

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANA CAROLINA COLNAGO

**METODOLOGIAS ÁGEIS NA GESTÃO DE PROJETOS: UM ESTUDO DE CASO
EM EMPRESAS DE SOFTWARE DE LONDRINA**

LONDRINA

2022

ANA CAROLINA COLNAGO

**METODOLOGIAS ÁGEIS NA GESTÃO DE PROJETOS: UM ESTUDO DE CASO
EM EMPRESAS DE SOFTWARE DE LONDRINA**

**Agile Methodologies In Project Management: A Case Study In Software
Companies In The City Of Londrina**

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Engenharia de Produção da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Dr. Fabiano Palhares Galão.

LONDRINA

2022



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ANA CAROLINA COLNAGO

**METODOLOGIAS ÁGEIS NA GESTÃO DE PROJETOS: UM ESTUDO DE CASO
EM EMPRESAS DE SOFTWARE DE LONDRINA**

Trabalho de conclusão de curso de graduação
apresentada como requisito para obtenção do título
de Bacharel em Engenharia de Produção da
Universidade Tecnológica Federal do Paraná
(UTFPR).

Orientador(a): Dr. Fabiano Palhares Galão.

Data de aprovação: 07 de maio de 2022

Fabiano Palhares Galão
Doutor
Universidade Tecnológica Federal do Paraná

Marco Antonio Ferreira
Doutor
Universidade Tecnológica Federal do Paraná

Regina Lucia Sanches Malassise
Doutora
Universidade Tecnológica Federal do Paraná

LONDRINA

2022

Dedico este trabalho aos meus pais e aos amigos mais próximos pela força e paciência durante meus anos de graduação.

AGRADECIMENTOS

Gostaria de começar agradecendo aos meus pais, que mesmo distante sempre demonstraram apoio e confiança nas minhas decisões. Obrigada por serem quem são!

Quero agradecer também aos poucos amigos que me acompanharam nessa jornada e que sempre me incentivaram a continuar mesmo nos momentos mais difíceis, permanecendo ao meu lado quando mais precisei. Sem vocês minha caminhada teria sido mais espinhosa e menos divertida, por isso, agradeço!

Também gostaria de agradecer ao meu orientador, Prof. Dr. Fabiano Palhares Galão, pela paciência com os meus avanços (ou falta deles) e pelas boas ideias empregadas ao longo do meu trabalho para que chegássemos até aqui.

Por fim, quero deixar meu agradecimento a todos os professores que cruzaram o meu caminho durante esses anos de graduação, obrigada pelos ensinamentos tanto acadêmicos quanto para a vida. Levarei com carinho cada um de vocês para onde quer que eu vá.

RESUMO

Os softwares estão sendo cada vez mais utilizados e requisitados em todas as esferas da sociedade e a adaptação das empresas do ramo a um ambiente dinâmico e mutável é necessária para gerar valor agregado aos sistemas desenvolvidos. Para tanto as empresas utilizam de diferentes metodologias para fazer com que o desenvolvimento de seus produtos seja assertivo e atenda as expectativas dos clientes. Dessa forma, esse trabalho tem como objetivo identificar quais características ágeis as empresas de software de Londrina possuem seguindo os princípios do Manifesto Ágil, criado para prover conceitos e princípios para o desenvolvimento desses produtos, bem como a utilização ou não outros métodos considerados ágeis como *Scrum*, XP, dentre outros, no intuito de observar se há a aplicação de alguma dessas metodologias e o que as empresas entendem sobre o assunto. A pesquisa se justifica pelo fato de não haver estudos semelhantes na cidade de Londrina nesse segmento. O método utilizado foi o de uma entrevista em profundidade com os representantes das 4 empresas que aceitaram participar da pesquisa, com um roteiro de perguntas pré-definido para que a entrevista fosse guiada para obtenção das informações necessárias. Os resultados obtidos variaram de acordo com o porte da empresa, mas o que se pôde notar é que todas as empresas entrevistadas entendiam sobre o assunto, mas que a aplicabilidade no dia a dia fica restrita a realidade de cada uma. Características de métodos como *Scrum*, XP, *Kanban* e FDD foram percebidos, com destaque para o *Scrum* pela maior facilidade de adaptação cultural dos colaboradores.

Palavras-chave: desenvolvimento; manifesto, métodos ágeis; software.

ABSTRACT

Software is being increasingly used and required in all spheres of society and the adaptation of companies in the field to a dynamic and changing environment is necessary to generate added value to the developed systems. For this, companies use different methodologies to make the development of their products assertive and meet customer expectations. In this way, this work aims to identify which agile characteristics the software companies in Londrina have following the principles of the Agile Manifesto, created to provide concepts and principles for the development of these products, as well as the use or not of other methods considered agile such as *Scrum*, *XP*, among others, in order to observe if there is the application of any of these methodologies and what companies understand about the subject. The research is justified by the fact that there are no similar studies in the city of Londrina in this segment. The method used was an in-depth interview with representatives of the 4 companies that agreed to participate in the research, with a pre-defined script of questions so that the interview was guided to obtain the necessary information. The results obtained varied according to the size of the company, but what could be noticed is that all the companies interviewed understood the subject, but that the applicability in everyday life is restricted to the reality of each one. Characteristics of methods such as *Scrum*, *XP*, *Kanban* and *FDD* were noticed, with emphasis on *Scrum* due to the greater ease of cultural adaptation of the collaborators.

Keywords: software; development; manifest; agile methods.

Lista de Tabelas

Tabela 1 – Resumo Métodos Tradicionais para Desenvolvimento de Software.....	24
Tabela 2 - Resumo Métodos Ágeis para Desenvolvimento de Software.....	38
Tabela 3- Empresas participantes por porte.....	43
Tabela 4 - Resumo dos resultados obtidos	70

Lista de Ilustrações

Figura 1- Sequência de atividades do modelo cascata	17
Figura 2 - Modelo de Prototipação Evolucionária	20
Figura 3 - Ciclo de Vida Espiral	22
Figura 4 - Ciclo de Vida XP	29
Figura 5 - Ciclo de Vida no OpenUp.....	34
Figura 6 - Atividades exercidas pelas indústrias de software de Londrina e Região metropolitana.....	41
Figura 7 - Classificação pelo porte da empresa	42

Sumário

1. INTRODUÇÃO	13
2. REFERENCIAL TEÓRICO.....	16
2.1 Métodos Tradicionais para Desenvolvimento de Software.....	16
2.1.1 Modelo Codificar e Consertar.....	16
2.1.2 Modelo Cascata	16
2.1.3 Modelo Incremental.....	18
2.1.4 Prototipação Evolucionária.....	20
2.1.5 Modelo Espiral	21
2.2 Metodologias Ágeis para Desenvolvimento de Software.....	24
2.2.1 Breve histórico e Manifesto Ágil	24
2.2.2 Extreme Programming (XP)	27
2.2.3 Scrum.....	31
2.2.4 OpenUp.....	33
2.2.5 Feature-Driven Development (FDD).....	35
2.2.5 Dynamic Systems Development Method (DSDM).....	36
3. MÉTODOS E TÉCNICAS DE PESQUISA	40
3.1 Natureza, Objetivos e Método da pesquisa	40
3.2 Etapas Metodológicas	41
3.2.1 Público-alvo da pesquisa e Fator de Decisão	41
3.2.2 Amostra.....	42
3.2.3 Método de coleta de dados	43
3.2.4 Análise dos dados.....	44
4. RESULTADOS DAS ENTREVISTAS	46
4.1 Empresa A	46
4.1.1 Início do Projeto e Documentação	46
4.1.2 Participação do Cliente	48
4.1.3 Equipe, Desenvolvimento e Entrega	49
4.1.4 Métodos Ágeis	50
4.2 Empresa B	52

4.2.1	Início do Projeto e Documentação	53
4.2.2	Participação do Cliente	54
4.2.3	Equipe, Desenvolvimento e Entrega	55
4.2.4	Métodos Ágeis	57
4.3	Empresa C	58
4.3.1	Início do Projeto e Documentação	59
4.3.2	Participação do Cliente	60
4.3.3	Equipe, Desenvolvimento e Entrega	61
4.3.4	Métodos Ágeis	62
4.4	Empresa D	63
4.4.1	Início do Projeto e Documentação	63
4.4.2	Participação do Cliente	66
4.4.3	Equipe, Desenvolvimento e Entrega	67
4.4.4	Métodos Ágeis	69
4.5	Compilado dos dados das empresas estudadas	70
5.	ANÁLISE DOS RESULTADOS	73
5.1	Empresa A	73
5.1.1	Início do Projeto e Documentação	73
5.1.2	Participação do Cliente	74
5.1.3	Equipe, Desenvolvimento e Entrega	75
5.1.4	Métodos Ágeis	77
5.2	Empresa B	79
5.2.1	Início do Projeto e Documentação	79
5.2.2	Participação do Cliente	80
5.2.3	Equipe, Desenvolvimento e Entrega	80
5.2.4	Métodos Ágeis	82
5.3	Empresa C	82
5.3.1	Início do Projeto e Documentação	82
5.3.2	Participação do Cliente	83
5.3.3	Equipe, Desenvolvimento e Entrega	84
5.3.4	Métodos Ágeis	86

5.4 Empresa D	86
5.4.1 Início do Projeto e Documentação	86
5.4.2 Participação do Cliente	87
5.4.3 Equipe, Desenvolvimento e Entrega	88
5.4.4 Métodos Ágeis	90
6. CONCLUSÃO	92
7. REFERÊNCIAS.....	94
APÊNDICE A – Questionamentos para Entrevista com Responsável pela Empresa	95

1. INTRODUÇÃO

Não é novidade que há muitos anos o mundo vem passando por uma revolução tecnológica. Em meio aos diferentes tipos de tecnologias utilizadas no cotidiano os softwares têm destaque por seus diferentes tipos de aplicações, seja para aplicativos voltados para entretenimento ou dentro de uma organização que os utiliza para gerir seu negócio como um todo, desde a gestão de estoques até prever futuras demandas. Segundo Prikladnick *et. al* (2014, p.19), “O software está presente em inúmeras atividades desde as mais simples, como processar informações básicas de compra e venda, até atividades mais complexas, como controlar o carro que dirigimos ou o avião que voa por dez horas sem escala.” Neste sentido, o autor ainda cita que com a evolução da indústria de software há a possibilidade dela se tornar umas das mais importantes da era moderna.

