etapas da Pesquisa
Fonte: Elaborado pelo Autor

A pesquisa teve uma visão prática, conforme Figura 8, dirigida à solução de um problema do cotidiano, dentro da área de desenvolvimento de *software*, podendo-se estabelecer uma nova maneira de trabalho, caso os resultados deste trabalho sejam comprovados. Por esse motivo foi utilizada a metodologia de pesquisa com natureza definida e aplicada. (Prodanov & Freitas, 2013).

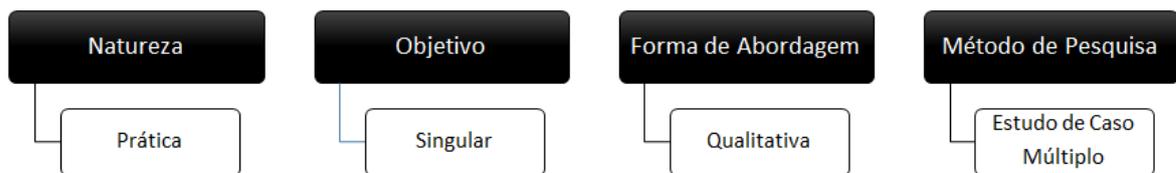


Figura 8 - Resumo da Qualificação da Pesquisa
Fonte: Elaborada pelo Autor

Considerando as particularidades no processo de desenvolvimento de *software*, optou-se por focar numa tipologia de pesquisa que contemplasse a pesquisa exploratória quando os procedimentos que fossem abordados no estudo de caso.

Escolheu-se a pesquisa exploratória, pois normalmente é utilizada quando há pouco conhecimento sobre a temática abordada. O objetivo foi conhecer com maior profundidade para deixá-la mais clara. Isso ocorreu devido ao aprofundamento dos conceitos preliminares sobre a temática que não era satisfatória e com isso contribuir para o esclarecimento de questões e proposições abordadas sobre o assunto (Eisenhardt, 1989; Yin, 2015).

A escolha correta do método de pesquisa foi um dos pontos mais importantes do estudo. A questão de pesquisa definida no trabalho apontou para uma estratégia de estudo de

caso de caráter exploratório, utilizando a forma de abordagem qualitativa e de observação direta das empresas em análise, como as técnicas mais adequadas, uma vez que se pretende verificar como são aplicados os métodos e processos abordados pelo *Design Thinking* e os métodos ágeis.

O método de estudo de caso é um modo de pesquisa que foca no entendimento da dinâmica presente em determinada situação, permitindo a construção de uma teoria, provendo descrições ou ainda testando uma teoria a partir de métodos de coleta de dados como entrevistas e coleta de documentos históricos (Eisenhardt, 1989).

Escolheu-se a abordagem do estudo de caso múltiplo pois permite a composição de um estudo mais robusto, uma vez que foi possível coletar dados com maior riqueza de informações e análises, em função da solidez dos resultados (Yin, 2015). Este método também é mais adequado para estudos no contexto prático das empresas, a partir de eventos reais, com o objetivo de explicar e descrever os fenômenos dentro do próprio contexto. Este método de pesquisa também foi escolhido com o objetivo de alcançar o contexto mais profundo e detalhado de uma aplicação prática (Eisenhardt, 1989; Yin, 2015).

A opção pelo método de coleta de dados qualitativo ocorreu após a escolha do método de pesquisa. O método qualitativo tem como objetivo principal o aprofundamento e compreensão do fenômeno, por meio da qualificação dos dados coletados durante a execução e coleta das informações, e não se preocupa com a representatividade numérica dos dados (Creswell, 2014; Yin, 2015).

O papel das proposições de estudo ou desenvolvimento da teoria, anterior a qualquer condução de coleta de dados, é o que caracteriza o estudo de caso, diferenciando-se de outros métodos. Desta maneira foram criadas proposições para cada constructo, baseando-se na revisão da literatura e refletindo o aspecto teórico mais importante, buscando evidências relevantes para definir e confirmar os benefícios específicos para a organização (Yin, 2008,

2015). Na Figura 9 é descrito como estão relacionadas as proposições com cada um dos temas que serão abordados no estudo.

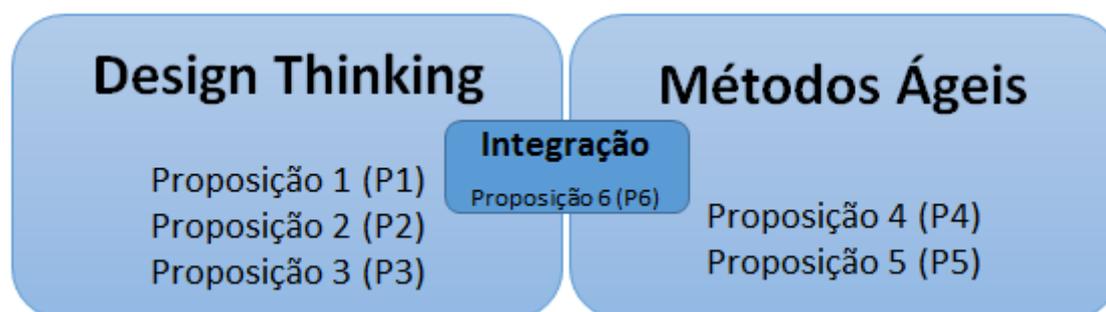


Figura 9 - Relação das Proposições de Estudos
Fonte: Elaborados pelo Autor

A Figura 10 demonstra como estão ligados os objetivos com as proposições, seguidas das questões, disponíveis no apêndice, que serão utilizadas na entrevista semiestruturada proposta.

Tema	Objetivo Secundário (OS)	Proposições (P)	Questões (Q)
<i>Design Thinking</i>	OS1	P1	Q18; Q19; Q20, Q21
<i>Design Thinking</i>	OS1	P2	Q18; Q19; Q20, Q21
<i>Design Thinking</i>	OS1	P3	Q21
Métodos Ágeis	OS2	P4	Q8, Q9, Q10, Q11, Q12, Q13, Q14
Métodos Ágeis	OS2	P5	Q10, Q25, Q26, Q31
Integração do <i>Design Thinking</i> com Métodos Ágeis	OS3	P6	Q25, Q26, Q27, Q30

Figura 10 - Relação Entre Proposições, Objetivos e Questões
Fonte: Elaborado pelo Autor

Para validação das proposições será utilizado o método qualitativo, conforme demonstrado na Figura 11, e serão utilizadas três possibilidades de validação: (i) Proposição Validada, ocorrerá quando pelo menos 75% das empresas estudadas estiver de acordo com a proposição, ou seja, quando significar a maioria absoluta das empresas. (ii) Proposição Validada Parcialmente, quando 25% a 75% das empresas estudadas estiverem alinhadas com

a proposição, quando houver empate. Por fim, (iii) Não Validada, esta ocorrerá quando menos que 25% das empresas analisadas estiver de acordo com a proposição, quando apenas a minoria ou nenhuma das empresas puder validar a proposição.

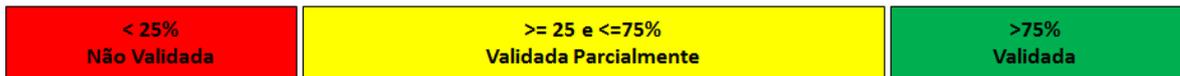


Figura 11 - Regra de Validação das Proposições

3.2. MÉTODO DE PESQUISA E UNIDADE DE ANÁLISE

O estudo de caso pode ser de dois tipos, único ou múltiplo. Cada um destes tipos pode ser derivado para holístico ou incorporado e, desta maneira, resultam em quatro tipos diferentes de estudos de caso, conforme é demonstrado na Figura 12 (Yin, 2015).

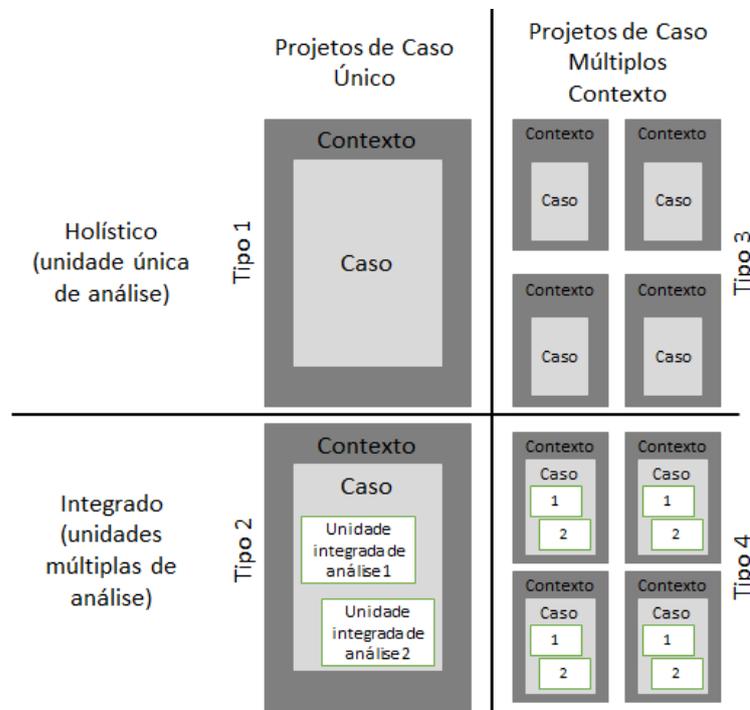


Figura 12 - Tipos Básicos de Projetos para Estudo de Caso
Fonte: Yin, 2015 p.40

Neste tipo de desenho, o estudo de casos múltiplos contribui para um estudo mais convincente, uma vez que é possível contestar e contrastar respostas obtidas de forma parcial, em cada um dos casos analisados (Yin, 2015).

A escolha dos casos baseou-se na disponibilidade das empresas e na sua receptividade, devido à dificuldade de acesso às informações, coleta de documentos e acesso aos colaboradores. A pesquisa foi realizada em quatro empresas de tamanhos diferentes, sendo que essas empresas têm destaque em seus segmentos por serem inovadoras, trazendo uma linha de serviços e/ou produtos voltados à linha de soluções digitais.

O fator determinante para a escolha das empresas foi o modelo estruturado que elas utilizam para desenvolvimento de *software* e de tamanhos diferentes para compreender se existe diferença na aplicação dos mesmos. Esse modelo contém as práticas do *Design Thinking* para a criação de soluções de T.I. e também utiliza os métodos ágeis para o desenvolvimento dos *softwares*.

3.3. COLETA DE DADOS

A coleta de dados foi realizada por meio de entrevistas, observações e arquivos contendo dados históricos ou informações relevantes ao tema. Estes modos de coleta são os indicados para um estudo em profundidade (Eisenhardt, 1989; Yin, 2015). Deste modo, será possível explorar como as empresas utilizam os temas abordados nesta pesquisa.

A escolha da coleta de dados por meio de entrevistas deve-se a característica de conseguir extrair informações objetivas com detalhes aprofundados sobre a temática e, além disso, extrair também, informações subjetivas como opiniões pessoais dos entrevistados, atitudes, como ocorre o relacionamento entre os valores (Martins, 2004). Dentre as opções de modelo de entrevistas encontradas na literatura, optou-se pela entrevista semiestruturada, que

combina perguntas abertas e fechadas, nas quais existe a possibilidade de discorrer sobre o tema proposto, mesmo que informalmente (Boni & Quaresma, 2005). Utilizando questões previamente estabelecidas, disponíveis no APÊNDICE I, o pesquisador, fazendo uso de uma conversa informal, conduziu a entrevista e, quando considerou oportuno, criou perguntas com o objetivo de ter mais profundidade sobre o tema. Este tipo de entrevista é muito utilizado quando se deseja delimitar o volume de informações, obtendo assim um direcionamento maior sobre o tema (Boni & Quaresma, 2005).

Para que informações não sejam perdidas e/ou algum detalhe tenha passado despercebido e também a forma como foi dito, as entrevistas foram gravadas com auxílio de um gravador de voz, com autorização prévia dos entrevistados.

As entrevistas podem se realizar de quatro formas: i) entrevistas pessoais; ii) entrevistas por telefone; iii) entrevistas em grupo e por fim, iv) entrevistas com profundidade (Runeson & Höst, 2009), sendo que apenas a entrevista em grupo não foi utilizada. Todas as entrevistas foram feitas com profundidade, de modo que fossem explorados determinados assuntos com mais detalhes, buscando entender as percepções, experiências e com informações mais detalhadas. Além das entrevistas pessoais, que foram realizadas *face a face*, foram utilizadas entrevistas à distância, com o auxílio do aplicativo *Skype*, para viabilizar as videoconferências. Este tipo de entrevista é utilizado quando o entrevistado possui dificuldade de agenda e também pelo fato do entrevistado se sentir menos pressionado pelo entrevistador (Bryman & Bell, 2015). Foi necessária a utilização de videoconferência para minimizar a dificuldade de observação da linguagem corporal, uma vez que é demonstrada em algumas ocasiões desconfortos e/ou confusões quando abordados alguns temas, além da apresentação de alguns desenhos explicativos para complementar o estudo (Bryman & Bell, 2015).

As entrevistas foram feitas com funcionários e prestadores de serviços ligados à área de gestão de projetos e de metodologias de desenvolvimento de produtos, por meio de perguntas diretas e indiretas. Na Figura 13 é apresentado o perfil dos entrevistados.

ID	Empresa	Prestador de Serviços	Cargo	Formação	Nível Hierárquico	Tipo Entrevista	Tempo (h)
1	Empresa Grande Porte	Sim	<i>Agile Coach</i>	MBA	Consultivo	Pessoal	00:45
2	Empresa Grande Porte	Sim	<i>Agile Coach</i>	MBA	Consultivo	Pessoal	00:40
3	Empresa Grande Porte	Não	Metodologia	Bacharel	Gerente	<i>Skype</i>	01:17
4	Empresa Médio Porte	Não	<i>Agile Coach</i>	MBA	Consultivo	Pessoal	01:03
5	Empresa Médio Porte	Não	Gerente Projetos	MBA	Gerente	<i>Skype</i>	01:10
6	<i>Software House</i>	Não	CEO	MBA	Diretor	<i>Skype</i>	01:25
7	<i>Startup</i>	Não	CEO	Pós-Doutorado	Diretor	<i>Skype</i>	00:52
8	<i>Startup</i>	Não	Gerente Produto	Mestrado	Gerente	<i>Skype</i>	01:06

Figura 13 - Perfil dos Entrevistados

Fonte: Elaborado pelo Autor

Os entrevistados 4 e 5 atuam numa empresa do ramo financeiro de médio porte, na área de tecnologia da informação no setor de desenvolvimento de *software*. O entrevistado 4, é consultor da empresa, atuando com gerente de projetos e *Agile Coach*, formado em engenharia de *software* com MBA em gestão de projetos, e diversas certificações ligadas a gestão de projetos utilizando metodologias ágeis. Sua atuação na empresa tem como objetivo melhorar a eficiência do desenvolvimento de *software* e para isso utiliza os métodos ágeis e *Design Thinking* como pilares do processo. O entrevistado 5, atua na empresa na área de governança corporativa de tecnologia da informação, é formado em administração de empresas, com MBA em gestão de projetos. Seu papel na empresa é direcionar o processo de desenvolvimento de *software*, seguindo as melhores práticas de acordo com as políticas da companhia e também, buscar melhorar o processo atual com novas técnicas e métodos disponíveis no mercado.

Os entrevistados 1, 2 e 3 atuam em uma empresa de grande porte do setor financeiro, sendo que os entrevistados 1 e 2 são consultores terceirizados e tem como principal objetivo, atuar no processo de transformação do centro de desenvolvimento de *software* anteriormente administrado por meio da gestão tradicional de projetos (*waterfall*) para a gestão de desenvolvimento de *software* com metodologias ágeis. Ambos possuem MBA em gestão de projetos. O entrevistado 2, é especialista em gestão de projetos de inovação (*innovation management*). O entrevistado 3 é bacharel em ciências da computação e é responsável pela criação e manutenção da metodologia de desenvolvimento de *software* da empresa.