Em se tratando do desenvolvimento de softwares, de acordo com Sommerville (2011), a sequência de atividades que devem ser realizadas para a produção de um software são chamadas de processo de software, sendo que quatro desses processos são fundamentais e comuns a todos os projetos: especificação, desenvolvimento, validação e evolução do software. No início dessa indústria, alguns métodos eram amplamente utilizados para auxiliar no desenvolvimento dos projetos de software.

As chamadas metodologias tradicionais ou clássicas para o desenvolvimento de software surgiram em um contexto muito diferente do atual. Segundo Gutierrez (2004), na época fazer alterações durante o desenvolvimento do projeto era muito custoso, pois não havia muitas ferramentas de apoio e até mesmo ter acesso a computadores era uma tarefa complicada. O principal modelo utilizado era o modelo Cascata (do inglês *waterfall*), sendo um modelo em que há uma sequência de atividades a serem realizadas e cada uma das etapas só poderia ser iniciada se a fase anterior tivesse sido devidamente encerrada e documentada. Apesar da inflexibilidade e da dificuldade em fazer alterações durante o projeto, o modelo Cascata comandou a forma de desenvolvimento de software até o início da década de 1990 sendo utilizado até os dias atuais.

No entanto, dados da The Standish Group de 1994 que utilizaram como base 8380 projetos, mostraram que apenas 16,2% dos projetos foram entregues dentro dos prazos e custos estipulados e com as funcionalidades especificadas. O relatório também apontou que 31% dos projetos foram cancelados antes de estarem

completos e que 52,7% foram entregues, mas com custos e prazos maiores e sem todas as funcionalidades requeridas pelo cliente.

Segundo Beck (1999), as falhas nas entregas e valores no desenvolvimento de software tem um impacto econômico e humano muito grande e para que esse problema seja solucionado é necessário encontrar uma maneira de desenvolver softwares com qualidade e entregas frequentes. Nesse sentido o autor ainda cita que para solucionar esses problemas seria adequada a utilização de métodos e práticas ágeis, capazes de se adaptar a uma realidade mutável oferecendo respostas rápidas a qualquer tipo de mudança e que ainda melhore a qualidade das entregas.

Assim, no início dos anos 2000, um grupo de profissionais da área de software se reuniu para selecionar e unir uma série de práticas ágeis que tornariam o desenvolvimento de software mais flexível e adaptativo, que posteriormente seriam chamadas de metodologias ágeis. Nesse encontro foi estabelecido o Manifesto Ágil que segundo Prikladnick *et. al* (2014, p.20) possui como conceitos chave “indivíduos e interações mais que processos e ferramentas, software funcionando mais que documentação abrangente, colaboração com o cliente mais que negociação de contratos e responder a mudanças mais que seguir um plano”

Segundo AbrahCorporateson (2003), as metodologias ágeis constituem uma nova classe de metodologias de desenvolvimento de software criada para atender à crescente pressão do mercado por processos mais ágeis e leves, com ciclos de desenvolvimento cada vez mais curtos.

As empresas de desenvolvimento de software têm um papel importante na economia e na geração de empregos. No Brasil, segundo dados do estudo “Mercado Brasileiro de Software – Panorama e Tendências de 2021” da ABES, mostraram que a indústria de tecnologia do país cresceu 22,9% e conquistou a 9ª posição no ranking mundial de TI em 2020. Além disso, o estudo apontou que nesse mesmo ano o Brasil ainda investiu cerca de U\$49,5 bilhões no ramo de tecnologia.

Como em outras regiões do País a cidade de Londrina conta com um expressivo número de empresas do segmento. Dados da CNAE 2019 apontaram que a cidade contava com 178 empresas do ramo, além das que já foram instaladas depois da realização do estudo. Além disso, Londrina também possui um APL de TIC (Arranjo Produtivo Local de Tecnologia da Informação e Comunicação) que envolve empresas do eixo Londrina, Apucarana e Cornélio Procópio para articulação

e ações em prol do setor de TIC além de apoio, organização e realização de eventos. Outro ponto a se destacar é que, segundo o Guia da Carreira, na cidade existem cerca de 39 cursos de graduação de ensino presencial e remoto que formam centenas de profissionais da área todos os anos.

Apesar da importância e relevância do setor na cidade, são poucos estudos específicos que abordem os processos de desenvolvimento de software por parte das empresas, em especial, sobre as metodologias ágeis e seus benefícios para as empresas.

Diante desse contexto, este trabalho tem por objetivo verificar as práticas e metodologias que as empresas de tecnologia de Londrina utilizam no desenvolvimento de seus softwares, quais características dos modelos ágeis estão sendo utilizados no seu dia a dia e quais semelhanças podem ser encontradas com os princípios descritos no Manifesto Ágil.

Assim, esse trabalho se justifica pelo fato de estudar como as empresas de tecnologia da segunda maior cidade do estado gerenciam e desenvolvem seus projetos no intuito de explorar novas opções para o desenvolvimento de software.

Na próxima seção será apresentado o referencial teórico. Os temas abordados serão os métodos tradicionais para desenvolvimento de software seguidos dos métodos ágeis para o desenvolvimento e suas características.

2. REFERENCIAL TEÓRICO

2.1 Métodos Tradicionais para Desenvolvimento de Software

2.1.1 Modelo Codificar e Consertar

O modelo Codificar e Consertar (do inglês *code and fix*), é um dos modelos mais simples para o desenvolvimento de um software. Wazlawick (2019) resume esse modelo em cinco etapas começando com construir um entendimento do sistema junto ao cliente, passando para implementar uma primeira versão do software para então apresentar a versão preliminar para o cliente novamente encontrando os erros do protótipo e os consertando até chegar em um ponto satisfatório. Após essa etapa de validação, outros testes eram feitos para corrigir outros erros até que finalmente o produto fosse entregue para o cliente.

O autor ainda cita que esse é um processo um tanto ingênuo para desenvolver um software, uma vez que não há previsibilidade relacionada tanto as atividades quanto ao resultado que se espera, mas que apesar disso ainda é um modelo que pode ser usado para projetos que tenham um curto período para serem desenvolvidos ou para a produção de protótipos e testes de conceito sendo descartado logo depois.

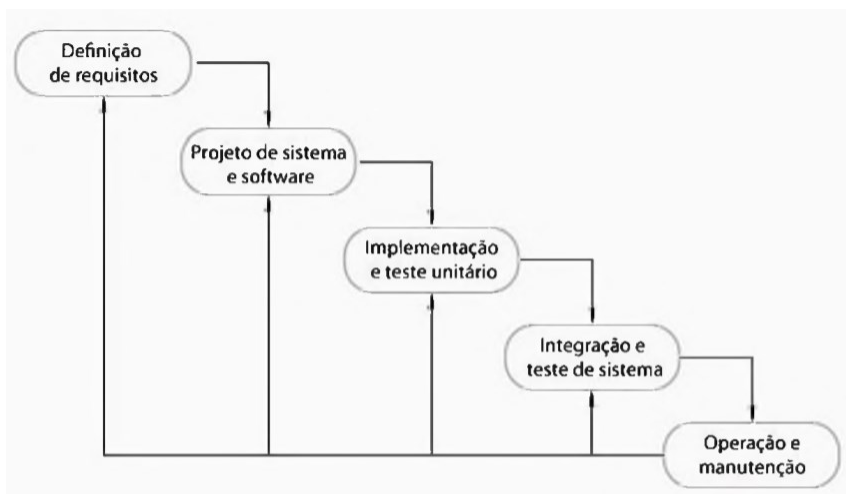
2.1.2 Modelo Cascata

O modelo cascata foi criado por Winston Royce na década de 1970. O autor acreditava que o processo de desenvolvimento deveria ocorrer de forma linear, assim as atividades eram agrupadas em tarefas e realizadas sequencialmente sendo que uma etapa só pode ter início se sua anterior tiver sido devidamente finalizada (Royce, 1970). Ellis (2010), diz que o modelo tem como característica uma revisão ao final de cada fase para avaliação do projeto. Caso essa revisão mostre que o projeto não está pronto para seguir a próxima etapa ele deve continuar na fase em que se encontra.

Sommerville (2011, p.20), diz que “O modelo em cascata é um exemplo de um processo dirigido a planos — em princípio, você deve planejar e programar todas as atividades do processo antes de começar a trabalhar nelas.” Segundo o mesmo autor, o modelo cascata segue as cinco etapas para desenvolvimento de software que consiste na análise e definição dos requisitos, projetar o sistema de software,

implementar, integrar e testar para finalmente chegar a fase de operação e manutenção, como mostra a Figura 1 a seguir.

Figura 1- Sequência de atividades do modelo cascata



Fonte: Engenharia de Software, Sommerville, 2011

Wazlawick (2019), diz que as ideias fundamentais do modelo cascata são coerentes e podem gerar benefícios relevantes já que a existência de fases bem definidas podem ajudar na detecção de erros de forma precoce tornando-o mais barato para ser corrigido. Outro ponto positivo que o autor atribui a esse modelo é que de certa forma ele procurou promover uma estabilidade nos requisitos, dessa forma o projeto só é continuado se os requisitos forem aceitos. Além disso, o autor concorda que esse modelo funciona bem para projetos nos quais os requisitos são conhecidos e estáveis se adequando a equipes inexperientes, pois é um modelo estruturado e sistemático.

Apesar dos pontos positivos citados por Wazlawick, o modelo cascata sofreu diversas críticas ao longo dos anos. Ellis (2010), deu suas considerações sobre os pontos negativos da utilização desse modelo. Para o autor o modelo Cascata não produz reais benefícios até a fase de codificação, exceto para aqueles cujo as técnicas de documentação são familiares. Outro ponto exposto pelo autor é de que não há flexibilidade nos requisitos e que voltar atras para corrigir requisitos mal estabelecidos ou que sofrem mudança ao longo do percurso é bastante custoso e trabalhoso.

Pressman (2016), cita que as críticas a esse modelo fizeram até mesmo seus maiores defensores questionarem sua eficácia. Para o autor, em projetos reais é

complicado usar um fluxo sequencial de tarefas pois qualquer tipo de mudança pode provocar confusões a medida em que a equipe do projeto prossegue. O autor também cita que é difícil para o cliente estabelecer explicitamente todas as necessidades que ele gostaria que o software atendesse e o modelo Cascata tem dificuldades para se adequar as incertezas existentes no início de muitos projetos.

O ritmo de trabalho atual para o desenvolvimento de software é bem acelerado e está sujeito a todo tipo de mudança ao longo do projeto. Apesar das críticas Pressman (2016) reconhece que o modelo cascata ainda que inadequado para esse ritmo de trabalho pode sim ser utilizado em situações nas quais os requisitos são fixos e o trabalho deve ser realizado de forma linear até sua finalização.

2.1.3 Modelo Incremental

Como já visto anteriormente, os processos tradicionais trazem uma concepção linear para o desenvolvimento de um projeto. No entanto Pressman (2016), aborda que em parte das situações apesar dos requisitos iniciais serem bem definidos de certa forma, o escopo do trabalho impede o uso de um processo que seja puramente linear.

Bezerra (2002, p.37) diz que “o modelo de ciclo de vida incremental e iterativo foi proposto como uma resposta aos problemas encontrados no modelo em cascata.” O autor detalha o modelo iterativo usando a ideia dos ciclos. Cada ciclo possui um subconjunto de requisitos que serão desenvolvidos uma vez que “alocados” naquele ciclo de desenvolvimento. No próximo ciclo, outro subconjunto de requisitos será considerado no desenvolvimento, introduzindo um novo incremento no sistema que irá conter extensões e refinamentos sobre o incremento anterior. Os ciclos consistem nas fases de análise, projeto, implementação e testes. Essas fases são realizadas em cada ciclo necessário para o desenvolvimento daquele software, o que se diferencia do modelo cascata onde essas etapas eram realizadas uma única vez.

Pressman (2016) exemplifica o modelo incremental utilizando como exemplo a produção de um software de textos. Nesse exemplo, os requisitos entregues no final do primeiro incremento poderiam ser as funções básicas do gerenciamento de arquivos, edição e produção de documentos que no segundo incremento sofreriam alterações e evoluções mais sofisticadas na parte de edição de produção do

documento. No terceiro incremento seriam inclusas as funções de revisão ortográfica e gramatical para que finalmente no quarto incremento os recursos avançados de *layout* de página fossem inclusos.

Sommerville (2011), cita três vantagens importantes que o modelo incremental tem sobre o modelo cascata. A primeira delas é que o custo de fazer uma modificação nos requisitos é menor nesse modelo, pois a documentação e concepção não é feita de uma única vez como sugere o modelo cascata. O modelo interativo avança com as funcionalidades do software conforme os requisitos que necessitam ser incrementados no decorrer dos ciclos a serem realizados até a entrega final do produto, o que o torna mais adaptável a possíveis mudanças. A segunda vantagem exposta pelo autor é que no modelo interativo é mais fácil conseguir *feedback* dos clientes sobre as etapas do projeto que já foram concluídas. Assim, os clientes conseguem fazer comentários sobre as demonstrações do software e ver o quanto foi implementado até aquele momento. No modelo cascata isso era um pouco mais complicado, uma vez que é difícil para o cliente avaliar a evolução por meio de documentos do projeto do software. Por fim, a última vantagem exposta por Sommerville (2011, p.22) é que “é possível obter entrega e implementação rápida de um software útil ao cliente, mesmo se toda a funcionalidade não for incluída. Os clientes podem usar e obter ganhos a partir do software inicial antes do que é possível com um processo em cascata.”

As dificuldades encontradas na utilização desse modelo tanto para Sommerville (2011), quanto para Bezerra (2002), são relacionadas ao gerenciamento dos projetos que utilizam essa abordagem. Os dois autores citam que para o gerente de projetos é bem mais difícil ter que lidar com fases de análise, projeto, testes e implementação ocorrendo paralelamente, isso acaba deixando o processo menos visível, o que para um gerente não é a melhor opção visto que para eles entregas regulares se fazem necessárias para mensurar o progresso que o projeto está tendo. Outro ponto levantado por Sommerville (2011, p.22), traz de volta a questão do custo que constantes mudanças no software podem ter “a menos que tempo e dinheiro sejam dispendidos em refatoração para melhoria do software, as constantes mudanças tendem a corromper sua estrutura. Incorporar futuras mudanças do software torna-se cada vez mais difícil e oneroso.”

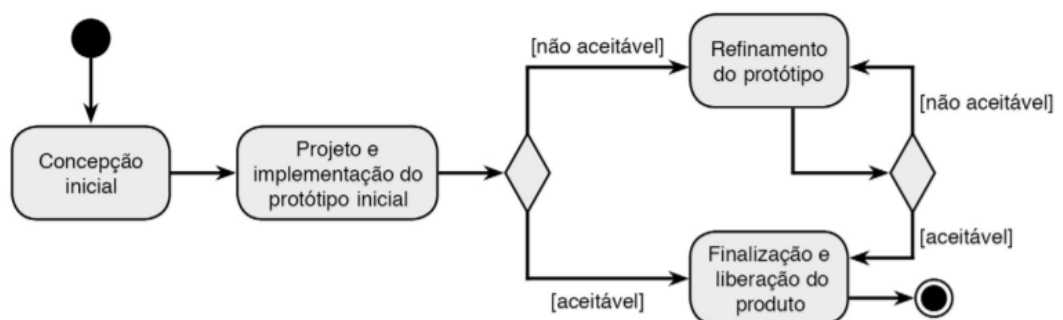
2.1.4 Prototipação Evolucionária

Sommerville (2011, p.30) descreve protótipo como sendo “uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções”. É utilizando protótipos que os clientes conseguem ter visões do suporte que o sistema poderá dar ao seu trabalho, obter novas ideias e enxergar vantagens e desvantagens do software.

Wazlawick na primeira edição de seu livro em 2013, diz que existem basicamente duas abordagens de prototipação. A primeira delas é a descartável, do inglês *throw-away*, em que os protótipos são desenvolvidos e usados unicamente para estudar aspectos do sistema, auxiliar na identificação de riscos e para entender melhor quais requisitos poderiam ser melhorados e após análise o protótipo é descartado. A outra abordagem na qual se baseia o modelo de prototipação evolucionário é a abordagem *cornerstone* (pedra fundamental), que consiste na construção de um protótipo que também pode ser usado para estudar aspectos do sistema, mas que será parte do produto, evoluindo no decorrer das etapas até se tornar um sistema que possa ser entregue ao cliente no final do projeto. Segundo o autor, essa abordagem exige um planejamento do protótipo mais completo do que a abordagem *throw-away*, visto que se os defeitos iniciais não forem consertados serão levados até o sistema final.

Ainda segundo Wazlawick (2013), o modelo de prototipação evolucionária faz com que a equipe de desenvolvimento trabalhe aspectos mais visuais do sistema junto ao cliente apresentando as interfaces criadas e analisando se são aceitáveis ou não. O esquema da Figura 2 a seguir, apresenta um diagrama de atividades que envolve justamente a aprovação ou não do cliente durante a execução do projeto.