A escolha do entrevistado 6 foi determinante pelo fato de ser CEO (*Chief Executive Officer*) e proprietário da empresa que presta serviço de desenvolvimento de *software* para empresas de diversos ramos de atuação. Este perfil proporciona uma visão ampla da utilização dos métodos ágeis e do *Design Thinking* em empresas de ramos de atividade variados. O entrevistado 6 é um profissional da área de tecnologia, com MBA em Identidade Empresarial, ligada ao empreendedorismo empresarial, possui mais de 50 certificações técnicas e é autor de mais de 20 livros técnicos. Atua como facilitador e *coach* em empresas de diversos ramos, como setor portuário, financeiro, varejo eletrônico em projetos de desenvolvimento de *software* e análise de negócios.

Neste estudo, também buscou-se uma empresa que fosse pequena e inovadora. Neste requisito encontra-se os entrevistados 7 e 8. O entrevistado 7 é formado, com mestrado e doutorado em física, tem perfil empreendedor e pesquisador na área de marketing e inovação tecnológica. Atua como diretor técnico na área de inovação. Possui pelo menos 2 duas empresas, sendo que, a que ora está sendo estudada, é caracterizada por ser uma empresa voltada ao desenvolvimento de *software* utilizando a IoT (*Internet of Things*) como um dos pilares. Já o entrevistado 8, é bacharel em ciência da computação e também possui mestrado na mesma área, é gerente de produto na empresa.

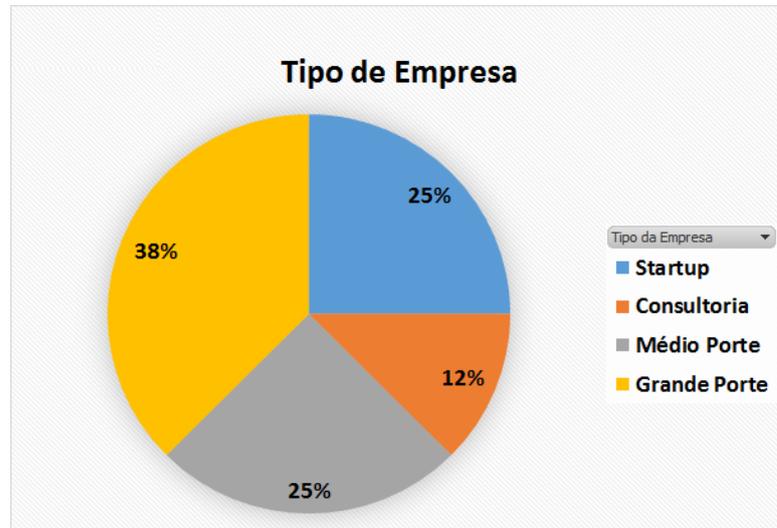


Figura 14 - Entrevistados por Tipo de Empresa
Fonte: Elaborado pelo autor

Desta maneira o perfil dos entrevistados escolhidos é uniforme com relação ao perfil das empresas em que atuam (Figura 14) e com funções ligadas a gestão dos projetos e/ou de metodologia da área de desenvolvimento de *software* das empresas estudadas (Figura 15).

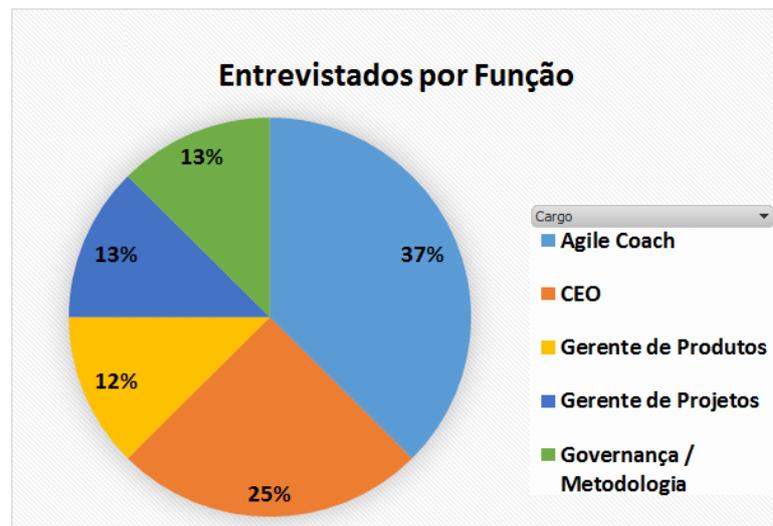


Figura 15 - Entrevistados por Função
Fonte: Elaborado pelo autor

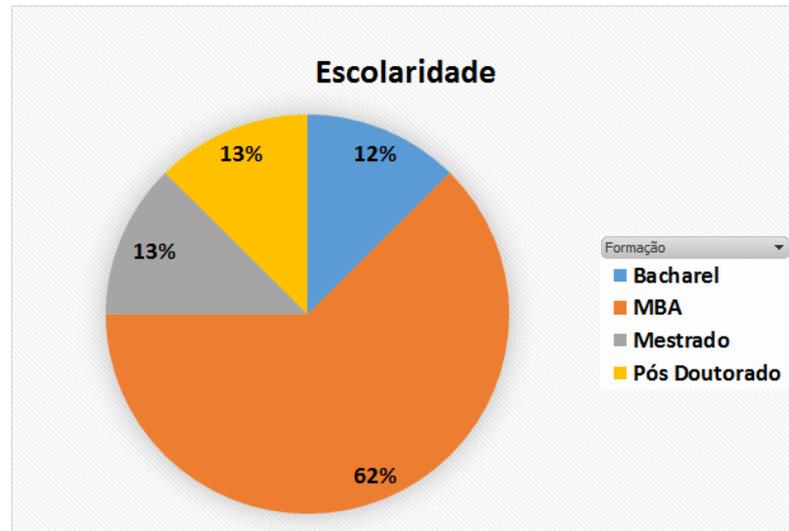


Figura 16 - Escolaridade dos Entrevistados
Fonte: Elaborado pelo autor

Todos os entrevistados possuem grande experiência na área de desenvolvimento de *software* e em tecnologia da informação (Figura 17) e com grande notoriedade acadêmica (Figura 16), uma vez que 78% dos entrevistados possuem nível acadêmico de no mínimo pós-graduado.



Figura 17 - Experiência dos Entrevistados em anos
Fonte: Elaborado pelo autor

3.4. ANÁLISE DOS DADOS

A análise dos dados é a etapa mais difícil da pesquisa, pois consiste em examinar e classificar as informações coletadas (Eisenhardt, 1989; Yin, 2015). Somente com a clara

descrição da análise de dados será possível extrair o melhor entendimento dos resultados por meio de processo sistemático e rigoroso (Dubé & Paré, 2003). Para viabilizar a análise do conteúdo das entrevistas, as respostas foram tabuladas utilizando o programa Microsoft Excel de modo que as informações pudessem ser confrontadas e avaliadas. Também neste passo, foram criadas categorias e subcategorias de acordo com o levantamento bibliográfico abordado no referencial teórico.

As variáveis que compõem um estudo de caso e associada a análise dos dados são: descrição de procedimentos, anotações de campo, esquema de codificação, flexibilidade, validade do constructo, evidências, comparação dos casos, técnicas de análise, validade interna, citações, revisão do relatório e comparação dos resultados com a literatura (Oliveira, Maçada, & Goldoni, 2009).

O procedimento para obtenção da validade de um constructo é feito por meio do encadeamento lógico das evidências (Yin, 2015). Este encadeamento consiste em um relatório com citações de aspectos relevantes, de modo organizado, para evidenciar os fatos (Yin, 2008, 2015). O uso destas evidências é uma forma de trazer os participantes da pesquisa para dentro do estudo de caso, auxiliando o leitor a elaborar, de forma independente, a adequação da análise (Oliveira et al., 2009; Yin, 2015).

A definição da estratégia de análise de dados no estudo de caso contribui para a decisão sobre o que analisar (Yin, 2015), utilizando as informações coletadas com as proposições teóricas, de modo que sejam comparados com o resultado e possa ser uma explanação válida ou não (Yin, 2015).

Para a apresentação dos resultados de um estudo de caso múltiplo, foram usados dois estágios de análise: a análise individual dos casos e a análise cruzada dos casos. A análise individual foi realizada para identificar as possibilidades de agrupá-las em categorias, e a análise cruzada foi a comparação entre elas, com o objetivo de identificar a similaridade e

diferenças entre os padrões, a fim de obter a validade externa dos dados. Os resultados emergentes das evidências empíricas, provenientes da análise cruzada, foram confrontados com os elementos encontrados na literatura.

Dentre as formas escritas sugeridas por Yin (2015) para análise dos dados, foi utilizada a modalidade que se aplica aos estudos de casos múltiplos. A análise foi apresentada por meio da análise cruzada de casos, apresentando informações resumidas sobre os casos individuais.

As entrevistas foram gravadas em áudio e posteriormente transcritas na íntegra para modo texto, utilizando programa Microsoft Word.

3.5. LIMITAÇÕES DA PESQUISA

Os temas abordados neste estudo são relativamente novos e pouco maduros nas empresas estudadas. E, também, como estes métodos *Design Thinking* e métodos ágeis estão diretamente ligados às áreas de produto e/ou inovação das empresas, existe grande preocupação com o sigilo das informações que são passadas por meio das entrevistas. As empresas que foram estudadas tiveram o número restrito de pessoas entrevistadas, o que pode prejudicar a qualidade das informações, devido a tendências de favorecimento da área ou do profissional, ou ainda limitação na interpretação dos fatos dados pelo profissional entrevistado.

Os resultados não puderam ser generalizados devido ao tipo de projetos nos quais os profissionais estão envolvidos, seu conhecimento técnico referente aos métodos e também o segmento de atuação da empresa.

4. ANÁLISE DOS DADOS

Este capítulo apresenta e caracteriza as empresas, descrevendo as práticas e métodos utilizados com relação aos métodos ágeis e o *Design Thinking*, e também como ocorre a integração entre eles.

Os entrevistados solicitaram que não fossem citados no estudo o nome das empresas e seus produtos, e que seus nomes fossem mantidos em sigilo, devido às regras corporativas referentes aos seus processos de inovação e metodologia de desenvolvimento.

Após a caracterização de cada uma das empresas, é apresentada a análise cruzada dos dados com as principais evidências confrontadas entre as empresas, e o levantamento bibliográfico referente aos temas estudados.

4.1. EMPRESA GRANDE PORTE

A Empresa Grande Porte é uma das maiores empresas privadas que atuam no mercado financeiro do Brasil, estando entre as 10 maiores do seu segmento, ultrapassando o valor de U\$ 100 bilhões em ativos (Wikipédia, 2017), oferecendo a seus clientes produtos relacionados a serviços de seguros, cartões de crédito, conta corrente e investimentos. Atualmente, o seu relacionamento com os clientes ocorre por meio de agências físicas espalhadas em todo o território nacional e também disponibiliza o meio eletrônico por intermédio de canais de atendimento como caixas eletrônicos, *websites*, aplicativos e *call centers*.

Seguindo a tendência de mercado, motivada pelas *Startups* e pelo novo conceito, chamado de *fintec* (do inglês: *finance* e *technology*, que traz soluções inovadoras para resolução de problemas ligados ao mercado financeiro), a empresa está fazendo grandes investimentos, superando cifras de U\$ 250 milhões na área de tecnologia e canais digitais,

com o objetivo de aproximar a instituição de seus clientes por meio da internet, utilizando websites e também aplicativos para celular (Gradilone, 2016).

Para sustentar essa nova abordagem, conhecida como *Digital*, a empresa deu início em meados de 2014 a primeira iniciativa utilizando um novo modelo de desenvolvimento de *software*, estruturado sobre *frameworks* e práticas ágeis, *Design Thinking* entre outros, de modo que o processo de desenvolvimento seja mais rápido e eficiente, proporcionando uma nova experiência com o usuário final, o cliente da empresa. No final de 2016, quase a totalidade dos projetos que envolvem canais digitais são desenvolvidos nessa nova área, o que envolve aproximadamente 500 colaboradores diretos, 4 linhas de produtos e orçamento mensal aproximado de R\$ 10 milhões.

O processo de desenvolvimento nessa área é estruturado por meio do *framework* SAFe (*Scaled Agile Framework*) e do DAD (*Disciplined Agile Delivery*) da IBM, conforme é demonstrado na Figura 18.

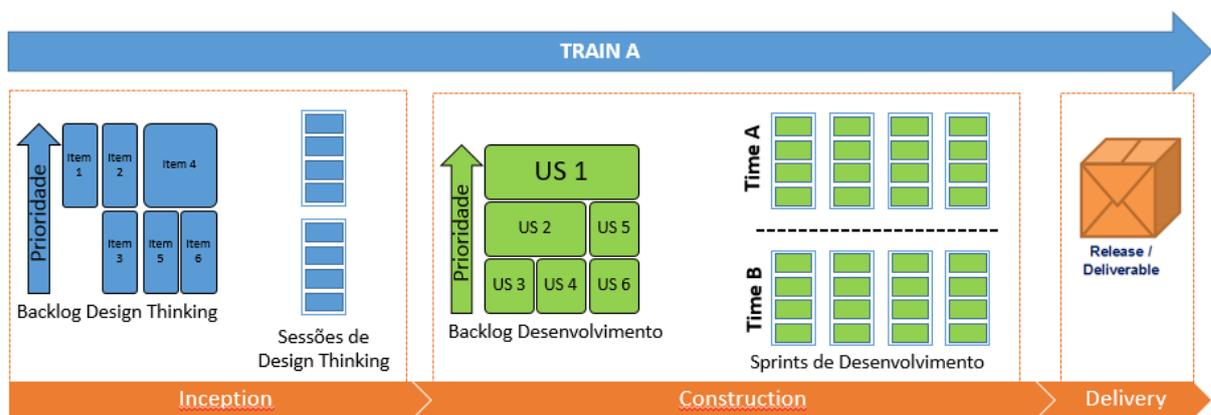


Figura 18 - Processo de Desenvolvimento - Empresa Grande Porte
Fonte: Adaptado dos documentos internos da empresa

Os projetos são divididos em *Trains*, que são as linhas de produtos da empresa. Na fase inicial do processo, chamado de *Inception* no *framework* da IBM, o DaD, todas as iniciativas são catalogadas no *Product Backlog* de *Design*. Nessa etapa é utilizado o *Design Thinking*, quando buscam a solução para os problemas apresentados e também são definidos

os requisitos funcionais e não funcionais do produto, sendo uma melhoria ou novo produto. Uma vez definidos esses requisitos, são elaboradas as *User Stories* e enviados para desenvolvimento.

Segundo o entrevistado 3, nesta etapa do processo, todos os envolvidos, incluindo os profissionais da área de negócio, que são os demandantes, e também os profissionais da área técnica, devem comparecer às sessões de *Design Thinking* para discutir e viabilizar uma solução para as iniciativas priorizadas. Porém, foi verificado com os outros entrevistados que existe uma variação dessa definição, pois nas iniciativas que envolvem alguma nova funcionalidade ou novo produto, inédito para o mercado, os profissionais de nível executivo e colaboradores com perfil criativo participam das sessões que tem como objetivo discutir a ideia, de modo que ela entre em outra sessão de *Design Thinking*, com outro perfil de profissional, e que a ideia inicial, entre com determinado grau de amadurecimento para discutir apenas a viabilidade técnica e como a solução deverá ser implementada. Já em iniciativas de melhoria de um produto já existente ou implementação de requisitos legais (requisitos obrigatórios por lei), não existe essa sessão com os profissionais de alto escalão, apenas a sessão que trata de como implementar a solução, com um perfil de profissionais que tenha como foco viabilizar a ideia proposta ou o requisito.

[Entrevistado 1]: Existem os dois casos. Tem ideias que são colocadas à prova, os executivos, alta gerência das empresas, muitas vezes tem uma ideia, tem um insight, e eles querem validar se aquilo é viável ... para validar e tangibilizar aquela ideia. Então tem situações onde a ideia já vem conhecida para ser validada e tem situações onde novas ideias surgem. Somente quando há um problema a ser resolvido, onde a solução não é conhecida. Então existem os dois casos, dependendo do contexto, uma abordagem ou outra.