Figura 2 - Modelo de Prototipação Evolucionária



Fonte: Engenharia de Software: conceitos e práticas, Wazlawick, 2013.

Pressman (2016), cita que tanto os desenvolvedores quanto clientes geralmente gostam da abordagem da prototipação evolucionária, uma vez que caso o cliente não saiba exatamente qual requisito gostaria no produto final, mas que durante o processo vão adquirindo uma visão de como o sistema está sendo desenvolvido participando ativamente dos estágios do processo, o que também traz segurança para o desenvolvedor que sabe que está trabalhando de acordo com as pontuações feitas pelo cliente evoluindo com o projeto a partir disso.

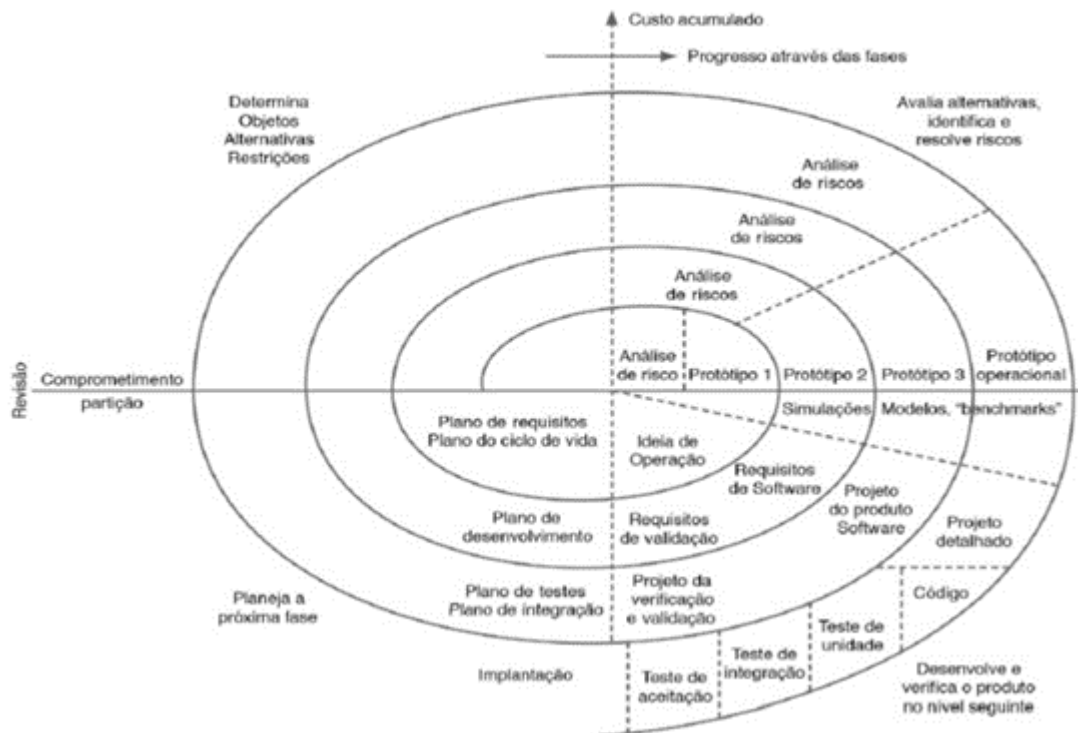
Wazlawick (2013), diz que a desvantagem dessa abordagem é no controle do tempo para o desenvolvimento do produto e em relação ao gerenciamento do projeto, uma vez que fica um pouco difícil dizer quando cada fase foi realmente efetivada. Para mitigar esses problemas, o autor cita que é necessário garantir que o processo “obtenha uma concepção real do sistema, com requisitos definidos da melhor forma possível e um projeto realista antes de iniciar a codificação propriamente dita” (Wazlawick, 2013, p.38).

2.1.5 Modelo Espiral

Proposto por Boehm em 1986, o modelo espiral progrediu durante muitos anos “baseado em experiências com vários refinamentos do modelo cascata como aplicado a grandes projetos de software do governo” (Boehm, 1988, p.4). Almeida (2018), diz que a maior diferença entre o modelo cascata e o modelo espiral é que o segundo tem uma abordagem voltada para a identificação de riscos ao invés de ter sua orientação voltada a documentação ou até mesmo a codificação.

Wazlawick (2019), concorda que o modelo espiral tem uma característica muito voltada a redução de riscos e explica que o projeto como um todo é dividido em subprojetos nos quais há a abordagem de um ou mais elementos fontes de risco até que todas as ameaças tenham sido tratadas. O autor ainda cita que esse modelo prevê iterações mais organizadas que evoluem de pequenos protótipos avançando para níveis cada vez mais complexos, como mostra a Figura 3 a seguir.

Figura 3 - Ciclo de Vida Espiral.



Fonte: Engenharia de Software: conceitos e prática, Wazlawick, 2013

Boehm (1988) explica que cada ciclo do modelo espiral começa com a identificação dos objetivos da parte do produto que está sendo elaborada, como funcionalidades, performance, habilidade de se adaptar a mudanças etc., passando para analisar quais meios alternativos podem ser utilizados para implementar essa parte do produto e quais as restrições que podem surgir da aplicação dessas alternativas como custos, interface e cronograma. Os meios alternativos podem ser basicamente qualquer tipo de modelo de desenvolvimento que se encaixe para a produção da etapa em questão. Assim, no próximo passo se avaliam alternativas que se relacionam aos objetivos e restrições que aquela etapa do projeto possui. Essa avaliação ajuda a identificar quais áreas podem gerar incertezas no desenvolvimento o que reflete diretamente nas fontes de risco que aquele projeto possui. Dessa forma, após a conclusão de cada um dos níveis de desenvolvimento há uma etapa de validação e preparação para a próxima etapa a ser executada avaliando quais riscos podem ser identificados e, como em outros modelos, há uma revisão por parte dos envolvidos no desenvolvimento daquele produto, como o cliente e partes da gestão da empresa. Essa revisão inclui os planos propostos para a fase seguinte bem como toda a análise da parte já concluída.

Apesar de ser um modelo que consegue se adaptar a quase todo tipo de desenvolvimento, o próprio proponente do modelo, Boehm (1988), enxerga uma desvantagem em sua utilização. O autor aponta que em um grupo de desenvolvedores inexperientes, a identificação dos riscos pode ser feita de forma incompleta. Boehm explica que existem dois tipos de identificação de riscos com diferentes tipos de detalhamento, aquela em que se avalia e se detalha todos os parâmetros até encontrar todos os fatores de risco para o projeto e a identificação feita de forma superficial enxergando apenas os elementos de alto risco durante as etapas de desenvolvimento. Caso a equipe seja inexperiente, é provável de que a análise dos riscos seja feita de forma rasa, identificando os elementos que realmente são de alto risco para o projeto, o que de certa forma ainda é positivo, mas levando em conta que uma das premissas do modelo espiral é a identificação de todos os fatores que podem ser ameaças ao desenvolvimento, esse ponto pode ser insuficiente.

Apesar da pequena desvantagem citada, Pressman (2016) vê o modelo espiral como “uma abordagem realista para o desenvolvimento de software” (2016, p.49). Para o autor como o modelo espiral utiliza a prototipação como mecanismo para redução de danos, torna-se possível aplicá-la em qualquer estágio do desenvolvimento do produto mantendo assim algumas características sistemáticas do modelo cascata, mas também incorporando uma metodologia iterativa que reflete de forma mais realista o mundo em o desenvolvimento daquele produto está inserido.

Para sintetizar os conceitos, a Tabela 1 a seguir traz as principais características dos Métodos Tradicionais para desenvolvimento de software mencionados.

Tabela 1 – Resumo Métodos Tradicionais para Desenvolvimento de Software

Método	Descrição
Code and Fix	<ul style="list-style-type: none"> • Lento; • Protótipos desenvolvidos sem visão clara de resultados.
Cascata	<ul style="list-style-type: none"> • Lento; • Orientado a documentação; • Mais bem aplicado em projetos com requisitos conhecidos e estáveis; • Modificações ao longo do projeto costuma ser alto; • Pouco <i>feedback</i> dos clientes durante a execução.
Incremental	<ul style="list-style-type: none"> • Utiliza a ideia de ciclos que incrementam requisitos de acordo com a evolução; • Modificações com custo menor; • <i>Feedbacks</i> dos clientes facilitado.
Prototipação Evolucionária	<ul style="list-style-type: none"> • Protótipos feitos e apresentados para o cliente de forma recorrente; • <i>Feedbacks</i> do cliente recebidos continuamente; • Cliente acrescenta requisitos conforme os protótipos são construídos.
Espiral	<ul style="list-style-type: none"> • Voltado para identificação de riscos; • Utiliza a prototipação para identificação de problemas; • Validações em cada etapa concluída e nova avaliação dos riscos da fase subsequente.

2.2 Metodologias Ágeis para Desenvolvimento de Software

2.2.1 Breve histórico e Manifesto Ágil

As metodologias ágeis surgiram em oposição aos métodos tradicionais, também chamados métodos pesados que, como visto, eram lentos e orientados a documentação. No entanto, essas metodologias só passaram a se chamar Metodologias Ágeis depois de um encontro ocorrido em 2001 com um grupo de dezessete especialistas da área de software que se reuniram em uma estação de ski, em Utah, nos Estados Unidos.