[Entrevistado 2]: Os gestores das áreas, o corpo técnico, participa, a gente traz as pessoas do operacional para participar também.

[Entrevistado 3]: Todos os envolvidos devem participar (das sessões de Design Thinking).

No processo de *inception* são utilizadas diversas ferramentas, apesar de não existir normalização para isso. Existe um toolkit de *softwares* e acessórios que pode facilitar as sessões, como o uso de aplicativos de prototipação (*wireframe*), post-its, desenhos manuais nas paredes que são preparadas para isso ou então em quadros apropriados.

[Entrevistado 1] Formalmente não. O que existe são opções que chamam até de TOOL KIT, que existem aí, dependente do contexto, várias técnicas e vários modos de trabalhar....

[Entrevistado 2 Não existe uma definição para isso, o que quiserem utilizar é válido desde que atinjam o objetivo.

Os protótipos que são gerados nesta fase são enviados para aprovação dos clientes finais da funcionalidade, porém, nesse processo existem algumas alternativas que podem mudar de acordo com a confidencialidade do projeto, ou ainda, da necessidade de aprovações extras como segurança, capacidade operacional ou ainda homologação legal. Nessa etapa também são criados indicadores para medir a satisfação desses usuários. Esses indicadores podem mudar de acordo com a origem da iniciativa, como por exemplo, indicadores de performance de negócios (aumento de vendas, captação de novos clientes), performance técnica (mais leve, mais rápido, mais seguro) ou receptividade de mercado (aumento do valor da marca ou das ações da empresa, número de downloads por dia, opinião de mercado). Os

critérios de avaliação fazem parte da definição da solução. Para avaliar esses indicadores, o processo também pode ser bastante vasto, como por exemplo contratação de empresas para coleta de informação de mercado, indicadores implementados dentro do produto ou ainda por meio de relatórios gerenciais.

[Entrevistado 1] Tem muitas respostas, mas depende. Depende do nível de confidencialidade do projeto. Cada negócio, cada área ou produto tem o seu objetivo. Esse objetivo, a gente procura estabelecer alguma medição para ele.

[Entrevistado 2] A gente sempre faz o feedback constante. A gente leva isso para as frentes que vão de fato operacionalizar, ou às vezes a gente acaba fazendo um processo de Family friend onde a gente consegue feedback dos usuários finais de fato do produto. Não existe uma regra clara. Ora o pessoal mede por pesquisa de satisfação ou simplesmente não mede.

[Entrevistado 3] Passa por feedbacks mais restritos. Normalmente dentro da própria área. Em projetos pioneiros ou que envolvem sigilo, são contratadas empresas especializadas para fazer esta coleta de informação de modo sigiloso. O modelo do banco prevê que além da pesquisa de satisfação, sejam incluídas métricas vinculadas ao objetivo do projeto, porém tenho ciência que isso não é feito.

O resultado da fase inicial, da fase de *Inception*, é uma lista de *User Stories*, chamado de *Product Backlog* que é a lista de desejos a serem desenvolvidos e implementados em cada uma das linhas de produtos da empresa, na fase chamada de *Construction*.

Na etapa *Construction*, realiza-se a arquitetura e o desenvolvimento do produto. Esta fase é estruturada com *frameworks* ágeis para gestão do desenvolvimento de *software* com o *Scrum* e diversas outras técnicas e práticas ágeis.

Assim é feita a arquitetura técnica da solução de modo que garanta o menor desperdício com retrabalho e também uma arquitetura que sustente o produto como um todo e não um código desestruturado e suscetível a falhas. Esta etapa é fundamental e de grande importância para o sucesso do produto, uma vez que os times que estão desenvolvendo as funcionalidades não têm visão clara do todo, porque neste modelo de trabalho, as funcionalidades são produzidas de maneira incremental.

[Entrevistado 1] Como não existe a visão a longo prazo do projeto, pode sim ocorrer muito retrabalho, mas este retrabalho é previsto.

[Entrevistado 2] Existe retrabalho, porém ele é calculado.

[Entrevistado 3] Existe um comitê que faz e garante que este retrabalho seja o menor possível. Técnicas de Inception e controle de qualidade de código minimizam isso. Temos ferramentas que garantem que isso não é um problema.

Foi verificado que não existe sincronia entre o que é produzido na fase de *Inception* e do que realmente entra na linha de produção. Normalmente existe um estoque de *User Stories*, apesar de não ser o modelo ideal proposto pela empresa e pelos colaboradores. Segundo os entrevistados, este processo é o mais complexo, pois existem restrições técnicas, fator de influência entre os *Sponsors* e prazos legais.

[Entrevistado 2] Depende da situação. Num projeto de um único Sponsor, é realizado dentro... Ocorre de maneira casada, são janelas. Dentro da janela podem ocorrer algumas reuniões, porém durante a janela são feitas todas atividades pertinentes às tarefas de design.

[Entrevistado 3] Teoricamente eles devem ser sequenciais, mas é comum um processo de Design Thinking gerar demandas para mais que uma Sprint ou ainda envolver

mais que um time. Então podemos dizer que trabalha-se de modo não integrado, mas sim um alimentando a esteira do outro, e sempre existe um estoque para desenvolvimento.

[Entrevistado 3] É uma das maiores dificuldades porque a área de negócio... Uma Sprint que já está em andamento não para... Para incorporar novos itens tem que ser concluída até ter uma nova janela de priorização.

A Figura 19 é uma reprodução do documento que os gestores da área utilizam para administrar as demandas que entram para desenvolvimento. Nele pode-se perceber que existe data de entrada da demanda, prioridade, o *release* de entrega e não tem vínculo algum com fase de *design* (*Inception*).

ID US	Data de Entrada	Prioridade	Descrição	Area	Produto	Tipo	Release	Team	S.O.

Figura 19 - Tabela de Informações da *Users Stories*
Fonte: Elaborado pelo Autor

O *Product Backlog* de desenvolvimento é ordenado por prioridade e colocados em diversas linhas de produção ou de desenvolvimento de *software*. Estas linhas de desenvolvimento são subdivididas em diversos times de desenvolvimento e estes, por sua vez, em *Sprints*, onde cada time tem características técnicas próprias e ao final do processo, juntam-se as entregas de todos estes times e *Sprints* e consolida num único *release* de entrega. Este sim é um incremento potencialmente implantável.

O processo de priorização é feito utilizando técnicas de mercado como WSJF (*Weighted Shortest Job First*), onde é calculado o custo do atraso (*Cost of Delay*) e tamanho do trabalho (*Job Size*), ou ainda a técnica de distribuição de pontos. Estas técnicas auxiliam o P.O. (*Product Owner*) a garantir o melhor resultado entre os valores que serão investidos e o

resultado esperado para o negócio. Apesar das técnicas, seja a mais complexa ou a mais simples, estão sujeitas a intervenção externa, como autoridade da alta gestão da empresa ou ainda acordos informais entre os times ou departamentos envolvidos com as demandas.

[Entrevistado 1] O Product Owner, ou pode-se chamar outras pessoas quando envolve decisões de outras áreas e tipicamente envolve alguns representantes do time. O time de gestão define a prioridade em comum acordo com os envolvidos.

[Entrevistado 2] Tem algumas que a gente trabalha por mapa de dependência, outras a gente trabalha por análise de plano, que são os benefícios atraídos pelo o cliente, algumas a gente trabalha com árvore mesmo, de produto.

[Entrevistado 3] Existe algumas técnicas que são propostas como Cost of delay e distribuição de pontos, mas normalmente é o fator político que define.

Durante o processo de *Construction*, são realizadas diversas técnicas de coleta de *feedbacks* das áreas que originaram as demandas para avaliação da assertividade, do que está sendo desenvolvido e também para verificar se o que foi idealizado durante as fases anteriores corresponde à necessidade ou expectativa dos demandantes, com relação ao funcionamento ou características visuais dos requisitos. Já neste processo é possível observar que o processo de *Design Thinking* traz ganhos quando utilizado para criar novas funcionalidades ou produtos, uma vez que o solicitante acompanha o processo de ideação e produção dos requisitos solicitados. Porém, o processo gera poucos ganhos como quando trata-se de melhoria ou implantação de alguma norma regulatória.

[Entrevistado 1] Quando uma solução já é conhecida, realmente o Design Thinking agrega pouco. Então você não dependeria dele para iniciar os trabalhos, o ágil seria

suficiente. Mas quando você não tem isso conhecido, você não sabe como resolver o problema, você tem uma complexidade maior e o Design Thinking agrega valor. Nas reuniões de Review do Ágil, as ideias vão surgindo e você vai aplicando técnicas diferentes, algumas vão tendo mais aceitação que outras.

[Entrevistado 3] Sempre tem chance de frustrar as expectativas. Isso está mais ligado ao timing do projeto. Quanto mais cedo iniciar o desenvolvimento e/ou quanto mais cedo envolver a área de negócios, ajuda nesta questão de expectativa.

Eventuais mudanças que ocorrerem durante o processo de produção, seja uma mudança de prioridade, cancelamento de *User Stories* ou redefinição de alguma funcionalidade, o *Product Owner* deverá dar as diretrizes, seja mudar o que está em andamento ou inserir esta mudança como uma nova *User Stories* no *Product Backlog* para ser priorizada, e assim seguir o fluxo normal de desenvolvimento.

[Entrevistado 2] É acionado o P.O. para ele negociar e definir o que será trocado, substituído e até mesmo sinalizar a inviabilidade da implementação daquilo.

[Entrevistado 3] Estando dentro do objetivo do projeto e com a aprovação do P.O., entra para o backlog, do contrário é descartada.

Na Figura 20, pode-se observar que existe legenda para diferenciar tarefas oriundas de mudanças, assim como *bugs*, que são tarefas originadas na etapa de controle de qualidade; tarefas não planejadas, que são as tarefas que os membros do time têm que executar e não foram planejadas no início da *Sprint*, além das tarefas normais criadas e estimadas em tempo de planejamento da *Sprint*.

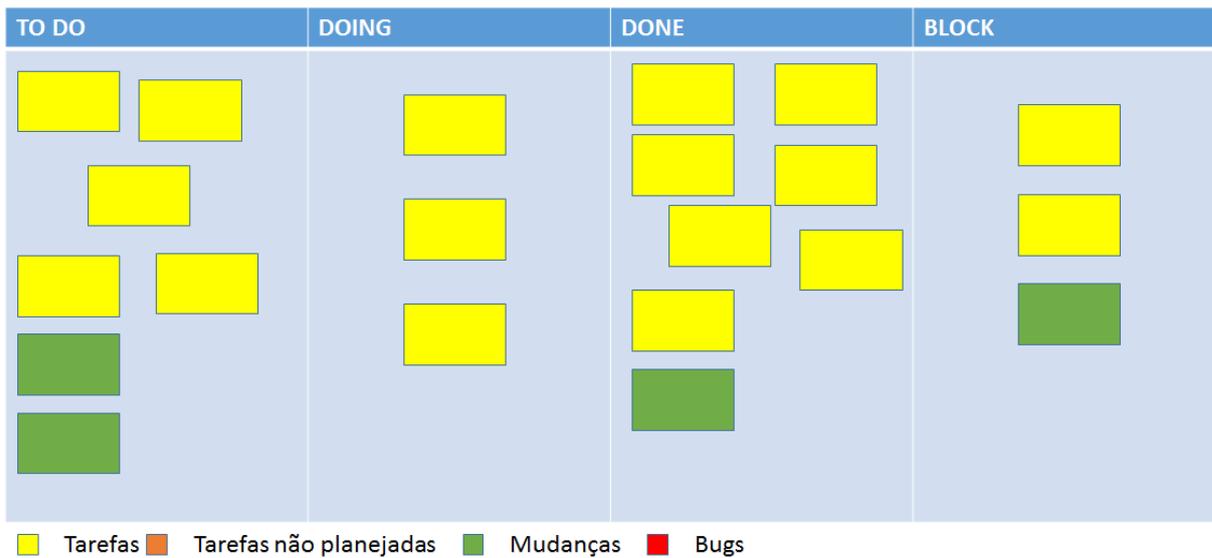


Figura 20 - Kanban de Desenvolvimento

Fonte: Elaborado pelo Autor

Ao final da etapa *Construction*, ao final das execuções das *Sprints*, é realizado um processo de fusão dos códigos produzidos por cada um dos times e/ou *Sprints*. É verificada a qualidade e padrões de código e, por fim, o *release* é implantado e disponibilizado para uso.

Também ao final do processo, é realizado outro ciclo de coleta de *feedback* com as áreas envolvidas e os clientes finais das funcionalidades ou produtos, aplicando os parâmetros definidos na fase de *inception*.

4.2. EMPRESA MÉDIO PORTE

A Empresa Médio Porte atua no mercado financeiro de capital privado. Foi considerada uma empresa de médio porte neste estudo, por não constar no ranking das dez maiores entidades em seu segmento, mas está entre as 50 maiores empresas do seu segmento (Wikipédia, 2017), contando com patrimônio ativo total superior a U\$ 2 bilhões e oferecendo a seus clientes serviços relacionados a conta corrente, cartão de crédito, financiamentos e

investimentos por meio exclusivamente de aplicativos ou plataformas de acesso por meio da internet.

Esta empresa é relativamente nova no mercado, uma vez que ainda não se passou 10 anos desde a sua fundação. A empresa busca manter relacionamento com os seus clientes por meio de tecnologia de ponta e dinâmica que permite a geração de negócios sete dias por semana e 24 horas por dia.

Tendo como característica principal o relacionamento por meio de plataformas digitais e acesso à internet, a empresa faz grandes investimentos na área de tecnologia para manter a proposta da companhia o seu diferencial no mercado.

Por ser uma empresa relativamente nova, a área de tecnologia já foi criada utilizando técnicas e modelos de gestão com características modernas e dinâmicas, como por exemplo a utilização dos métodos ágeis como modelo de gestão de desenvolvimento de *software*, em quase todos os projetos que envolvem a área de tecnologia.



Figura 21 - Processo de Desenvolvimento Empresa Médio Porte
Fonte: Adaptado dos documentos internos da empresa

Conforme a Figura 21, o processo de desenvolvimento é bem simples, estruturado principalmente pelo *framework Scrum*, técnicas originadas do XP (*Extreme Programming*) e outras técnicas com princípios ágeis. As iniciativas geradas pelas áreas de negócio são cadastradas no sistema de gestão e em determinados períodos, com agendamento prévio e sem

padrão, as iniciativas são priorizadas. As iniciativas priorizadas passam por um processo de amadurecimento e combinação de temas. O resultado deste processo é uma lista denominada *Product Backlog*, que são enviados para desenvolvimento por meio de *Sprints*. Conforme as *Sprints* são concluídas, estes incrementos são incorporados no produto conforme orientação prévia definida no planejamento do *release*.

Este processo foi definido após diversas técnicas experimentadas, para garantir que não existam problemas de arquitetura de *software* ou grande esforço despendido com retrabalho. Também neste processo, foi definida a utilização das técnicas de *Design Thinking* pouco antes do processo de desenvolvimento para melhorar a definição dos requisitos.

[Entrevistado 4] Se existir é muito pouco (arquitetura e desperdício), pois o projeto tem um objetivo e o produto tem seu desenvolvimento orientado àquele rumo. No início utilizamos o Design Thinking apenas para definir requisitos e protótipos e depois de uma janela de tempo, iria para produção. Isso gerava muita mudança porque as pessoas mudavam de opinião, as pessoas mudavam e as necessidades mudavam também. Quando aproximamos as sessões de design para bem próximo do desenvolvimento, isso mudou, e hoje temos projetos que os times compartilham o mesmo espaço físico.

[Entrevistado 5] Se a Inception meeting estiver sendo bem realizada, não tem este problema (arquitetura e desperdício).

A primeira etapa do ciclo de desenvolvimento consiste em priorizar as iniciativas solicitadas ou sugeridas pelas áreas de negócio. Nesta etapa do processo, são utilizadas primeiramente técnicas como *MoSCoW*, que consiste em avaliar as solicitações e classificá-las de acordo com a sua importância em relação ao negócio ou de distribuição dos pontos que consiste em distribuir um número limitado de pontos entre todas as demandas e

posteriormente classificar de ordem decrescente as demandas com maior número de pontos acumulados. Porém outros fatores podem influenciar como questões políticas ou marcos regulatórios, que envolvem legislação.

[Entrevistado 4] Normalmente MoSCoW e distribuição de pontos

[Entrevistado 5] Depende de vários fatores, temos situação que é definido por meio de diretoria, requisitos legais ou ainda alguma técnica que quem estiver no comando do Inception sugerir em usar.

Nesta etapa também são definidos os indicadores de performance do projeto. Estes indicadores servem para medir o desempenho das iniciativas com relação ao negócio e também, satisfação dos usuários quanto ao que foi desenvolvido. Uma vez definidos, os indicadores são cadastrados em sistema e avaliados ao final do processo de desenvolvimento e implantação do *release*.

[Entrevistado 4] É um compliance do banco, ao fechar a Sprint ou o release, automaticamente é liberada uma pesquisa de satisfação. Temos que manter um índice acima de X, senão seremos prejudicados.

[Entrevistado 5] Os indicadores definidos para o release são colocados no sistema e ao final, o sistema solta pesquisa de satisfação. Não temos interferência sobre isso.

Uma vez que estas solicitações são priorizadas e selecionadas para entrar no processo de desenvolvimento, o próximo passo é a cerimônia de *inception* que consiste em um processo de imersão e desenvolvimento das iniciativas, gerando os requisitos funcionais, critérios de aceite e protótipos, que são os insumos para organizá-los, levando em conta as

afinidades técnicas, de modo que entrem para desenvolvimento nos times que melhor se alinham com as características de cada um dos itens. Estas sessões não possuem uma regra clara de execução com relação a sua duração ou agenda, são fatores que dependem de cada situação de projeto ou iniciativa que influenciam e determinam estas variáveis.

[Entrevistado 4] Depende do tamanho do projeto. Normalmente são sessões de 4 horas, mas dependendo da complexidade do projeto pode ter várias sessões, contínuas ou não, dependendo do andamento..... quando o projeto está em curso, temos as sessões para refinar o que precisa ser feito dentro do release ou da onda, quando o projeto é novo e ainda não tem as definições necessárias, são feitas algumas sessões também...

[Entrevistado 5] Inicia-se com uma sessão, mas pode ocorrer mais que isso. Depende de uma série de fatores.

Nessa cerimônia de *inception* são utilizadas as técnicas de *Design Thinking* para definição e elaboração dos requisitos funcionais e não funcionais. Participam deste evento os usuários chaves que solicitaram as iniciativas escolhidas para desenvolvimento, o corpo técnico e também os gestores dos produtos, que são responsáveis pelo ciclo de vida do produto e sua orientação mercadológica e a orientação estratégica do produto frente a estratégia de empresa.

[Entrevistado 4] Sim. As áreas envolvidas enviam seus representantes e também o time técnico.

[Entrevistado 5] Todos os envolvidos participam, sem ser desta maneira não é Design Thinking, salvo demandas que envolvem membros executivos, estes por sua vez enviam prepostos...

Uma vez definido os requisitos, eles são escritos em formato de *User Stories* e catalogados em uma lista definida como sendo o *Product Backlog* de desenvolvimento. Dependendo do tamanho desta lista, é definida a estratégia de desenvolvimento com relação aos times, como por exemplo, onde são definidos, quantos e quais os times de desenvolvimento serão alocados para executar esta lista.

[Entrevistado 4] Os protótipos e soluções criadas no Design Thinking entram numa fila para desenvolvimento, normalmente já existem times disponíveis para absorver a demanda, porém existem exceções.

[Entrevistado 5] Os requisitos são transformados em User Stories e estimados de maneira superficial para definir qual será a estratégia de desenvolvimento com relação aos times e a quantidade de times, inclusive contratação de mão de obra terceirizada, se necessário.

Na estratégia criada para a entrega das demandas, também são definidos os tamanhos das *Sprints*, que sempre tem como objetivo realizar um incremento que gere valor ao negócio. Porém fica a critério das áreas demandantes a opção de implantar ou receber este incremento ou aguardar uma sequência de incrementos.

[Entrevistado 4] O objetivo da Sprint tem que gerar incremento que faça sentido a área de negócio. Estes objetivos são definidos antecipadamente com granulação pertinente ao tamanho da Sprint

[Entrevistado 5] Sim, porém a área demandante tem a opção de não levar para publicação.

Durante a execução das *Sprints* são realizadas reuniões onde o produto que está sendo desenvolvido é apresentado para os usuários que demandaram os requisitos para coleta de *feedback* e pré aprovação. Neste momento podem surgir, além de eventuais correções que são absorvidas pelo time, mudanças que necessitem ser melhor analisadas ou ainda mudanças que não são possíveis de absorver dentro da *Sprint*. Nestes casos, estas mudanças deverão seguir o fluxo de uma nova iniciativa.

[Entrevistado 4] Este processo é feito pelo Product Owner e ele usa critérios próprios para isso. Normalmente é o fator que impacta mais o negócio ou o poder político que o solicitante tem, em conjunto com a capacidade do time.

[Entrevistado 5] Se o time conseguir absorver dentro da Sprint, será feito sem mais delongas ou se não conseguir absorver, seguirá o caminho de uma nova iniciativa.

4.3. SOFTWARE HOUSE

A *Software House* é uma empresa de tecnologia, com objetivo de prover soluções para as empresas por meio do desenvolvimento de produtos com tecnologia de ponta, consultoria de processos de T.I. e gestão e alocação de profissionais da área. É uma empresa reconhecida na área por possuir uma equipe de aproximadamente 1.000 profissionais e mais de duzentos cases de sucesso em seu portfólio.

Dentre diversos pilares de negócio da empresa, o que foi estudado nesta pesquisa foi à frente de desenvolvimento de produtos. Nesta frente são desenvolvidos produtos para seus clientes por meio de soluções para plataformas digitais e também, sustentação e manutenção das plataformas já existentes.

Porém, este modelo de negócio tem a particularidade de as demandas serem contratadas pelos clientes como um projeto e, devido a isso, foram feitas algumas adaptações nos métodos utilizados para atender a necessidade do negócio. Também a empresa entende que qualquer demanda que seja enviada a ela é tratada como um projeto, independente do seu tamanho ou complexidade.

[Entrevistado 6] *Gostaríamos muito de utilizar os Scrum ou qualquer outro método do modo original, mas como somos contratados por projeto, nossa unidade de medida é hora e recebemos pelo que foi aprovado, temos que seguir etapas obrigatórias que fogem um pouco do modelo.*

[Entrevistado 6] *Quando existe uma demanda. É a primeira etapa do projeto.*

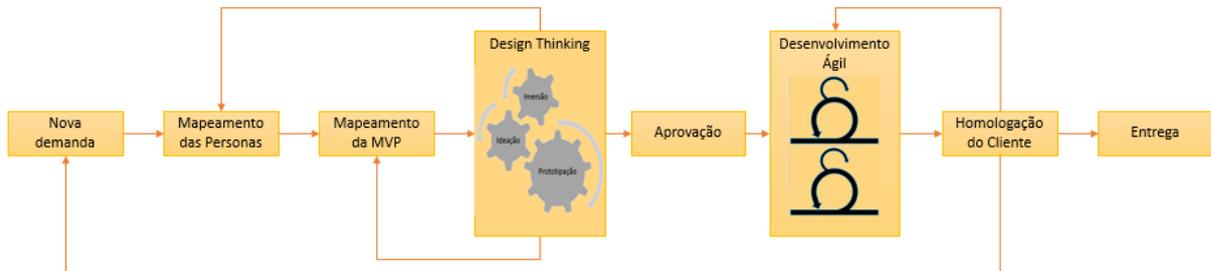


Figura 22 - Modelo de Desenvolvimento - *Software House*
Fonte: Elaborado pelo Autor

Quando uma nova demanda é solicitada pelo cliente, o primeiro passo é agendar uma reunião com o solicitante e também todas as pessoas envolvidas. Esta reunião tem dois objetivos: (i) identificar a influência de cada um no projeto e (ii) mapear o MVP (*Minimum Viable Product*) que consiste em produzir uma versão simplificada do produto que possibilite a coleta de feedbacks (Ries, 2012) e, por consequência, os principais requisitos ou funcionalidades além das premissas e restrições do projeto.

[Entrevistado 6] ...ocorre no momento de inicialização do projeto. São realizadas algumas sessões para posteriormente fazer o orçamento e desenvolver.

[Entrevistado 6] Dura o tempo que for necessário, como default marcamos uma agenda de 4 horas. Normalmente utilizamos desenhos, post-it e, para finalizar o protótipo inicial, fazemos figuras básicas com breves descrições em PowerPoint.

O analista de negócios, utilizando o resultado e as informações colhidas nas reuniões anteriores, agenda e realiza junto aos demais membros do time as sessões de *Design Thinking* seguindo os passos convencionais como imersão, ideação e prototipação. Estas sessões não possuem um prazo fixo ou número de sessões limitadas para acontecer. Normalmente participam destas sessões os colaboradores que participaram da reunião com o cliente e o time técnico. Às vezes, dependendo da complexidade, o cliente também é convidado a participar. São utilizadas ferramentas de prototipação, *mindmap* e outras técnicas para elucidação dos requisitos funcionais.

[Entrevistado 6] Cliente faz a solicitação e o time técnico é responsável pela solução. Muitas vezes a área de negócio é representada pelo analista de negócios. É muito difícil fechar o assunto dentro de uma única reunião, existe a dificuldade de agenda entre os envolvidos. O processo existe, mas são necessárias algumas reuniões para fechar o assunto.

[Entrevistado 6] As sessões ocorrem no início do projeto para definição dos requisitos, depois são descritos na documentação, orçadas e se aprovadas.

Os protótipos e características definidos nesta etapa são refinados e passados para o cliente em forma de proposta comercial, com o orçamento. Este orçamento pode ou não possuir limitação de tempo e escopo para sua conclusão, ou seja, o orçamento passado pode

ser para atender a um determinado escopo, com tempo e custo definido, ou ainda, ser apresentado orçamento em que o time técnico está disponível por um tempo limitado, exclusivamente para atender o cliente e, neste caso, a gestão do produto e o momento de finalizar o projeto seja de responsabilidade do cliente.

[Entrevistado 6] Uma vez definido e o cliente aprovando os protótipos ou POC's é colocado em processo de desenvolvimento.

Quando o cliente aprovar a proposta enviada, o projeto é encaminhado para a fase de desenvolvimento. Nesta etapa, os requisitos funcionais são descritos sob forma de *User Stories* e previamente organizado em entregas periódicas, determinadas ondas ou *releases*. Nesta etapa as histórias são executadas, seguindo a prioridade definida por algum critério técnico ou negociação com o cliente.

[Entrevistado 6] Como trabalhamos sob demandas de clientes, as demandas já chegam priorizadas... Algum critério de relevância combinado com o cliente no MVP é priorizado, o que atende minimamente o negócio conforme a persona mais relevante.

Conforme os requisitos ficam prontos, são disponibilizados para o cliente. Este processo é denominado de Homologação. Nele o potencial incremento ou *release* é implantando em um ambiente segregado do cliente, para que o mesmo possa avaliar o produto com relação ao que foi pedido e também checar a qualidade. Esse processo é particular de cada cliente; alguns utilizam equipe de controle de qualidade profissional ou ainda existem aquelas empresas onde o próprio cliente que solicitou as demandas avalia o que foi feito.

[Entrevistado 6] O incremento é gerado após atender as expectativas das personas envolvidas para sua posterior aprovação. Esta sequência de passos não está vinculada a Sprint, ou no caso do MVP as ondas ou jornadas.

[Entrevistado 6] Quando finalizamos o processo de desenvolvimento é enviado para o cliente para homologar; o retorno disso pode ser bug ou melhorias e até mesmo mudanças.

O processo de homologação pode gerar dois tipos de novas demandas: (i) *bugs*, que são erros de codificação ou programação errônea e, ainda, erros de entendimento do que foi solicitado em detrimento ao que foi feito; (ii) mudanças, neste caso são melhorias que os clientes solicitam, que não estavam previstas, ou ainda que determinada funcionalidade tenha o comportamento alterado ao que foi pedido. No caso da primeira condição, as novas demandas são absorvidas pelos times de desenvolvimento. Assim que forem concluídas, são disponibilizadas para um novo ciclo de homologação. Já no caso da segunda situação, a nova demanda é encaminhada como se fosse uma nova demanda, passando por todo o fluxo novamente.

[Entrevistado 6] O Design Thinking ajuda muito na questão da definição dos requisitos e alinhamento das expectativas, mas não garante que tudo que for feito não vai ter mudanças. Existe muito problema de interpretação e entendimento.

Neste modelo de desenvolvimento não existe métrica de avaliação padronizada pela empresa, apenas quando negociada com o cliente. Normalmente, de modo aleatório, é enviado para o cliente uma pesquisa de satisfação com o objetivo de avaliar a condição geral do projeto e a satisfação do cliente no relacionamento com a empresa.

[Entrevistado 6] Não é claro isso, mas no final do projeto temos uma pesquisa de satisfação onde o cliente (persona mais influente) responde falando se gostou ou não, e se nos indicaria para outros clientes.

4.4. STARTUP

Seguindo a definição que *Startup* é uma empresa projetada para o desenvolvimento de novos produtos e/ou serviços em ambientes cercados de incerteza (Ries, 2012), a *Startup* se enquadra neste perfil. Ela atua no ramo de tecnologia e produtos digitais. Constituída há menos de 5 anos, a *Startup* tem como objetivo desenvolver produtos orientados a tecnologia IoT (*Internet of Things*), gerando informações que alimentam um imenso banco de dados de informações aleatórias conhecido como *Big Data* e a partir destes dados gerar algum tipo de informação relevante para seus clientes.

Esta empresa possui uma estrutura simples e enxuta, o que permite que atualmente seu quadro de funcionários não ultrapasse os 50 colaboradores. Uma característica importante deste tipo de empresa que está iniciando o desenvolvimento de um mercado inédito, é que os projetos são cercados de incertezas.

A principal diferença entre a definição de risco e incerteza está na possibilidade de estabelecer ou prever seu impacto. Riscos são eventos que podem ocorrer e permitem que sejam previstos orçamentos para compensá-los. No caso das incertezas, isso não é possível devido às condições desconhecidas, inclusive devido à falta de conhecimento em que estes eventos podem ocorrer (Perminova et al., 2008).

O modelo de negócio utilizado por esta empresa, exige que o empreendedor, que tem um papel central na organização, tome decisões partindo de decisões genéricas, interdependentes e simultâneas, que possibilitem iniciar novos negócios. Estas decisões não

são baseadas em teoria prescrita ou de modo coerente, mas sim orientadas a um futuro imprevisível. Estas características que visam diminuir as incertezas estratégicas estão definidas na lógica *Effectuation* (Sarasvathy, 2001).

É nesta lógica de negócio, descrita por Sarasvathy (2001), que a *Startup* está estruturada, utilizando como base o *Design Thinking* para solucionar problemas e os métodos ágeis como modelo de gestão de desenvolvimento de *software* (Sarasvathy, 2001).