Segundo Gomes *et. al* (2014), o objetivo dessa reunião foi discutir o desenvolvimento de software de forma mais leve, rápida e centrada em pessoas. O autor ainda cita que nessa reunião os termos “Desenvolvimento Ágil de Software” e “Métodos Ágeis” foram criados, bem como o chamado “Manifesto Ágil”, composto pelos princípios e valores que deveriam ser utilizados para o desenvolvimento de software.

O Manifesto Ágil foi publicado em 2001 e conta com quatro valores e doze princípios que serviriam como guias para a produção dos softwares. Segundo Beck *et. al* (2001), os valores do manifesto são:

- **Indivíduos e interação** mais que processos e ferramentas;
- **Software em funcionamento** mais que documentação abrangente;
- **Colaboração com o cliente** mais que negociação de contratos;
- **Responder a mudanças** mais que seguir um plano

Gomes *et. al* (2014), comenta cada um dos valores e o que está “por trás” deles. O autor cita que com o passar do tempo passou-se a seguir cegamente os processos e ferramentas utilizadas para desenvolver o software e que o mais importante acabou ficando de lado: a comunicação. Afinal, as ferramentas e processos são sim de suma importância para a produção de um software, mas são as pessoas que fazem acontecer. É importante se lembrar e valorizar isso, pois na maioria dos casos os softwares são desenvolvidos por equipes que precisam da comunicação como guia para o trabalho. Com relação à documentação, o autor diz que o manifesto não nega a importância dessa etapa, mas que é mais importante ter um software que funcione do que um arranjo de documentos exagerados. O modo mais prático é saber o que documentar e quando documentar para que isso não se torne o centro das atenções quando na verdade o que deveria ser o centro é ter um software em pleno desempenho para entrega ao cliente. Isso leva ao terceiro valor mencionado que é a colaboração com o cliente. Gomes explica que muitas vezes o processo de desenvolver um software é um aprendizado que ocorre conforme as etapas do projeto vão sendo constituídas. Ter uma relação colaborativa com o cliente é ponto chave para que esse projeto alcance seus requisitos e expectativas, e sim, é importante ter pontos colocados em contrato, até mesmo para a segurança tanto do contratante quanto do contratado, mas que, mais uma vez, esse não deve

ser o guia para o projeto, e sim, ambas as partes construirão uma relação de confiança e colaboração para juntos alcançarem o objetivo principal. Por fim, o autor aborda o tema de responder a mudanças, e que como citado, o processo de desenvolver um software é um aprendizado tanto para quem desenvolve quanto para o cliente, o que torna mudanças inevitáveis ao longo do caminho. O que seria interessante é receber essas mudanças como oportunidades para que o sistema seja feito de forma mais próxima às necessidades do cliente contribuindo para um resultado mais completo. Para essa adaptação é necessário que planejamentos sejam feitos durante todo o tempo, o que pode ser feito utilizando de ferramentas como o PDCA.

Além dos valores, o Manifesto conta com doze princípios que ajudam a formar os pilares do desenvolvimento ágil. Segundo Beck *et. al* (2001), os princípios do Manifesto Ágil são:

1. Nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado;
2. Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;
3. Entregar frequentemente software funcionando, de poucas semanas a poucos meses, com preferência à menor escala de tempo;
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho;
6. O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
7. Software funcionando é a medida primária de progresso;
8. Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
9. Contínua atenção à excelência técnica e bom design aumenta a agilidade;
10. Simplicidade -a arte de maximizar a quantidade de trabalho não realizado- é essencial;

11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis;
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo.

Gomes *et. al* (2014), aponta que ao chegar ao final dos doze princípios, pode-se notar que nada sobre uma sequência de fases ou de atividades que devem ser seguidas para a produção de um projeto de software de sucesso foi mencionado. O autor explica que a engenharia de software insiste em uma definição do que poderia ser uma “receita de bolo” para o desenvolvimento de sistemas, mas que até hoje, nenhum processo de software foi capaz de criar tal sequência de atividades o que nos leva a duas conclusões: continuar buscando o processo perfeito ou aceitar que processos perfeitos não existem e que é necessário partir para outras abordagens.

O autor continua falando que o que pode se notar dos princípios ágeis é que eles preservam o fato de que cada equipe, em cada projeto deve encontrar sua dinâmica de trabalho e que regras previamente estabelecidas quase não existem. A única regra básica que persiste é a da melhoria contínua em que o autor explica “a cada ciclo de trabalho, deve-se refletir sobre o que foi feito, aprender com o que não funcionou (descartando-o, se for o caso) e potencializar o que estiver dando certo... Passados os primeiros ajustes, a tendência é de convergência do ritmo, das expectativas e das prioridades dos envolvidos, reforçando o potencial de sustentabilidade dos trabalhos.” (Gomes *et. al*, 2014, p.15)

2.2.2 Extreme Programming (XP)

O método da programação extrema talvez seja um dos métodos ágeis mais conhecido atualmente. Beck (1999), aponta que esse método é mais útil para equipes pequenas ou médias que desenvolvem seus produtos a partir de requisitos vagos que podem sofrer mudanças rapidamente.

Assim como o Manifesto, a XP também conta com um conjunto de princípios e práticas em que o desenvolvimento deve se basear. Vettorazzo (2018), explicita que a XP possui alguns valores que norteiam o trabalho, o primeiro deles é a comunicação, essencial para o trabalho em equipe. A autora explica que esse método procura manter comunicações certas e fluidas. Bassi (2014), também comenta sobre o assunto, ao falar sobre comunicação o autor aponta que para a

criação de um produto é necessário haver muito entendimento e transparência entre os desenvolvedores. O autor ainda complementa dizendo que a comunicação é importante para espalhar o conhecimento e auxiliar na identificação de soluções além de ajudar a disseminar habilidades particulares para o grupo todo. O segundo valor que Vetorazzo (2018) e Bassi (2014) apresentam sobre a XP é a simplicidade. Vetorazzo (2018), aponta que para a programação extrema é melhor fazer uma coisa simples no início e investir depois para fazer modificações, se necessário, do que fazer algo complexo que pode nunca ser utilizado. Bassi (2014), explica que uma arquitetura simples é mais fácil de ser implementada do que uma com arquitetura mais complexa, o mesmo é válido para a modelagem e codificação. O autor aponta que “o software deve ser mantido o mais simples possível pelo maior tempo possível, pois sofisticação precoce dificulta o crescimento e a manutenção da arquitetura, atrasando a entrega de funcionalidades básicas.” (Bassi, 2014, p.61).

O terceiro valor abordado por Bassi (2014), é a coragem. O autor explica que é necessário ter coragem para inovar, mudar e aceitar que não se sabe tudo sobre o desenvolvimento de software, visto que essa é uma atividade complexa demais para ser tratada de maneira determinística. O quarto valor abordado pelo autor é o *feedback*, nesse ponto Bassi aponta que quanto mais rápidos os *feedbacks* acontecerem melhor será para o projeto, uma vez que quanto mais cedo eles ocorrerem mais cedo a equipe saberá se está no caminho correto, e se não estiver, pelo menos o esforço para corrigi-lo será menor.

Além dos valores, a XP conta com uma série de práticas a serem utilizadas para basear o desenvolvimento. Martin (2020), divide as práticas utilizando o diagrama de Ron Jeffries apresentado na Figura 4 a seguir que também pode ser chamado de Ciclo de Vida.

Figura 4 - Ciclo de Vida XP



Fonte: Desenvolvimento Ágil Limpo: De volta as origens. Martin, 2020.

Martin (2020), descreve a divisão dos três círculos do Ciclo de Vida. O autor explica que o círculo externo apresenta as práticas que são orientadas aos negócios da XP comparando-o com o processo do *Scrum*. As quatro práticas que compõe essa parte da figura fornecem o *framework* do modo com que a equipe de desenvolvimento do produto se comunica com os negócios envolvidos e as práticas que a equipe de negócios gerencia o projeto. Analisando um a um o autor descreve que o planejamento do jogo é a prática que informa como dividir o projeto em funcionalidades, tarefas e histórias se relacionando com as estimativas e priorização das atividades a serem desempenhadas, bem como o cronograma das funcionalidades requeridas. As pequenas versões são as que orientam a equipe a trabalhar pequenas partes do projeto de cada vez. Os testes de aceitação são os que marcam determinada etapa como “concluída” e também mostram a equipe como identificar critérios de conclusão incorretos. O último ponto dessa “fase” descrito pelo autor é a equipe como um todo que traz a noção de que uma equipe de desenvolvimento é composta por uma série de profissionais que trabalham juntos para alcançar o mesmo objetivo, sejam eles os desenvolvedores, gerentes ou testadores.