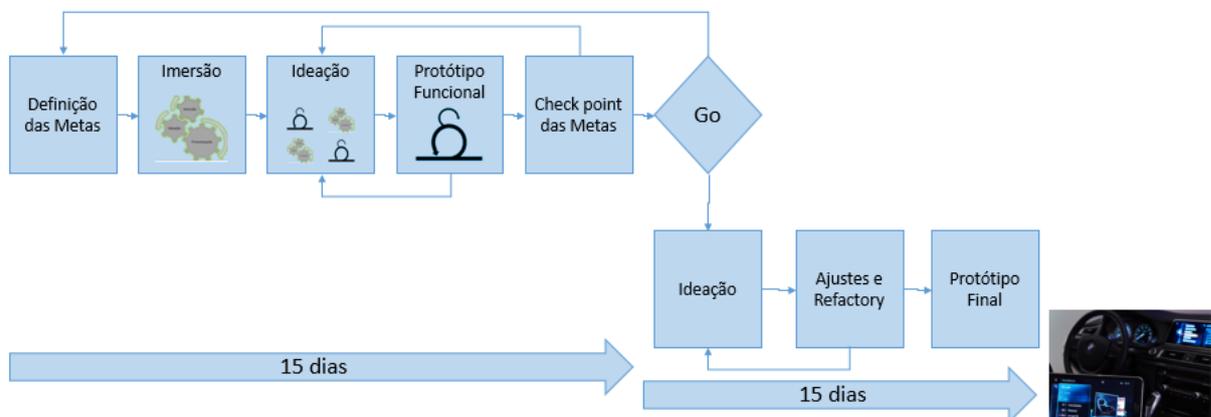


Figura 23 - Processo de Desenvolvimento - *Startup*
 Fonte: Adaptado dos documentos internos da empresa

O fluxo de desenvolvimento de *software* da *Startup* é definido por uma *timeline* com prazo fixo de 30 (trinta) dias corridos, possui sentido único e não sequencial onde cada um dos passos é realizado sempre que necessário, exceto as reuniões de definição e confirmação de metas. O processo é iniciado na definição das metas para os próximos 30 dias e destas metas, são derivadas as demandas. Este período é dividido em duas janelas quinzenais. Esta divisão tem como objetivo fazer uma revisão das metas e da viabilidade das soluções propostas e construídas até o momento, ou se ainda estiverem válidas, o fluxo segue por mais quinze dias até gerar um incremento ao produto da empresa. A primeira etapa deste fluxo, descrito na Figura 23, trata-se da definição das metas a serem atingidas neste ciclo de desenvolvimento. No passo seguinte, as pessoas envolvidas iniciam uma sequência de

atividades com o objetivo de estudar e entender como funcionam os itens envolvidos, as tecnologias necessárias, e demais atributos que possam auxiliar na definição da melhor solução para as funcionalidades necessárias, a fim de atingir as metas estabelecidas no passo seguinte, de ideação. A etapa de ideação e protótipo funcional é realizada em paralelo, e em constante interação. Na segunda quinzena, o primeiro passo é realizar uma reunião onde são revalidadas as metas traçadas no início do ciclo e uma vez que elas sejam confirmadas, continuam o processo de ideação, porém com viés de melhoria de desempenho, segurança e ajustes funcionais.

São três pontos de entradas das demandas: (i) comercial, que são provenientes de prospecção de algum cliente em potencial; (ii) produto, que são as demandas originadas de mercado, ou colhidas por meio de *feedback* recebido pelos canais de sugestões; e por fim, (iii) técnico, que são as demandas originadas do time técnico e que buscam melhorar a tecnologia, arquitetura ou performance do produto.

As demandas são listadas em um único local, numa planilha de controle de *Product Backlog*, onde são informados de modo simplificado a origem, descrição simplificada, objetivo, justificativa e data, conforme é possível observar na Figura 24.

ID	Descrição	Origem	Data	Objetivo	Justificativa	Status
1	XXXXXXX	Comercial	01/01/17	Medir vibração do dispositivo	Solicitado pelo cliente X que é uma seguradora e gostaria de monitorar a carga	Em processo
2	XXXXXXX	Produto	01/01/17	Medir consumo de energia e temperatura	Solicitação enviada no <u>hackaton</u>	Aguardando
3	XXXXXXX	Produto	01/01/17	Permitir dispositivo com IOS	Mercado	Rejeitada

Figura 24 - *Product Backlog - Startup*
Fonte: Adaptado dos documentos internos da empresa

A primeira atividade do ciclo de desenvolvimento é definir quais itens do *Product Backlog* irão entrar para desenvolvimento. Isso é feito por meio de uma reunião onde os

representantes das áreas comercial, produto, desenvolvimento e administração corporativa participam.

[Entrevistado 7] O processo se inicia com time-box. Os itens são gerados por meio de pesquisa de mercado e solicitações dos clientes e um comitê composto por um colegiado formado pela alta direção da empresa, marketing, gestor do produto e time técnico.

[Entrevistado 8] Sempre que um ciclo iniciar são definidos conforme necessidades de mercado e solicitações. Utilizamos o Canvas para mapear os impactos...

Utilizando técnicas de análise de mercado e produto como o Canvas, exemplificado na Figura 25, e matriz SWOT exemplificado na Figura 26, os itens são priorizados conforme orientação deste comitê.

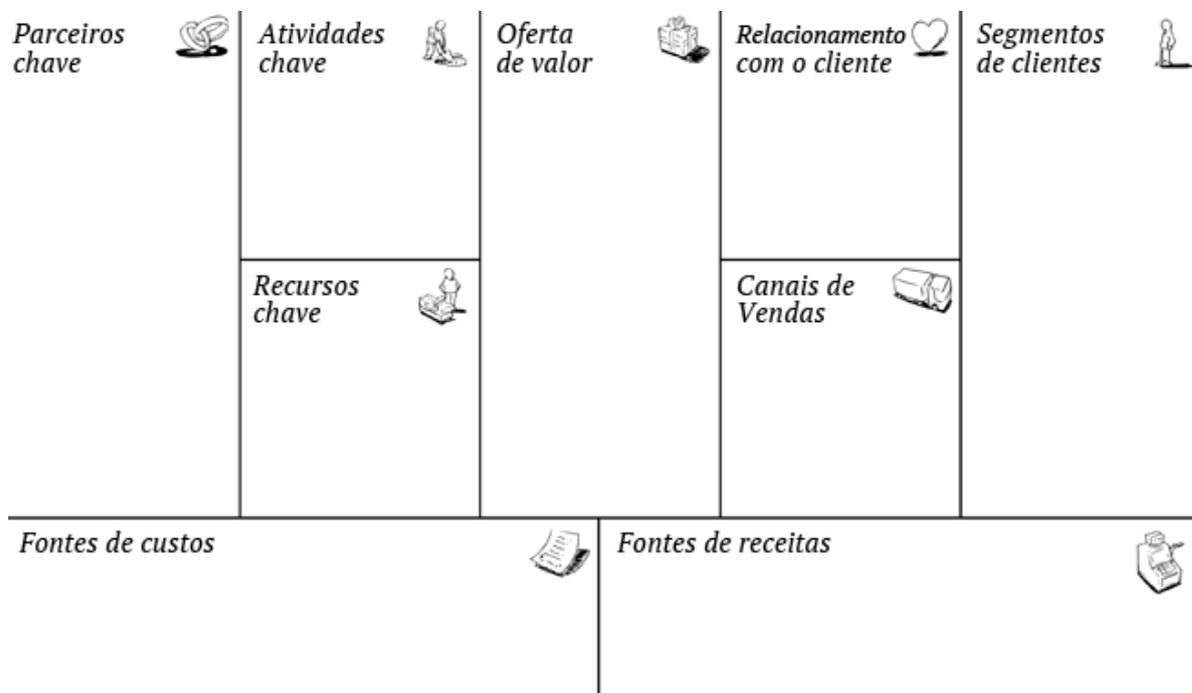


Figura 25 - Canvas

Fonte: Adaptado dos documentos internos da empresa

SWOT	POSITIVOS	NEGATIVOS
INTERNOS	Pontos Positivos do nosso produto	Pontos Negativos do nosso produto
EXTERNOS (Concorrentes)	O que o nosso produto tem que o dos concorrentes não tem? O que nosso produto faz melhor que o dos concorrentes?	O que produtos dos concorrentes fazer melhor que o nosso? O que o produtos dos concorrentes tem que o nosso não tem?

Figura 26 - Matriz SWOT da *Startup*
Fonte: Adaptado dos documentos internos da empresa

[Entrevistado 8]... e colocamos pesos em cada uma das áreas e com isso é priorizado.

Uma vez traçadas as metas para o produto, o próximo passo é a etapa de imersão. Nesta etapa, o time técnico estuda e discute as demandas para propor soluções que obrigatoriamente, tem que ser feitas dentro de um ciclo de desenvolvimento. Esta obrigatoriedade deve-se a lógica da *Effectuation*, que propõe o que tem que ser desenvolvido, tem que ser feito rápido e com baixo investimento e num segundo momento ele pode ser melhorado.

[Entrevistado 7] Estudamos (time técnico) uma maneira de solucionar o problema com uma proposta que seja possível fazer com o menor investimento possível, porque existe o risco de a solução não servir, ou ainda a demanda não ter utilidade no mercado.

[Entrevistado 8] Nossa solução é construída como um protótipo melhorado, é mais que um protótipo, mas não é o produto final.

O processo de ideação consiste no processo de desenvolvimento de *software* propriamente dito. É a etapa onde as soluções propostas na fase de imersão são materializadas e colocadas a prova. Neste processo são utilizadas práticas de *Scrum* e *Extreme Programming (XP)*, principalmente com relação as cerimônias de alinhamento, a *daily meeting*, uso de *kanbans* para gestão visual do progresso e gráficos de indicadores como o *Burndown*.

[Entrevistado 7] O processo de Design Thinking é estruturado dentro de um ciclo de 30 dias. Dentro deles são feitos vários pequenos ciclos de desenvolvimento ágil logo, não existe integração entre o ágil e Design Thinking, eles ocorrem juntos.

O produto desenvolvido na etapa de ideação é um protótipo que contempla as necessidades solicitadas, com aspecto rudimentar e primário, mas que permite que os clientes e seus usuários utilizem e gerem *feedbacks* iniciais.

[Entrevistado 8] ...para nós é a mesma coisa. Dentro do ciclo, temos atividade de Design e desenvolvimento, nosso protótipo é um produto de venda.

Assim como sugere os métodos ágeis, é necessário validação e alinhamento constantes do que está sendo desenvolvido com as necessidades de mercado ou ainda estratégia da empresa. Ao final da primeira quinzena é realizada uma reunião para verificar se o que está sendo desenvolvido ainda está alinhado com a estratégia da empresa e/ou produto; se estiver, o fluxo continua por mais uma quinzena, do contrário podem ser feitos ajustes e mudanças, ou até mesmo descartar o que foi feito e iniciar outro ciclo de desenvolvimento.

[Entrevistado 7] Ocorre quinzenalmente, normalmente na segunda sexta-feira do mês. Revalidamos o que está sendo feito, se ainda está de acordo com a estratégia da empresa, ou para traçar uma nova linha. Se necessário, mudamos ou começamos de novo.

[Entrevistado 8] Tem que ser em uma única sessão, e tem que ser rápida, validamos, ajustamos e segue o barco ou não.

Durante a segunda quinzena, o processo de desenvolvimento continua. São feitos os ajustes necessários e também são enfatizados os ajustes de arquitetura e performance, esta última de modo superficial no caso de nova funcionalidade.

[Entrevistado 7] Precisamos tomar cuidado com a arquitetura e qualidade do código pois, como estamos construindo algo que não existe e não sabemos onde iremos chegar, existe sim muito retrabalho e problemas de arquitetura. No modelo Effectuation ele prevê isso.

[Entrevistado 7] É um processo evolutivo e com isso podemos ter que refazer algumas coisas. É fundamental termos profissionais competentes para ver quando isso está ocorrendo e inserir no processo tarefas de refactory, para fazer os ajustes.

[Entrevistado 8] É um produto que está nascendo, é impossível fazer algo definitivamente. Ocorre o tempo inteiro (problemas de arquitetura e desperdício), temos que ficar atentos a isso e fazer ajustes sempre que necessário, até mesmo a reescrita de algumas funcionalidades.

Ao final dos 30 dias é possível ter um protótipo que pode ser incorporado ao produto que está no mercado e também disponibilizado para avaliação no mercado por meio de feiras

e eventos. No caso de demandas solicitadas para os clientes, é disponibilizada uma versão para o cliente utilizar e avaliar.

[Entrevistado 7] O produto desenvolvido no final do ciclo é apresentado em feiras e eventos ou ainda para os clientes. Este feedback ou solicitações entram para discussão no início do processo.

[Entrevistado 8] Colocamos ele a disposição dos clientes quando é uma solicitação ou ainda em grandes eventos para coleta de feedback e novas ideias.

4.5. COMPARAÇÃO ENTRE OS MODELOS DAS EMPRESAS

Neste item será feita comparação entre as características dos processos de desenvolvimento de *software* obtidos nas entrevistas e descritos nos capítulos anteriores. Serão destacados os processos semelhantes e os divergentes das empresas estudadas.

4.5.1. Frameworks e técnicas ágeis utilizadas

Os modelos apresentados pelas empresas não possuíam apenas um *framework* ou alguma técnica específica. O que existe são modelos de desenvolvimento que seguem os princípios ágeis, de modo que as utilizações, uma combinação de diversos modelos e técnicas, resultem em um modelo aderente às necessidades de negócio.

Na Tabela 1 é possível verificar que o *Scrum* é utilizado por todas as empresas, mas também é notório que nenhuma empresa utiliza este *framework* isolado. Todas combinam pelo menos com outro modelo, sendo que o mais utilizado é o MVP (*Minimum Viable Product*). Apenas uma empresa não o utiliza, sendo que esta, por sua vez, utiliza modelos de

desenvolvimento ágil em escala, caso do SAFe (*Scaled Agile Framework*) e do DaD (*Disciplined Agile Delivery*).

Tabela 1 - Frameworks e Técnicas Utilizadas

Framework / Técnica	Empresas				Total
	1	2	3	4	
Scrum	X	X	X	X	100%
XP (<i>Extreme Programming</i>)	X	X	X	X	100%
MVP (<i>Minimum Viable Product</i>)		X	X	X	75%
Distribuição dos Pontos		X			25%
WSJF (<i>Weighted Shortest Job First</i>)	X				25%
SAFe (<i>Scaled Agile Framework</i>)	X				25%
DaD (<i>Disciplined Agile Delivery</i>)	X				25%
Effectuation				X	25%

(1) Empresa de Grande Porte – (2) Empresa de Médio Porte – (3) *Software House* (4) *Startup*

Existe outro padrão com relação à utilização das técnicas ágeis empregadas no desenvolvimento de *software* nas empresas estudadas, que é a utilização do XP, dado que todas as empresas o utilizam. Para priorização, foram citadas duas técnicas: a de distribuição dos pontos e também a WSJF (*Weighted Shortest Job First*), porém apenas 25% das empresas as utilizam. Surgiu também a lógica de *Effectuation*, em na *Startup*, técnica utilizada para desenvolvimento de produtos e novos negócios em ambientes de grande incerteza.

4.5.2. Utilização do *Design Thinking*

Conforme apresentado na Tabela 2, é confirmado que todas as empresas utilizam o *Design Thinking*, mas nenhuma delas o utiliza de maneira completa. As características que se destacam são a empatia, a utilização de protótipos e a constante coleta e utilização de *feedbacks* na melhoria do produto. Dentre as características menos presentes estão as que são vinculadas a comunicação, que são a interação entre os membros das equipes e outras equipes e a utilização de meios diferentes de comunicação, que não seja a conversa por meio de reunião ou *feedbacks*. Não é possível definir o motivo que isso ocorre com precisão, mas

existe a possibilidade que a causa seja o não conhecimento sobre o *Design Thinking* com profundidade ou ainda o anseio em ceder informações aos pares ou ainda demonstrar falta de conhecimento sobre o produto.

Tabela 2 - Características do *Design Thinking* Encontradas

Característica <i>Design Thinking</i>	Empresas				Total
	1	2	3	4	
Empatia	X	X	X	X	100%
Desenvolvimento de Protótipos por meio da transformação de ideias	X	X	X	X	100%
Coleta de <i>feedbacks</i> e consideração dos mesmos para amadurecimento das ideias	X	X	X	X	100%
Envolvimentos dos <i>stakeholders</i> desde o início do processo	X	X		X	75%
Participação dos <i>stakeholders</i> da definição dos requisitos e protótipos	X	X	X		75%
Alinhamento entre time técnico e <i>stakeholders</i>		X	X	X	75%
Interação entre os membros das equipes		X		X	50%
Interação entre as equipes		X		X	50%
Utilização de Formas Diferenciadas de Comunicação		X	X		50%

(1) Empresa de Grande Porte – (2) Empresa de Médio Porte – (3) *Software House* (4) *Startup*

4.5.3. Priorização e Ideação das Iniciativas

Este processo consiste numa etapa na qual são organizadas todas as iniciativas. São priorizadas para, posteriormente, serem analisadas com mais profundidade e com isso, promover a definição dos requisitos, protótipos e insumos suficientes para o processo de desenvolvimento seguinte.

Tabela 3 - Priorização e Ideação

Priorização e Ideação	Empresas				Total
	1	2	3	4	
Participação do time de negócio (solicitante)	X	X	X	X	100%
Utilização de ferramentas criativas e leves (<i>mindmap</i> , <i>post-its</i> , desenhos)	X	X	X	X	100%
Coleta de <i>feedback</i> antes de enviar ao desenvolvimento	X	X	X	X	100%
Técnicas simples de priorização	X	X	X	X	100%
Utilização de Protótipos simplificada (desenhos e navegação por meio de imagens)	X	X	X		75%
Múltiplas Sessões		X	X		50%
Sessões com objetivo único (apenas uma área, produto, modulo, etc..)	X		X		50%
Sessão única	X			X	50%
Participação do time técnico completo (<i>design</i> , arquitetura e desenv)		X		X	50%
Utilização de Protótipos avançados (POC's, Protótipos navegáveis, etc..)	X			X	50%
Sessões com objetivo corporativo		X		X	50%
Sessões de no máximo 4 horas				X	25%
Utilização de ferramentas criativas e avançadas (<i>softwares</i> , etc)	X				25%
Técnicas avançadas de priorização	X				25%

(1) Empresa de Grande Porte – (2) Empresa de Médio Porte – (3) *Software House* (4) *Startup*

As características mais predominantes no processo de ideação e priorização, como pode ser visto na Tabela 3, são: (i) a utilização de ferramentas simples de prototipação, como desenhos manuais e feitos em *software* de imagens, sem a preocupação de navegação ou consistência dos dados; (ii) a participação da área solicitante na solução dada para a iniciativa apresentada e (iii) a coleta de *feedbacks* com pessoas que participaram da solução, antes de enviar para o desenvolvimento. O modelo de priorização simplificado consiste na priorização por meio de determinação, por uma área ou pessoa, sem a utilização de regras claras ou modelos de mercado, neste caso, em todas as empresas existe este tipo de situação.

Referente aos protótipos, é unanime a necessidade da sua utilização, porém, não houve consenso entre a utilização de protótipos simplificados ou complexos, inclusive existem empresas que utilizam os dois tipos de prototipação.

Com relação às sessões de ideação e priorização, também não houve característica determinante com relação aos objetivos a serem tratados em cada uma das cerimônias e também quanto a sua duração.

4.5.4. Desenvolvimento do *software*

Nesta etapa são tratados os processos de desenvolvimento das iniciativas, desde a aprovação do protótipo até a entrega final para o solicitante. Algumas características do modelo de desenvolvimento ágil estão presentes em todas as empresas estudadas, conforme é demonstrado na Tabela 4. Estas características são: (i) absorver pequenas mudanças em tempo de execução do desenvolvimento; (ii) utilização de times multidisciplinares; (iii) coleta de *feedbacks* periódicos durante o desenvolvimento da solução e (iv) previsão de esforço com atividades de melhoria de código fonte (*refactory*).

Tabela 4 - Desenvolvimento

Desenvolvimento	Empresas				Total
	1	2	3	4	
Mudanças de pequeno impacto são absorvidas pelo time de desenvolvimento	X	X	X	X	100%
Os times são multidisciplinares	X	X	X	X	100%
São colhidos <i>feedbacks</i> durante o processo de desenvolvimento	X	X	X	X	100%
O modelo prevê desperdícios gerados com retrabalho (<i>refactory</i>)?	X	X	X	X	100%
Utilização de <i>buffer</i> entre o processo de ideação e desenvolvimento	X	X	X		75%
O processo de ideação e desenvolvimento é sequencial	X	X	X		75%
Mudanças de grande impacto são absorvidas pelo time de desenvolvimento com autorização do <i>Product Owner</i> e revisão do plano de entrega		X	X	X	75%
O <i>release</i> é composto por entregas de mais do que um time ou iniciativas distintas	X	X			50%
Mudanças de grande impacto não absorvidas e são criadas novas iniciativas	X				25%
Processo de desenvolvimento sincronizado com o processo de ideação				X	25%
Os solicitantes podem utilizar o que já está pronto durante o desenvolvimento para avaliação e <i>feedbacks</i>			X		25%
Os incrementos gerados ao final da <i>Sprint</i> são utilizáveis				X	25%
O modelo prevê problemas de arquitetura de <i>software</i>				X	25%

(1) Empresa de Grande Porte – (2) Empresa de Médio Porte – (3) *Software House* (4) *Startup*

Há outras características que são comuns no processo de integração entre os processos de ideação e desenvolvimento. A primeira delas é o fato de serem sequenciais e também como é o caso da falta de sincronia entre elas e a necessidade de utilização de um *buffer* para não parar o processo de desenvolvimento. Com relação ao incremento gerado ao final das *Sprints*, a maioria das empresas não geram entregas funcionais, e normalmente são necessárias mais que uma *Sprint* para gerar uma entrega que faça sentido para a área demandante. Outro ponto

que a maioria das empresas adotou, foram processos que garantem que não haverá problemas de arquitetura de *software*.

4.5.5. Comparação dos Fluxos

Nesta etapa será demonstrado e comparado o fluxo de desenvolvimento das iniciativas, com o objetivo de comparar como funciona a dinâmica do processo, partindo da sua criação até a sua integração no produto atual ou lançamento do novo produto.

Na Figura 27, é possível verificar que até a etapa de priorização, o processo é semelhante a todas as empresas estudadas, exceto na *Software House*. Isso se deve ao modelo de negócio da empresa, que trabalha sob encomenda dos seus clientes e, portanto, a etapa de priorização não é necessária.

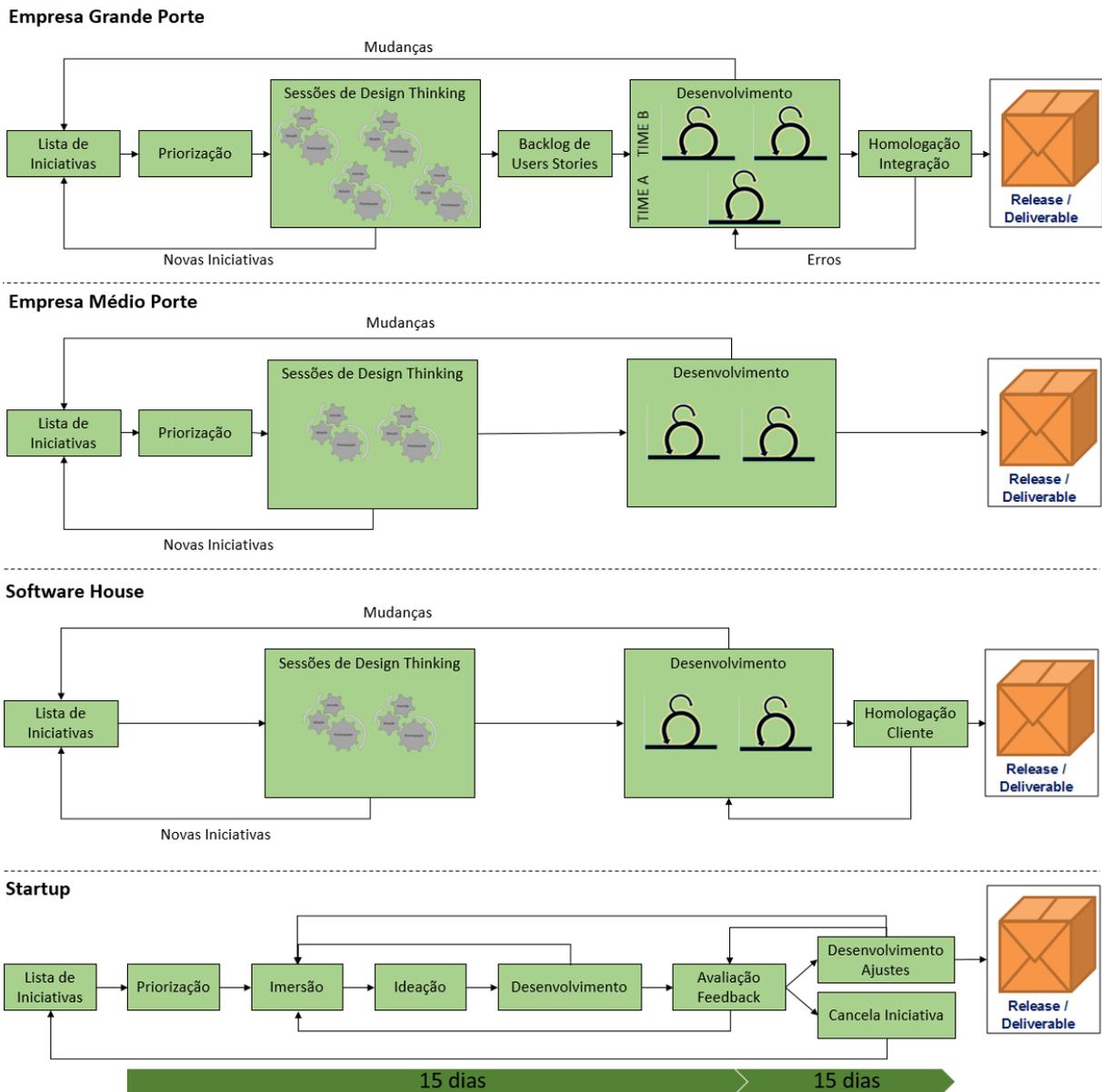


Figura 27 - Comparação do Processo de Desenvolvimento
Fonte: Elaborador pelo Autor

O processo seguinte são as sessões de *Design Thinking* que ocorrem de maneira distinta nas empresas. Na Empresa Grande Porte o processo ocorre de maneira aleatória, sem se preocupar com a capacidade de desenvolvimento. Isso, inclusive, permite que exista variação dessa capacidade, e desta maneira gera a necessidade de ter um pulmão (*buffer*) de *User Stories* aguardando para entrar no processo de desenvolvimento. Na Empresa Médio Porte e na *Software House*, o processo é sincronizado com o processo de desenvolvimento, ou seja, as sessões ocorrem de acordo com a capacidade de desenvolvimento. Na *Startup*, o

processo de *Design Thinking* ocorre dentro do processo de desenvolvimento, ou seja, não existe diferença entre o processo de *design* e o processo desenvolvimento.

O produto gerado no processo de desenvolvimento passa por processo de homologação e integração na Empresa Grande Porte, já nas demais empresas, quando finalizado, é incorporado ao produto. As mudanças, quando geram impacto no planejamento do projeto, são enviadas para o início do processo como uma nova iniciativa.

4.6. ANÁLISES DAS PROPOSIÇÕES

Neste capítulo são apresentadas as discursões e validações das proposições indicadas no capítulo Síntese da fundamentação teórica e proposições.

P1. Os membros da equipe das áreas de negócio das empresas participam desde a fase inicial do projeto.

O envolvimento dos *stakeholders* desde as etapas iniciais do projeto é considerado fundamental no *Design Thinking*. Este processo de desenvolvimento e cocriação tem como principal objetivo utilizar a complexidade de uma situação criada por meio de diferentes pontos de vista e com isso, buscar solução de problemas ou requisitos que não claramente definidos, sejam usados com insumos para novas demandas ou melhorias dos produtos ou serviços que estão sendo discutidos no momento (Bonini & Sbragia, 2011; Brown, 2008; Council, 2005).

A proposição P1 foi validada, pois grande parte das empresas disponibiliza e prevê que os *stakeholders* participem do processo desde o seu início, inclusive na elucidação dos requisitos e definição da solução.

P2. A interação da área de negócio com a área técnica no grupo que participou do processo do *Design Thinking* ocorre com frequência para coleta de *feedback*.

Como ferramenta de experimentação e ideação do *Design Thinking*, a coleta de *feedback* está em entre as mais utilizadas. Esta ferramenta proporciona a evolução de ideias e a geração de protótipos bem elaborados de modo que atenda o maior número de requisitos e com mais qualidade (Brown, 2008). Utilizando ciclos de *feedback* de maneira interativa ao processo de desenvolvimento, reduz as chances e o risco de retrabalho ligados a problemas de entendimento ou interpretação (Ratcliffe, 2009).

Em 80% das empresas estudadas é possível verificar que existe frequente coleta de *feedback* tanto na fase de definição da solução, quanto na fase de desenvolvimento do produto. A exceção está na empresa que desenvolve produtos sob encomenda dos clientes; os *feedbacks* existem, mas não com tanta frequência. Os *feedbacks* são coletados de maneira diversificada e com frequência não padronizada. Com relação às pessoas envolvidas neste processo de coletas de *feedback*, estão diretamente ligadas ao grau de inovação que está empreendido no produto. Quanto mais próximo do inédito, mais restritos são os *feedbacks*, porém eles sempre existem. Desta maneira, a proposição **P2 está validada**.

P3. O protótipo criado no processo de *Design Thinking* é submetido a *feedbacks* dos usuários finais, antes da integração com os métodos ágeis e após o desenvolvimento.

O processo de imersão, ideação e prototipação proposto no *Design Thinking*, proporciona soluções por meio de múltiplas ações e pensamentos divergentes e por meio da interação entre os *stakeholders* (Brown & Katz, 2011). Este processo de amadurecimento das

ideias consiste num ciclo de refinamento gerado pelos usuários e corpo técnico, até a sua disponibilização para o mercado (Brown & Katz, 2011; Plattner, 2013).

A utilização de protótipos para a experimentação e testes do produto, além da coleta de *feedbacks*, é a base para a definição dos requisitos (Brown, 2008). Sua utilização torna possível a experiência gradativa com novas tecnologias, produtos ou serviços, de modo que testes possam ser realizados e confrontados com as ideias propostas. E com isso, o processo de aprendizado empregado seja evolutivo e assim, reduzir as incertezas e os riscos de insatisfação dos usuários.

Foi possível verificar que os protótipos são submetidos a *feedbacks*, porém, o momento em que ele é apresentado aos usuários depende do nível da inovação e estratégia de lançamento para o mercado. Se o produto ou a funcionalidade que estiver sendo desenvolvido não existir restrição quando a confidencialidade, todas as empresas estudadas submetem seus protótipos para *feedback* dos usuários finais e também, apresentam as primeiras versões desenvolvidas para estes usuários, sendo assim, **a proposição P3 foi validada.**

P4: A aplicação do método ágil no desenvolvimento de *software* possui como característica principal o fato de ser adaptável.

A utilização dos métodos ágeis nas empresas não é simples, pois existe a necessidade de superar o paradigma da proposta sugerida nos modelos para a realidade dos projetos (Abrahamsson et al., 2003). Os modelos criados pelas empresas permitem que a criatividade e a adaptação sejam ferramentas para solução dos problemas complexos que existem nos projetos.