Na sequência Martin (2020), esclarece as práticas contidas no círculo do meio do diagrama. Nessa etapa os *frameworks* fornecidos se ligam a como a equipe de desenvolvimento se comunica e gerencia. O ritmo sustentável se refere a prática que impede que a equipe esgote suas funcionalidades antes do ponto de chegada e perca o fôlego durante o processo. O conceito de propriedade coletiva é aquele que assegura que o código e todo o projeto não esteja nas mãos de apenas uma pessoa ou de uma parte delas, o que é muito benéfico caso algum dos integrantes do time saia com o projeto em andamento, dessa forma todos estarão por dentro do que está acontecendo e poderão dar continuidade sem maiores problemas. A integração contínua é relacionada ao ciclo do *feedback*, que deve ser fechado com frequência para que todos saibam o andamento do projeto com espaços curtos de tempo. O uso de metáforas é a prática que cria terminologias para que todos da equipe “falem a mesma língua” com objetivo de se comunicar com mais facilidade a respeito do produto.

Por fim, o autor apresenta as práticas contidas no círculo mais interno do Ciclo de Vida da XP. Essas práticas são voltadas a orientar e restringir os programadores com o intuito de garantir uma qualidade técnica mais elevada. Martin (2020) começa explicando a programação em duplas, que é uma prática utilizada para compartilhamento de conhecimento entre a equipe técnica que auxilia na identificação de erros no código e colabora para que haja precisão e inovação durante o processo. O *design* simples guia a equipe para evitar esforços desnecessários e a refatoração incentiva a equipe na melhoria contínua buscando o refinamento dos produtos de trabalho. Por último o autor aborda o desenvolvimento orientado a testes que traz a sensação de segurança para a equipe que também o utiliza para avançar no desenvolvimento mais rapidamente e mantendo a alta qualidade.

Com relação a documentação, Bassi (2014), cita que esse é um ponto polêmico na indústria de software. Na XP as pessoas são o foco, assim, as interações entre elas são favorecidas o que deixa a documentação em segundo plano. O autor explica que nesse método a documentação é o próprio código e os testes automatizados realizados para verificação do que foi produzido, ainda assim, em alguns casos documentações intermediárias são necessárias para necessidades legislativas. “Além de código e testes, pouco material extra é produzido... A ausência de documentação tradicional é possível porque clientes e desenvolvedores

trabalham juntos, de modo que dúvidas sobre o código e sobre os requisitos são esclarecidas a partir de conversas, explicações, demonstrações e trabalho em equipe.” (BASSI, 2014, p.65)

2.2.3 Scrum

Vetorazzo (2018, p.17), define o *Scrum* como “um *framework* dentro do qual as pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. É leve, simples de entender e extremamente difícil de dominar.”

Assim como a XP, o *Scrum* também possui pilares que auxiliam no controle do processo, sendo eles: transparência, inspeção e adaptação. Maschietto *et. al* (2020) explica que o pilar da transparência se relaciona a visibilidade sobre os aspectos do processo que requerem definições em um padrão comum para que independente do observador, a leitura seja feita da mesma maneira. Sobre a inspeção o autor cita que os times que aplicam o *Scrum* tem que inspecionar os artefatos e o progresso do projeto para detectar possíveis desvios, é necessário ter uma periodicidade determinada para essas inspeções para que não haja interferências na execução das tarefas. Por fim, o autor explica o pilar da adaptação como sendo uma forma de solução desvios relacionados a qualidade em que a equipe deve encontrar uma solução o mais rápido possível. Maschietto *et. al* (2020, p.61) ainda cita algumas características dos times que utilizam essa abordagem, como: criativos, produtivos, flexíveis, multifuncionais e auto-organizáveis.

No *Scrum*, existem papéis principais em que cada membro tem sua atribuição envolvendo habilidades e responsabilidades. Wazlawick (2013) identifica três perfis importantes nesse modelo. O primeiro deles é o *Scrum Master*, que não é gerente nem líder, mas sim um facilitador que conhece bem o modelo e auxilia na solução de conflitos. O segundo perfil é o do *Product Owner* que é a pessoa responsável pelo projeto em si, tendo como atribuição marcar quais são os principais requisitos a serem tratados em cada sprint. O *product owner* também é responsável pelo ROI e por conhecer as necessidades do cliente. O terceiro perfil abordado pelo autor é o *Scrum team*, que é a equipe de desenvolvimento que não necessariamente é dividida em papéis como programador, analista etc. Assim, todos interagem para desenvolver o produto em conjunto. A recomendação é que as equipes sejam pequenas ou médias contando de 6 a 10 pessoas.

No método do *Scrum*, os projetos são divididos em ciclos chamados de *sprints* que incorporam um conjunto de atividades a serem executadas, sendo elas: requisitos, análise, projeto, evolução e entrega. Pressman (2016, p.78), explica que “o trabalho realizado dentro de uma *sprint* é adaptado ao problema em questão e definido, e muitas vezes modificado em tempo real pela equipe *Scrum*”. Vettorazzo (2018, p.17) define uma *sprint* como sendo “unidades de trabalho solicitadas para atingir um requisito estabelecido no *backlog* e que precisa ser ajustado pra atender o prazo já acordado para entrega.”

Maschietto et. al (2020), identifica a sequência de eventos de uma *sprint*, sendo eles: *Sprint Planning* (Reunião de Planejamento), *Daily Meeting* (Reunião Diária), *Sprint Review* (Revisão da *Sprint*) e *Sprint Restrospective* (Retrospectiva da *Sprint*).

Segundo o autor, a reunião de planejamento marca o início da *Sprint*. É nessa reunião que o *backlog* do produto será avaliado e onde o time determinará o que pode ser entregue como incremento nessa *sprint* bem como a organização para que o trabalho seja realizado. Para a realização dessa definição, o time precisa determinar a capacidade e produtividade do que pode ser realizado dentro daquela *Sprint*, assim também devem ser determinadas as atividades a serem desempenhadas para a entrega da *Sprint*. Ao final da reunião de planejamento todos devem estar alinhados com o objetivo daquela *Sprint*, com a lista dos itens a serem executados dentro do período proposto, essa lista se chama *Sprint Backlog*.

Maschietto continua explicando sobre as reuniões diárias, que devem ter no máximo 15 minutos de duração. Essa reunião é conduzida pelo próprio time de desenvolvimento e tem como objetivo que cada membro compartilhar o que foi feito até então para que o time permaneça alinhado.

Assim, quando algum membro identifica necessidade de entender melhor o processo ou se há a percepção de algum impedimento, após a reunião os problemas são tratados apenas com os membros necessários e o tempo da reunião é mantido.

A revisão da *Sprint* e a reunião em que todo o trabalho realizado e concluído pela equipe é evidenciado. Ainda sobre esse tema, Prikladnicki e Magno (2014), explicam que nessa reunião também são coletadas opiniões e impressões dos presentes sobre o trabalho apresentado para que, caso haja necessidade, ajustes sejam feitos no plano para a *Sprint* seguinte.

Dessa forma, na etapa de retrospectiva da *Sprint*, segundo Maschietto, o time poderá avaliar a si mesmo e traçar um plano de melhorias para aplicação na *Sprint* seguinte. Segundo o autor o objetivo dessa etapa é evidenciar e incentivar a continuidade das práticas que deram certo no andamento dessa *Sprint* além de encontrar alternativas para as práticas que não tiveram tanta eficiência.

2.2.4 OpenUp

O *OpenUp* é uma metodologia que segue o *Unified Process* (processo unificado) da engenharia de software, mas com uma cultura mais aberta e enxuta e possui um ciclo de vida iterativo e incremental que foca na natureza colaborativa para o desenvolvimento dos produtos. Assim como os métodos apresentados anteriormente, o *OpenUp* também possui princípios a serem adotados para sua utilização. Vetorazzo (2018), apresenta os princípios como: colaborar para que haja compartilhamento dos conhecimentos adquiridos e alinhamentos dos interesses, balancear as prioridades concorrentes para maximizar os valores das partes interessadas, concentrar-se na arquitetura na busca de minimizar os riscos e planejar o processo, e, por último, envolver os *stakeholders* para que haja *feedback* contínuo sobre o desenvolvimento.

Levando em conta os princípios apresentados por Vetorazzo, Balduino (2007) faz um comparativo interessante sobre como cada pilar do *OpenUp* pode ser facilmente identificado nos valores do Manifesto Ágil, como mostra a Tabela 2 a seguir.