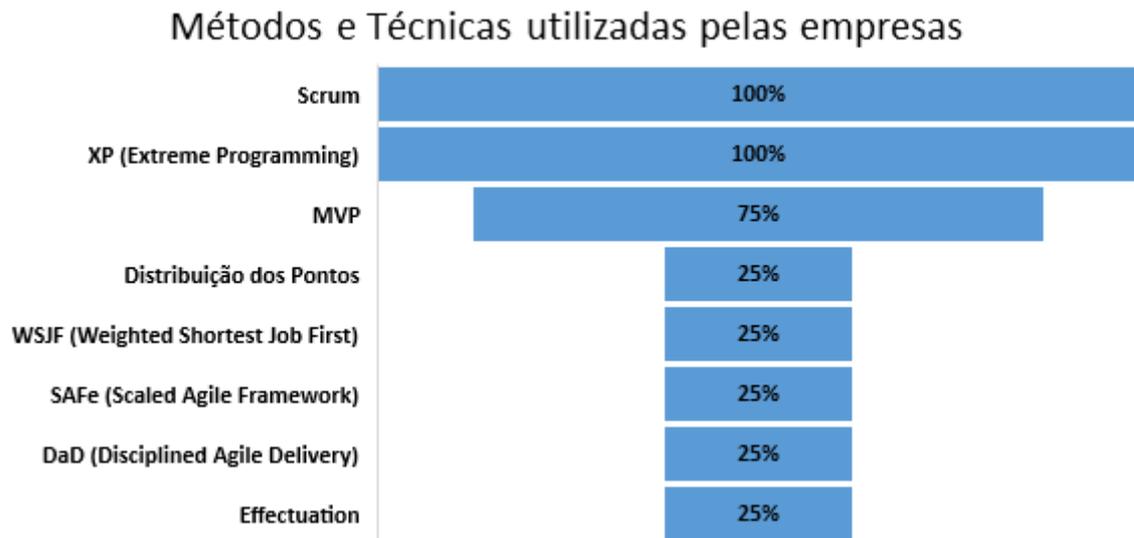


Figura 28 - Métodos e Técnicas utilizadas pelas empresas
Fonte: Elaborado pelo Autor

No estudo realizado, foi possível verificar que as empresas utilizam métodos diferentes para sustentar seu processo de desenvolvimento de *software*. Na Figura 28 é possível verificar que apenas o *framework Scrum* e as técnicas do XP estão presentes em todas as empresas. Porém, de acordo com as características das empresas, é utilizado outro modelo para dar escala em projetos mais complexos ou maiores, ou dar dinamismo quando existe a necessidade de inovação rápida. Desta maneira é correto afirmar que **a proposição P4 pode ser considerada válida.**

P5: O incremento de produto entregue ao final da interação ou *Sprint*, é um incremento viável para implantação e utilização dos usuários.

Um dos *frameworks* mais utilizados pelas empresas que foram estudadas é o *Scrum*. Este modelo é pautado por uma sequência de cerimônias previamente determinadas, e definido por *time-box* que são eventos com tempo determinado e fixo. A *Sprint* é o principal evento deste *framework* e consiste em uma interação que pode durar de 1 a 4 semanas, e ao

final deve ser entregue um incremento de produto utilizável (Abrahamsson et al., 2002; Schwaber & Sutherland, 2013, 2016).

No estudo, apenas 20% das empresas consegue gerar um incremento de produto após a conclusão da *Sprint*. Isso se deve a complexidade e tamanho das entregas e também ao processo de desenvolvimento adotado que pode utilizar diferentes e múltiplos times de desenvolvimento. Nas empresas o conceito de entrega por *releases* é mais utilizado. Desta maneira, **a proposição P5 não pode ser considerada válida.**

P6. A integração do *Design Thinking* e dos Métodos ágeis ocorre de modo sequencial.

Os métodos de *Design Thinking* são bastante utilizados no levantamento dos requisitos dos produtos ou ainda no refinamento das ideias e aprimoramento das soluções apresentadas ou sugeridas, de forma que a solução seja alinhada com as expectativas dos *stakeholders* (Bonini & Sbragia, 2011; Brown, 2008). Já os métodos ágeis possuem características que permitem que os processos sejam flexíveis, de maneira que seja possível fazer entregas para os clientes de maneira rápida e constante (Boehm & Turner, 2003; Serrador & Pinto, 2015).

A integração entre os dois modelos foi estudada detalhadamente neste trabalho. Segundo os entrevistados, existe grande dificuldade nesta integração devida, principalmente, ao ritmo dos trabalhos de cada um dos modelos e a não sincronia de momentos entre encontrar uma solução para o problema e viabilizar esta solução.

Para 60% das empresas, são necessárias mais que uma sessão de *Design Thinking* para concluir um único objetivo. Nestas sessões com duração acima de 4 horas, para 80% das empresas é uma grande dificuldade para conciliação de agendas entre todos os envolvidos. Existe também a necessidade de realização de pesquisa entre uma sessão e outra do *Design*

Thinking para coleta de *feedbacks*, ou busca de soluções inovadoras para determinados problemas.

Para solucionar este problema, 80% das empresas utilizam um *buffer* entre a solução do problema e o desenvolvimento da solução. Isso se deve também a diferença de complexidade entre propor a solução e desenvolver e incrementar esta solução no produto por meio de desenvolvimento de *software*. Outros problemas também são apontados, como problemas de desperdício com retrabalho, que foi apontado em todas as empresas e também a necessidade de projetar arquitetura de *software* que seja sustentável.

Por consequência destes pontos apresentados, 80% das empresas utilizam os modelos de maneira sequencial e assim **se faz válida a proposição P6**.

4.7. NOVO MODELO PROPOSTO

Na Figura 6 - Integração *Design Thinking* e Desenvolvimento Ágil, página 37, foi proposto um modelo conceitual de integração entre os métodos. Com os fatos e as características encontradas nas empresas estudadas, esse modelo foi atualizado para a versão apresentada na Figura 29. Este modelo abrange os processos encontrados nas empresas estudadas, exceto a empresa *Startup*, que possui um processo bastante particular.

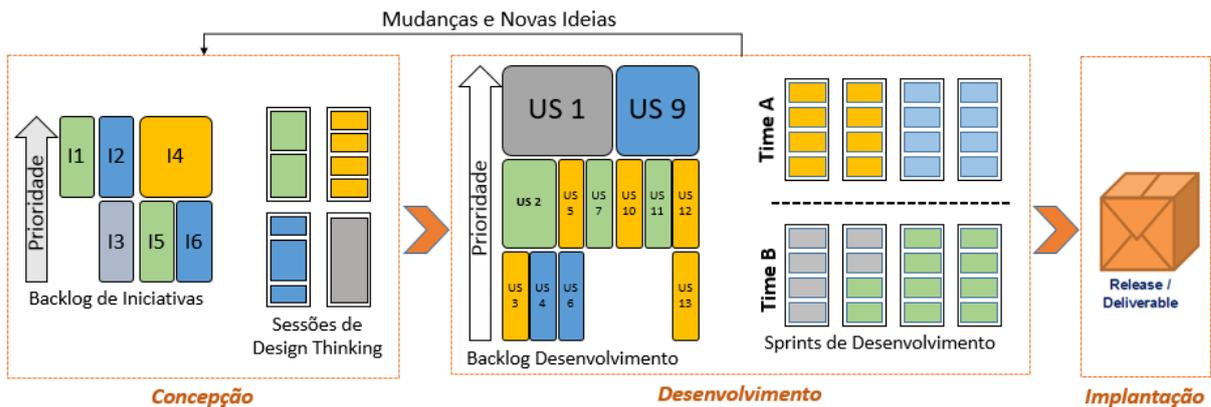


Figura 29 - Modelo Integrado - *Design Thinking* e Métodos Ágeis
 Fonte: Elaborado pelo Autor

Nesse novo modelo, o processo ocorre de maneira sequencial e assíncrona, como encontrados na maioria das empresas estudadas. O processo começa quando as iniciativas geradas pela empresa são catalogadas e classificadas por objetivo ou afinidade.

De posse de uma coleção de iniciativas com as mesmas características (objetivo ou afinidade), são criadas diversas sessões de *Design Thinking*, nas quais as iniciativas serão discutidas em cada uma das sessões. Nessas sessões, participam representantes de todos os *stakeholders* envolvidos, de modo que seja possível que as soluções criadas atendam as expectativas de todos. Essas sessões não tem um padrão comum de tempo e frequência, ou seja, serão criadas quantas sessões forem necessárias para elucidar todos os requisitos necessários para que a solução tenha a maior cobertura possível.

Uma vez que os requisitos estejam definidos nas sessões de *Design Thinking*, eles serão encaminhados para o processo de desenvolvimento de *software* por meio de protótipos e também com os requisitos escritos em forma de *User Stories* (US). Este material será classificado por afinidade e distribuído dentro dos times de desenvolvimento que atendam as características técnicas necessárias para o desenvolvimento das *User Stories*. Quando existir um pacote de *user stories* concluídas e passível de implantação, será gerado um pacote único (release) para implantação. Durante o processo de desenvolvimento, poderão surgir

mudanças. Se a mudança não tiver impacto dentro do planejamento de entregas, elas serão absorvidas pelos times de desenvolvimento. Na hipótese de a mudança impactar o planejamento de entregas, o responsável pela gestão do produto deverá decidir se absorve esta mudança refazendo o planejamento, ou se a mudança seguirá para o *backlog* de iniciativas como uma nova demanda. Ao final da fase de desenvolvimento, o produto já estará apto para implantação ou quando necessário, passar pelo processo de homologação.

Este novo modelo permitirá que as iniciativas sejam estudadas com profundidade e amadurecidas antes de entrar para o processo de desenvolvimento, o que poderá gerar menos problemas de arquitetura técnica ou necessidade de mudanças. Já no processo de desenvolvimento, é aberta a possibilidade da inclusão de times conforme a demanda, os objetivos ou afinidades, o que gera grande capacidade e escalabilidade no processo de desenvolvimento de *software*.

5. CONTRIBUIÇÕES PARA A PRÁTICA

As informações geradas por este estudo podem ser de grande valor para as empresas que estão passando por processo de transformação em sua área de tecnologia, de diferentes ramos de atividade e porte. O modelo proposto neste estudo pode ser utilizado como inspiração para a criação de nova estratégia de desenvolvimento de *software* da empresa. Esse novo processo irá gerar fluxo contínuo de geração de valor.

As soluções podem ser mais assertivas por meio de um processo que permita a cocriação entre as áreas de negócio e time técnico, para criar as soluções dos problemas por meio de desenvolvimento dos produtos e novas funcionalidades necessárias para o funcionamento do negócio. É provável que este modelo permita também que o processo de desenvolvimento de *softwares* das empresas tenha flexibilidade suficiente para atender as demandas de mercado, e ainda suporte as alterações e dinamismo existente na era digital.

Este modelo irá contribuir também para a redução do lead time de desenvolvimento dos produtos, uma vez que é previsto pequenos ciclos, com a participação dos principais interessados.

É possível, também verificar ganhos com relação a redução de desperdício de esforço com atividades de desenvolvimento com funcionalidades desnecessárias a área negócios das empresas, normalmente criadas para valorizar o produto, porém com pouca aplicabilidade na utilização corriqueira.

6. CONCLUSÃO

O objetivo deste estudo é identificar como o *Design Thinking* pode ser integrado com os métodos ágeis em projetos de desenvolvimento de *software*. Para atender esse objetivo, foi realizada uma pesquisa qualitativa, por meio de estudo de caso múltiplo, com quatro casos. Os dados foram coletados por meio de entrevistas em profundidades, com oito entrevistados e documentos cedidos pelas empresas. Cada uma das empresas escolhidas tem características únicas para que seja possível entender como ocorre a integração do *Design Thinking* com os métodos ágeis em diferentes ramos de atuação.

Foram encontradas as seguintes características do *Design Thinking* em todas as empresas: centradas no humano, por meio times multidisciplinares para a criação da solução dos problemas; utilização de processos de ideação e experimentação, por meio de protótipos que permitem a coleta de *feedbacks* e com isso as ideias são amadurecidas; e o mapeamento do maior número de requisitos necessários, permitindo que as expectativas dos usuários sejam atendidas de maneira assertiva.

No processo de desenvolvimento foram encontradas algumas divergências com relação ao processo de integração entre os métodos (ágil e *Design Thinking*). Isso se deve principalmente à característica dos projetos das empresas. Em algumas empresas, seus projetos são maiores ou mais complexos. Essa característica gera a necessidade de uma estrutura mais complexa para suportar tais projetos. Neste caso, se faz necessário *frameworks* que possibilite escalar os métodos ágeis, como é o caso do SAFe.

Já em outra empresa se fez necessária a utilização de processos permitindo que o produto fosse gerado da maneira mais rápida possível e com o menor investimento, devido ao número de incertezas que envolvem o produto que está sendo desenvolvido. Nesse caso, o método *effectuation* também foi aplicado.

Comum a todas as empresas é a utilização de *frameworks* simples, para produção das demandas. O *Scrum* e *XP* são utilizados por todas as empresas estudadas, porém, nem todos os artefatos, eventos ou técnicas são necessários para as essas empresas.

O processo de integração entre os dois modelos, também ocorre de maneira diferente entre as empresas. Neste caso foram encontradas 3 formas de integração: (i) sequencial com *buffer* entre as fases, encontrado na maioria das empresas estudadas. Neste caso as iniciativas tratadas na fase de *design* entram para uma fila (*backlog*) de desenvolvimento que, quando priorizada é *designada* para algum time de desenvolvimento para ser produzida; (ii) sequencial sincronizada, encontrada em uma das empresas. Neste caso, quando a iniciativa é explorada na fase de *design*, já existe um time *designado* para desenvolver a solução, e (iii) o processo de *design* e desenvolvimento acontece junto, encontrado apenas em uma empresa *Startup*.

Como base nestas informações, as proposições: **P1.** Os membros da equipe das áreas de negócio das empresas participam desde a fase inicial do projeto. **P2.** A interação da área de negócio com a área técnica, no grupo que participou do processo do *Design Thinking*, ocorre com frequência para coleta de *feedback*. **P3.** O protótipo, criado no processo de *Design Thinking*, é submetido a *feedbacks* dos usuários finais, antes da integração com os métodos ágeis, e após o desenvolvimento. **P4:** A aplicação do método ágil no desenvolvimento de *software* possui como característica principal o fato de ser adaptável. **P6.** A integração do *Design Thinking* e dos Métodos ágeis ocorre de modo sequencial. Foram consideradas válidas por ser confirmada em mais de 75% das empresas. Com relação a proposição **P5:** O incremento de produto entregue ao final da interação ou *Sprint*, é um incremento viável para implantação e utilização dos usuários. Porém, não foi considerada válida pois não demonstra relevância, uma vez que nem 25% das empresas possuem esta característica.

6.1. LIMITAÇÕES E CONTRIBUIÇÕES PARA ESTUDOS FUTUROS

Dentre algumas das limitações deste estudo, é importante destacar que:

- As empresas estudadas, na maioria dos casos, são empresas complexas e com grande estrutura nas empresas de grande e médio porte. Sendo assim se faz necessário estudar outras empresas *Startups* e empresas de pequeno porte que desenvolvem produtos não tão inovadores. Houve uma dificuldade para encontrar empresas dispostas a colaborar com o estudo, devido à confidencialidade do processo de inovação.
- Dentre os entrevistados, foi possível acessar apenas profissionais que atuam na área de gestão de projetos e metodologia das empresas. Não foram entrevistados profissionais das áreas de negócios nem usuários finais.

Para estudos futuros, sugere-se a aplicação do modelo em uma empresa de forma a validá-lo. Estudos quantitativos de grande amplitude no mercado poderiam também validar os processos de integração, gerando uma teoria substanciada. São sugeridos também estudos específicos para área de inovação de novos produtos diferentes de *software*, nos quais ainda seja necessária a criação de iniciativas para as novas funcionalidades, e não apenas para suprir necessidades com origem nos clientes, como grande parte das demandas atuais de desenvolvimento de *software* encontrados durante o estudo. Também é sugerido que sejam adicionados entrevistados das áreas de negócio, pessoas que contratam e recebem os produtos criados pelo processo e destes, sejam colhidas informações de satisfação e percepção de melhoria na qualidade dos produtos e aplicabilidade.