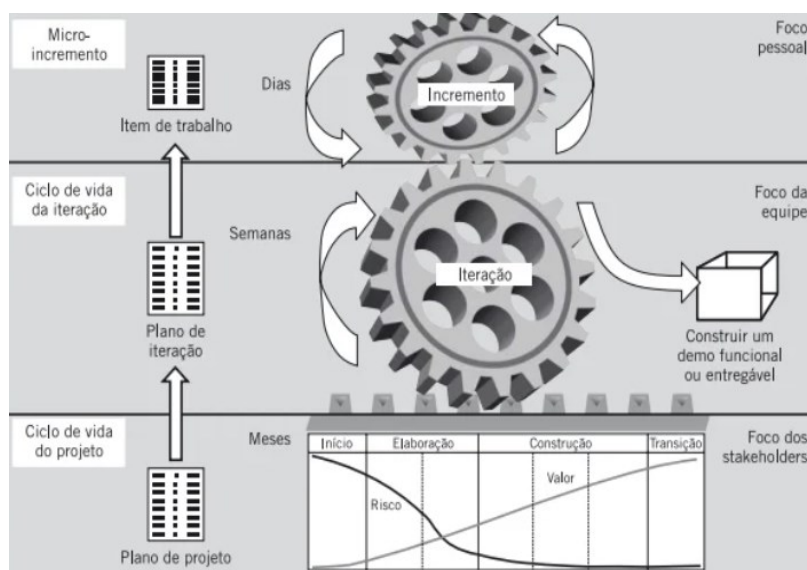
Tabela 2- Mapeamento entre os princípios do OpenUp e do Manifesto Ágil

Princípios do OpenUp	Valores do Manifesto Ágil
Colaborar para que haja compartilhamento dos conhecimentos adquiridos e alinhamento dos interesses	Indivíduos e iterações acima de processos e ferramentas
Concentrar-se na arquitetura buscando minimizar riscos e planejar o processo	Software em funcionamento mais que documentação abrangente;
Balanceamento das prioridades concorrentes para maximizar os valores dos stakeholders	Colaboração com o cliente mais que negociação de contratos
Envolvimento dos <i>stakeholders</i> para que haja feedback contínuo sobre o desenvolvimento	Responder a mudanças mais que seguir um plano

Fonte: Introdução ao OpenUp, Balduino R. (2007)

Papo (2014), explica que o método OpenUp pode ser mais facilmente entendido a partir do entendimento do seu ciclo de vida, apresentado na Figura 5 a seguir.

Figura 5 - Ciclo de Vida no OpenUp.



Fonte: Métodos Ágeis para Desenvolvimento de Software, Prikladnick, 2014

O autor explica que o ciclo de vida do projeto é dividido nas fases de iniciação, elaboração, construção e transição. As etapas de iniciação e elaboração compreendem as análises de risco referentes ao negócio e a arquitetura do projeto para que na fase de construção sejam produzidas as funcionalidades requeridas e os testes sejam feitos sem maiores problemas. Assim, o produto chega na fase transição com foco na implantação, nos testes beta e treinamento do usuário para utilização do sistema. Sobre o ciclo de vida da iteração o autor aponta que o *OpenUp* recomenda iterações de 2 a 6 semanas e que ao final desse período a equipe deve entregar um *build* executável do software, implementado e testado. As iterações têm início com reuniões de planejamento que servem para elaborar detalhes de arquitetura e detalhamento dos itens dos microincrementos. A iteração termina com uma avaliação do que foi feito em comparação com o objetivo inicial.

Por fim, o autor explica que os esforços dos membros da equipe são organizados em microincrementos que devem ser bem definidos para que haja controle diário e podem ser resultado de algumas ou de alguns poucos dias de trabalho.

Papo (2014), ainda aponta que os papéis dos envolvidos no projeto são divididos em seis atribuições. O primeiro deles é o stakeholder, que são todas as pessoas e grupos interessados no projeto, depois há o gerente de projeto que irá liderar o planejamento do projeto auxiliando a equipe a se manter focada nos objetivos. Outro papel apresentado pelo autor é o do arquiteto que é o responsável por definir a arquitetura do software e tomar as decisões técnicas mais críticas liderando os desenvolvedores tecnicamente. Os desenvolvedores são os responsáveis por desenvolver os requisitos dos sistemas. O *fester* desempenha o papel de realizar todo o esforço de teste do produto e finalmente, o autor apresenta o papel do analista que é encarregado de representar o cliente e suas necessidades sendo responsável por entender o problema e elicitando os requisitos.

2.2.5 Feature-Driven Development (FDD)

O FDD, desenvolvimento dirigido por funcionalidade, é um método ágil que frisa a orientação voltada a objetos. Ratamal (2014), explica que o ciclo de vida desse método é dividido em duas etapas: inicialização que é a etapa em que se concebe o produto que será construído usando uma visão geral de sua estrutura e

suas funcionalidades e a fase de construção em que a meta é entregar os incrementos do produto de forma frequente e com qualidade suficiente para ser utilizado pelo cliente.

Wazlawick (2013), explica que a fase de iniciação é composta por três processos: DMA (Desenvolver Modelo Abrangente) que como o nome já diz, consiste em produzir um modelo geral do processo seguido por um detalhamento em modelos mais específicos onde se estabelecem grupos de trabalho formados por desenvolvedores e especialistas de domínio. Após isso é iniciado o processo CLF (Construir Lista de Funcionalidades) em que se identificam as funcionalidades que satisfazem os requisitos levando terceiro procedimento que é o PPF (Planejar Por Funcionalidade) que visa gerar o plano de desenvolvimento para a fase seguinte que é formada por ciclos iterativos.

Concluídos esses três processos o desenvolvimento passa para a fase de construção. Wazlawick relata que essa fase incorpora dois processos: o DPF (Detalhar Por Funcionalidade) que consiste basicamente em produzir o *design* de implementação da funcionalidade e as atividades de detalhamento são realizadas para cada funcionalidade identificada, e o CPF (Construir Por Funcionalidade) que tem como objetivo a produção do código para as funcionalidades apontadas feito a partir do *design* produzido na etapa anterior e de acordo com o cronograma estabelecido.

2.2.5 Dynamic Systems Development Method (DSDM)

O DSDM também é um modelo ágil que se baseia no desenvolvimento incremental e iterativo com participação contínua do usuário durante o projeto. Segundo Pressman (2016), esse modelo utiliza uma versão modificada do princípio de Pareto em que “80% de uma aplicação pode ser entregue em 20% do tempo que levaria para entregar a aplicação completa (100%).” (PRESSMAN, 2016, p.79). Sobre esse assunto, Wazlawick (2013), aponta que o DSDM inicia seu estudo e implementação a partir dos 20% dos requisitos que serão determinantes para o sistema final. O autor ainda cita que esse modelo possui essencialmente três fases: o pré-projeto, ciclo de vida e pós projeto.

Wazlawick explica que a fase de pré-projeto compreende a identificação do projeto e sua negociação definindo o orçamento a ser utilizado e a assinatura do

contrato referente a ele. Nesse ponto, o DSDM se diferencia da XP e do *Scrum* no quesito documentação, pois é fortemente baseado nesse conceito.

Na sequência, o autor explica a fase do ciclo de vida, que consiste na análise de viabilidade, que é a fase em que se verifica se é realmente viável utilizar o DSDM para o desenvolvimento do produto. Além disso, há a identificação dos primeiros riscos que o projeto pode conter e se ele atenderá as expectativas de negócio da empresa. Após a conclusão dessa etapa, passa-se para a fase de análise do negócio em que as características do negócio e possíveis tecnologias são estudadas. Nessa etapa há o uso da técnica do *timeboxing*, que fixa previamente o prazo e orçamento a serem utilizados. Feito isso, há a análise dos requisitos que poderão ser implementados. É nesse momento que o princípio de Pareto se faz presente, pois há a possibilidade de que alguns requisitos levantados não tenham o tempo necessário para serem implementados, porém os 20% dos requisitos mais importantes para o funcionamento do sistema já terão sido concluídos.

A próxima fase do ciclo de vida do DSDM, segundo Wazlawick (2013), é a iteração do modelo funcional que tem como primeira tarefa a identificação do protótipo funcional, ou seja, há a identificação das funcionalidades que serão implementadas no ciclo atual, após isso é feita a agenda de quando e como cada uma dessas funcionalidades será implementada. Dessa forma, há a criação do protótipo funcional que é a etapa em que a implementação preliminar da aplicação é feita de acordo com a definição de datas feita na agenda. Por fim, há a revisão do protótipo que é feita tanto a partir de documentações quanto da avaliação do cliente.

Na próxima fase do ciclo de vida é feita a iteração de *design* e construção, que trata da integração das funcionalidades aprovadas. Nessa etapa outro protótipo é produzido incluindo todas as características não funcionais que não foram estabelecidas no protótipo da etapa anterior, bem como as mudanças solicitadas pelo cliente. A última etapa do ciclo de vida citada por Wazlawick é a fase de implantação que tem como foco entregar o sistema ao usuário e treiná-los para usufruir do produto.

Após a conclusão do ciclo de vida, Wazlawick explica que a última etapa do projeto é o pós-projeto, que é o período considerado de manutenção. Nessa etapa a evolução do software é considerada como uma continuação do processo de

desenvolvimento, sendo que se problemas forem apresentados há a possibilidade de retornar a fases anteriores, caso se faça necessário.

A seguir, a Tabela 2 traz um resumo das principais características dos Métodos Ágeis apresentados.

Tabela 2 - Resumo Métodos Ágeis para Desenvolvimento de Software

Método	Descrição
<i>Xtreme Programming</i>	<ul style="list-style-type: none"> • Útil para equipes pequenas ou médias; • Requisitos vagos ou mutáveis; • <i>Feedback</i> contínuo; • Desenvolvimento orientado a testes; • Foco em pessoas mais que documentação.
<i>Scrum</i>	<ul style="list-style-type: none"> • Framework para resolução de problemas complexos e adaptativos; • Processo dividido em sprints com um conjunto de atividades a serem executadas; • Reuniões diárias de acompanhamento.
OpenUp	<ul style="list-style-type: none"> • Foco na natureza colaborativa para o desenvolvimento do produto; • Esforços da equipe distribuídos em microincrementos; • Microincrementos formam a iteração que tem duração de 2 a 6 semanas; • O conjunto final das iterações formam o ciclo de vida do projeto.
FDD	<ul style="list-style-type: none"> • Desenvolvimento dirigido a funcionalidades; • De um modelo abrangente se constrói a lista de funcionalidades; • Planejamento distribuído por funcionalidade; • Detalhamento das funcionalidades para posterior construção.
DSDM	<ul style="list-style-type: none"> • Modifica o princípio de Pareto em que apenas 20% dos requisitos são determinantes para o sistema final; • <i>Timeboxing</i> utilizado para definição de tempo e orçamento para realização das etapas; • Há foco na documentação do projeto; • Entrega de protótipos funcionais.