REFERÊNCIAS

- Abrahamsson, P., & Koskela, J. (2004). Extreme Programming: A Survey of Empirical Data from a Controlled Case Study. *Empirical Software Engineering*, 73–82. <http://doi.org/10.1109/ISESE.2004.19>
- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods. *Vtt Publications*, 478, 167–168. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.161.5931&rep=rep1&type=pdf>
- Abrahamsson, P., Warsta, J., Siponen, M. T. M. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. *Software Engineering, 2003. Proceedings. 25 International*, 6, 244–254. <http://doi.org/10.1109/ICSE.2003.1201204>
- Alt, L., & Pinheiro, T. (2011). *Design Thinking Brasil: empatia, colaboração e experimentação para pessoas, negócios e sociedade (1ª)*. Rio de Janeiro: Elsevier.
- Baregheh, A., Rowley, J., & Sambrook, S. (2009). Towards a multidisciplinary definition of innovation. *Management Decision*, 47(8), 1323–1339. <http://doi.org/10.1108/00251740910984578>
- Beck, K. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Professional.
- Boehm, B., & Turner, R. (2003). Observations on balancing discipline and agility. *Proceedings of the Agile Development Conference, 2003. ADC 2003*. <http://doi.org/10.1109/ADC.2003.1231450>
- Boehm, B., & Turner, R. (2005). Management Challenges to implementing agile processes in traditional software development organizations. *IEEE Software*, 22(5), 30–39.
- Boland, R. J., & Collopy, F. (2004). Design Matters for Management. *Managing as*

Designing.

- Boni, V., & Quaresma, J. (2005). Aprendendo a entrevistar: como fazer entrevistas em Ciências Sociais, 2(3), 68–80.
- Bonini, L. A., & Sbragia, R. (2011). O Modelo de *Design Thinking* como Indutor da Inovação nas Empresas: Um estudo Empírico. *Revista de Gestão E Projetos - GeP*, 2(1), 3–25. <http://doi.org/10.5585/gep.v2i1.36>
- Brown, T. (2008). *Design Thinking*. *Harvard Business Review*, 86(6), 29–32. <http://doi.org/10.1145/2535915>
- Brown, T., & Katz, B. (2011). Change by design. *Journal of Product Innovation Management*, 28(3), 381–383. <http://doi.org/10.1111/j.1540-5885.2011.00806.x>
- Bryman, A., & Bell, E. (2015). *Business research methods* (4th Editio). Oxford University Press.
- Cahyadi, A., & Prananto, A. (2015). Reflecting *Design Thinking*: A case study of the process of designing dashboards. *Journal of Systems and Information Technology*, 17(3), 296–306. <http://doi.org/10.1108/JSIT-03-2015-0018>
- Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4), 332–343. <http://doi.org/http://dx.doi.org/10.1057/ejis.2009.26>
- Carvalho, B. V. de, & Mello, C. H. P. (2012). Aplicação do método ágil scrum no desenvolvimento de produtos de *software* em uma pequena empresa de base tecnológica. *Gestão & Produção*, 19(3), 557–573.
- Clune, S. J., & Lockrey, S. (2014). Developing environmental sustainability strategies, the Double Diamond method of LCA and *Design Thinking*: a case study from aged care. *Journal of Cleaner Production*, 85, 67–82. <http://doi.org/10.1016/j.jclepro.2014.02.003>
- Cohn, M. (2005). *Agile estimating and planning*. Pearson Education.

- Council, D. (2005). The “double diamond” *design* process model. *Design Council*.
- Creswell, J. W. (2014). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (4^a Edition). Los Angeles: Sage Publications, Inc.
- Cross, N. (2001a). *Design / Science / Research : Developing a Discipline. Proceedings of the Fifth Asian Design Conference: International Symposium on Design Science*. Retrieved from http://web.science.mq.edu.au/~isvr/Documents/pdf_files/presence-design-philosophy/DesignScienceResearch.pdf
- Cross, N. (2001b). *Designerly ways of knowing: design discipline versus design science. The Open University, 17(3)*, 49–55. <http://doi.org/10.1162/074793601750357196>
- Dorst, K., & Cross, N. (2001). Creativity in the *design* process: Co-evolution of problem-solution. *Design Studies, 22(5)*, 425–437. [http://doi.org/10.1016/S0142-694X\(01\)00009-6](http://doi.org/10.1016/S0142-694X(01)00009-6)
- Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: current practices, trends, and recommendations. *MIS Quarterly, 597–636*.
- Dybå, T., & Dingsøyra, T. (2008). Empirical studies of agile *software* development: A systematic review. *Information and Software Technology, 50(9–10)*, 833–859. <http://doi.org/10.1016/j.infsof.2008.01.006>
- Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *Management Review, 14(4)*, 532–550. <http://doi.org/10.5465/AMR.1989.4308385>
- Flynn, M., Dooley, L., O’Sullivan, D., & Cormican, K. (2003). Idea management for organizational innovation. *International Journal of Innovation Management, 7(4)*, 417–442.
- Fox, D., Sillito, J., & Maurer, F. (2008). Agile methods and user-centered *design*: How these two methodologies are being successfully integrated in industry. *Proceedings - Agile 2008 Conference, 63–72*. <http://doi.org/10.1109/Agile.2008.78>

- Góes, R., Russo, R., & Felipe, L. (2015). *Design Thinking: Bibliometric Study*. In *13^o CONTECSI - International Conference on Information Systems and Technology Management*. São Paulo, BR: Faculdade de Economia, Administração e Contabilidade da USP.
- Gradilone, C. (2016). O contra-ataque digital dos grandes bancos. Retrieved from <http://www.istoedinheiro.com.br/noticias/mercado-digital/20160926/contra-ataque-digital-dos-grandes-bancos/416192>
- Gurusamy, K., Srinivasaraghavan, N., & Adikari, S. (2016). An Integrated Framework for *Design Thinking* and Agile Methods for Digital Transformation. In A. Marcus (Ed.), *Design, User Experience, and Usability: Design Thinking and Methods: 5th International Conference* (pp. 34–42). Toronto, Canada: Springer International Publishing. <http://doi.org/10.1007/978-3-319-40406-6>
- Hasle, P., Bojesen, A., Jensen, P. L., & Bramming, P. (2012). Lean and the working environment: a review of the literature. *International Journal of Operations & Production Management*, 32(7), 829–849. <http://doi.org/10.1108/01443571211250103>
- Johansson-sköldberg, U., & Woodilla, J. (2013). *Design Thinking* : Past , Present and Possible Futures. *Creativity and Innovation Management*, 22(2), 121–147. <http://doi.org/10.1111/caim.12023>
- Kaur, R., & Sengupta, J. (2011). *Software Process Models and Analysis on Failure of Software Development Projects*. *International Journal of Scientific & Engineering Research*, 2(2), 1–4. Retrieved from <http://arxiv.org/abs/1306.1068>
- Kautz, K., Johanson, T. H., & Uldahl, A. (2014). The Perceived Impact of the Agile Development and Project Management Method Scrum on Information Systems and Software Development Productivity. *Australasian Journal of Information Systems*, 18(3), 303–315. <http://doi.org/http://dx.doi.org/10.3127/ajis.v18i3.1095>

- Kowark, T., Häger, F., Gehrer, R., & Krüger, J. (2013). A Research Plan for the Integration of *Design Thinking* with Large Scale *Software* Development Projects. In L. Leifer, H. Plattner, & C. Meinel (Eds.), *Design Thinking Research* (pp. 183–202). Springer International Publishing. http://doi.org/10.1007/978-3-319-01303-9_12
- Kruchten, P. (2013). Contextualizing agile *software* development. *Journal Of Software: Evolution and Process*, 25, 351–361. <http://doi.org/10.1002/smr>
- Lindberg, T., Meinel, C., & Wagner, R. (2011). *Design Thinking: A Fruitful Concept for IT Development?* In C. Meinel, L. Leifer, & H. Plattner (Eds.), *Design Thinking* (pp. 3–18). Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-642-13757-0_1
- Manifesto for Agile *Software* Development. (2015). Retrieved January 1, 2015, from <http://www.agilemanifesto.org/>
- Martin, R. L. (2007). *Design* and business: why can we be friends? *Journal of Business Strategy*, 28(4), 6–12. <http://doi.org/10.1108/02756660710760890>
- Martin, R. L. (2009). *The design of business: why Design Thinking is the next competitive advantage*. Boston, Massachusetts: Harvard Business Press.
- Martins, H. H. T. de S. (2004). Metodologia qualitativa de pesquisa. *Educação E Pesquisa*, 30(2), 289–300.
- Michels, E., & Ferreira, M. G. G. (2013). Gerenciamento Ágil no Processo de Desenvolvimento de Produtos Inovadores: Uma Análise Bibliográfica Sistemática. *Revista de Gestão E Projetos - GeP*, 4(1), 52–76. <http://doi.org/10.5585/gep.v4i1.119>
- Oliveira, M., Maçada, A. C. G., & Goldoni, V. (2009). Forças e fraquezas na aplicação do estudo de caso na área de sistemas de informação. *Revista de Gestão USP*, 16(1), 33–49.
- Paula, D. F. O. De, & Ara, C. C. (2016). Pet Empires: Combining *Design Thinking*, *Lean Startup* and Agile to Learn from Failure and Develop a Successful Game in an Undergraduate Environment. In *HCI International 2016 -- Posters' Extended Abstracts:*

- 18th International Conference, HCI International 2016* (Vol. 617, pp. 30–34). Toronto, Canada: Springer International Publishing. <http://doi.org/10.1007/978-3-319-40548-3>
- Perminova, O., Gustafsson, M., & Wikström, K. (2008). Defining uncertainty in projects - a new perspective. *International Journal of Project Management*, 26(1), 73–79. <http://doi.org/10.1016/j.ijproman.2007.08.005>
- Plattner, H. (2013). An introduction to *Design Thinking*. *Institute of Design at Stanford*, 1–15. http://doi.org/10.1007/978-1-4302-6182-7_1
- PMBOK. (2013). *PMBOK - Um guia do conhecimento em gerenciamento de projetos* (5ª Edição). Newton Square (USA): PMI - Project Management Institute.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
- Prodanov, C. C., & Freitas, E. C. de. (2013). *Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico* (2ª Edição). Editora Feevale.
- Ratcliffe, J. (2009). Steps in a *Design Thinking* Process. Retrieved January 1, 2017, from <https://dschool-old.stanford.edu/groups/k12/wiki/17cff/>
- Ries, E. (2012). *A Startup enxuta: Como os empreendedores atuais utilizam a inovação contínua para criar empresas extremamente bem-sucedidas*. São Paulo, BR: Lua de Papel.
- Royce, W. W. (1970). Managing the development of large *software* systems. *Electronics*, 26(August), 1–9. [http://doi.org/10.1016/0378-4754\(91\)90107-E](http://doi.org/10.1016/0378-4754(91)90107-E)
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in *software* engineering. *Empirical Software Engineering*, 14. <http://doi.org/10.1007/s10664-008-9102-8>
- Sarasvathy, S. D. (2001). Causation and Effectuation: Toward a Theoretical Shift From Economic Inevitability to Entrepreneurial Contingency, 26(2), 243–263.

- Schwaber, K., & Sutherland, J. (2013). Guia do SCRUM. *Harvard Business Review*, Boston, IV, 163–179. Retrieved from https://www.scrum.org/Portals/0/Documents/Scrum Guides/Scrum_Guide.pdf
- Schwaber, K., & Sutherland, J. (2016). *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*. (Scrum.Org & Scrum Inc, Eds.). Creative Commons.
- Serrador, P., & Pinto, J. K. (2015). Does Agile work? — A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <http://doi.org/10.1016/j.ijproman.2015.01.006>
- The Standish Group. (2011). CHAOS Manifesto 2011. Retrieved June 11, 2011, from http://standishgroup.com/newsroom/chaos%5C_
- Thummadi, B. V., Shiv, O., Berente, N., & Lytinen, K. (2011). Enacted *software* development routines based on waterfall and agile *software* methods: Socio-technical event sequence study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6629 LNCS, 207–222. http://doi.org/10.1007/978-3-642-20633-7_15
- Westerski, A., Iglesias, C. A., & Nagle, T. (2011). The road from community ideas to organisational innovation: a life cycle survey of idea management systems. *International Journal of Web Based Communities*, 7(4), 493–506.
- Wikipédia. (2017). Lista dos maiores bancos do Brasil. Retrieved January 1, 2017, from https://pt.wikipedia.org/wiki/Lista_dos_maiores_bancos_do_Brasil
- Yin, R. K. (2008). The Case Study Crisis : Some Answers. *Administrative Science Quarterly*, 26(1), 58–65.
- Yin, R. K. (2015). *Estudo de Caso: Planejamento e Métodos* (5ª Edição). Porto Alegre – RS: Bookman editora.

APÊNDICE I – PROTOCOLO DE PESQUISA

Parte I – identificação do entrevistado e seu perfil

	Número	Questão
Perfil do entrevistado	Q0	Número de identificação do entrevistado (Controle Interno)
	Q1	Você autoriza a gravação desta entrevista e também autoriza que estas informações sejam utilizadas para fins acadêmicos? Esta gravação não será divulgada, apenas utilizada neste estudo.
	Q2	Qual é a área principal da empresa do entrevistado?
	Q3	Qual a principal atividade ou função do entrevistado?
	Q4	Qual é o seu papel? Conte-me sobre os níveis hierárquicos da sua empresa e qual é o seu nível de decisão (aprovar participar, informar ou executar)
	Q5	Quantos anos de experiência você tem na área de TI?
	Q6	Quantos anos de experiência você tem na gestão de projetos?
	Q7	Quando anos de experiência na área de inovação e desenvolvimento de soluções em TI?

Parte II – Métodos Ágeis

	Número	Questão
Métodos Ágeis	Q8	Quais técnicas e <i>frameworks</i> são utilizados pela empresa?
	Q9	O conceito de <i>Sprint</i> é utilizado pela empresa?
	Q10	O incremento gerado pela <i>Sprint</i> é utilizável?
	Q11	Como é definido o <i>Product Backlog</i> ?
	Q12	Como é definido o <i>Sprint Backlog</i> ?
	Q13	Quem participa da priorização do <i>Product Backlog</i> ?
	Q14	Como é feita a priorização do <i>Product Backlog</i> ?
	Q15	Eventuais mudanças não aceitas na reunião de revisão da <i>Sprint</i> entram como novo item de <i>backlog</i> ?
	Q16	O princípio de desenvolvimento incremental e adaptativo do desenvolvimento proposto pelos métodos ágeis, geram desperdícios com retrabalho?
	Q17	O princípio de desenvolvimento incremental e adaptativo do desenvolvimento proposto pelos métodos ágeis, geram problemas de arquitetura de <i>software</i> ?

Parte III – *Design Thinking*

	Número	Questão
<i>Design Thinking</i>	Q18	Existe interação contínua entre os membros das equipes?
	Q19	Quem participa de cada uma das fases do <i>Design Thinking</i> , desde a fase inicial da concepção do projeto?
	Q20	Como ocorre a interação dos membros da equipe no processo do <i>Design Thinking</i> ?
	Q21	O protótipo criado neste processo é submetido a <i>feedbacks</i> com pessoas que não participaram do <i>Design Thinking</i> ?
	Q22	Qual a frequência que ocorrem as sessões de <i>Design Thinking</i> ?
	Q23	Qual a duração de cada sessão de <i>Design Thinking</i> ?
	Q24	Quais ferramentas e artefatos são utilizados nas sessões de <i>Design Thinking</i> ?

Parte IV – Integração entre os Métodos Ágeis e o *Design Thinking*

Integração Métodos Ágeis e <i>Design Thinking</i>	Numero	Questão
	Q25	Em que momento é realizado o processo de <i>Design Thinking</i> ?
	Q26	Como ocorre a transição entre o <i>design</i> e a construção da solução?
	Q27	Como é feito o sincronismo entre os eventos dos métodos ágeis com as fases do <i>Design Thinking</i> ?
	Q28	Como são coletados os <i>feedbacks</i> dos usuários finais?
	Q29	A utilização dos métodos, quando usados isoladamente, geram produtos que não aderem às expectativas dos usuários?
	Q30	Como são priorizados e gerenciados os itens gerados pela etapa de <i>design</i> para construção?
	Q31	Como é feita a entrega final para o cliente?
	Q32	Como é medida a satisfação do cliente?