3. MÉTODOS E TÉCNICAS DE PESQUISA

3.1 Natureza, Objetivos e Método da pesquisa

Quanto à natureza da pesquisa, é de abordagem Qualitativa, pois observou-se o comportamento dos dados coletados analisando a aplicação das ferramentas utilizadas e sua relação com os conceitos descritos no Manifesto Ágil de acordo com as informações obtidas nas entrevistas. De acordo com Prodanov e Ernani (2013, p.70), a “pesquisa qualitativa: considera que há uma relação dinâmica entre o mundo real e o sujeito, isto é, um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números.” Além disso, Vale diz que, os autores indicam que, “a utilização desse tipo de abordagem difere da abordagem quantitativa pelo fato de não utilizar dados estatísticos como o centro do processo de análise de um problema, não tendo, portanto, a prioridade de numerar ou medir unidades. Os dados coletados nessas pesquisas são descritivos, retratando o maior número possível de elementos existentes na realidade estudada.” Prodanov e Ernani (2013, p. 70)

Quanto aos objetivos, a pesquisa é do tipo Exploratória, pois busca-se maior compreensão sobre os fenômenos abordados, ou seja, como os métodos utilizados para desenvolver os produtos e como a gestão de projetos desse desenvolvimento funciona. De acordo com Prodanov e Ernani (2013, p. 51), a pesquisa exploratória é “a pesquisa se encontra na fase preliminar, tem como finalidade proporcionar mais informações sobre o assunto que vamos investigar, possibilitando sua definição e seu delineamento, isto é, facilitar a delimitação do tema da pesquisa; orientar a fixação dos objetivos e a formulação das hipóteses ou descobrir um novo tipo de enfoque para o assunto.”

Quanto ao método da pesquisa será o Estudo de Caso, pois o estudo aprofundou o tema nas empresas pesquisadas, especialmente mediante a realização de entrevistas em profundidade e visita aos sites para um maior entendimento sobre as soluções tecnológicas ofertadas, para que posteriormente fosse feita uma análise da aplicação ou não das metodologias ágeis na gestão de seus projetos. De acordo com Prodanov e Ernani (2013, p.60) O Estudo de Caso “consiste em coletar e analisar informações sobre determinado indivíduo, uma família, um grupo ou uma comunidade, a fim de estudar aspectos variados de sua

vida, de acordo com o assunto da pesquisa. É um tipo de pesquisa qualitativa e/ou quantitativa, entendido como uma categoria de investigação que tem como objeto o estudo de uma unidade de forma aprofundada, podendo tratar-se de um sujeito, de um grupo de pessoas, de uma comunidade etc.”

3.2 Etapas Metodológicas

Quanto às etapas metodológicas, esta pesquisa é dividida em 3 etapas: revisão bibliográfica, elaboração de um roteiro de perguntas para coleta de informações relacionadas ao desenvolvimento de software das empresas selecionadas e por fim, a análise dos dados obtidos.

3.2.1 Público-alvo da pesquisa e Fator de Decisão

De acordo com a caracterização da CNAE as empresas do ramo de tecnologia se dividem em três grupos básicos: indústrias de software, indústria de serviços de TI e telecomunicações. O grupo de empresas escolhidos para esse trabalho é o das indústrias de software pelo fato de que nesse segmento há uma maior probabilidade de se identificar características tradicionais ou ágeis no desenvolvimento de software. Dados da CNAE 2019, apontam que Londrina possui 64 empresas desse segmento, divididas em três atividades principais, como mostra a Figura 6 a seguir.

Figura 6 - Atividades exercidas pelas indústrias de software de Londrina e Região metropolitana.

CNAEs	TOTAL
Desenvolvimento de programas de computador sob encomenda	14
Desenvolvimento e licenciamento de programas de computador customizáveis	25
Desenvolvimento e licenciamento de programas de computador não-customizáveis	25

Fonte: Ministério da Economia/CGCIPE/RAIS, (2019).

O foco da pesquisa será empresas de pequeno médio e grande porte da cidade de Londrina. Para nortear a escolha levando em conta o porte, utilizou-se a

classificação do SEBRAE, a qual organiza esse fator entre empresas de comércio e serviços separadamente das indústrias. A Figura 7 a seguir mostra a classificação das instituições de acordo com a quantidade de colaboradores que a mesma possui.

Figura 7 - Classificação pelo porte da empresa

Porte	Comércio e Serviços	Indústria
Microempresa (ME)	Até 9 empregados	Até 19 empregados
Empresa de Pequeno Porte (EPP)	De 10 a 49 empregados	De 20 a 99 empregados
Empresa de médio porte	De 50 a 99 empregados	De 100 a 499 empregados
Grandes empresas	100 ou mais empregados	500 ou mais empregados

Fonte: SEBRAE-NA/ Dieese. Anuário do trabalho na micro e pequena Empresa C013, p. 17

3.2.2 Amostra

O fator de decisão sobre as empresas escolhidas para participar do estudo foi o de acessibilidade aos gestores ou desenvolvedores para que o contato seja facilitado e as informações sejam adquiridas de acordo com a disponibilidade de cada um. A pesquisa conta com uma amostra de 4 empresas, sendo uma de pequeno porte, uma de médio porte e três de grande porte. A Tabela 4 a seguir traz o compilado das empresas participantes, com o número de colaboradores que cada uma possui bem como uma breve descrição do perfil do entrevistado de cada companhia.

Tabela 3- Empresas participantes por porte

Empresa	Porte	N° Colaboradores	Perfil do entrevistado
Empresa A	Médio	80	Brunno, 37 anos. Há 2 anos na Empresa A. 14 anos ná área de TI
Empresa B	Grande	70 internos e 250 externos	Claudemir, 50 anos. Há 17 anos na Empresa B. 32 anos de experiencia em TI.
Empresa C	Grande	Mais de 10 mil	Lucas, 38 anos. 1 ano e 1 mês na EMPRESA C. 10 anos de experiencia em TI
Empresa D	Pequeno	15	André, 42 anos. Há 4 anos na Empresa D. 25 anos de experiênciã na área de TI

3.2.3 Método de coleta de dados

Para obtenção das informações necessárias, foi feita uma entrevista em profundidade, que segundo Duarte (2005), é um recurso que, com base em teorias

definidas pelo investigador, visa levantar respostas a partir da experiência subjetiva da fonte, a qual foi selecionada por obter as informações necessárias.

Para auxiliar a coleta de informações, um roteiro semiestruturado de perguntas foi elaborado para guiar as respostas do entrevistado para as informações que se desejava obter. O roteiro não necessariamente foi seguido à risca, visto que dúvidas ou outros questionamentos surgiram no decorrer da conversa em questão. Os questionamentos contidos no Apêndice A representam o roteiro que auxiliou na coleta das informações necessárias para posterior análise.

Para a apresentação dos dados coletados, as perguntas do roteiro foram divididas em 4 blocos para diferenciação de cada tema abordado os quais permitiram identificar os objetivos propostos para esse trabalho, começando por entender os métodos que as empresas utilizam no desenvolvimento de seus produtos passando a entender as semelhanças de valores com o Manifesto Ágil através de questões relacionadas ao cliente, equipe e documentação, para que por fim se possa verificar se há alguma característica de Métodos Ágeis tanto no gerenciamento dos projetos quanto no desenvolvimento dos mesmos. A estrutura de coleta de dados ficou assim organizada:

1. Início do projeto e documentação;
2. Participação do cliente;
3. Equipe, Desenvolvimento e Entrega;
4. Métodos Ágeis.

Por fim, para garantir que nenhum dado coletado fosse perdido, a entrevista teve seu áudio gravado mediante aprovação do entrevistado.

3.2.4 Análise dos dados

Após a realização das entrevistas, tendo o arquivo de áudio, foi feita a transcrição da conversa para facilitar a visualização dos resultados obtidos com a entrevista, bem como para utilizar diretamente falas importantes para a pesquisa na apresentação dos dados adquiridos.

A partir das respostas obtidas com a aplicação dos questionamentos foi feita uma análise das respostas para identificar quais características dos modelos ágeis cada empresa possui, como cada Empresa Atua na gestão de seus projetos e se é possível reconhecer traços dos princípios descritos no Manifesto Ágil ou de algum método ágil em específico.

As seções a seguir apresentarão as informações coletadas bem como a análise individual de cada Empresa Além de comparativo entre elas caso mais de uma possua o mesmo porte ou características de equipe similares.

4. RESULTADOS DAS ENTREVISTAS

4.1 Empresa A

A Empresa A foi fundada no ano de 2002 trazendo soluções ligadas ao mundo do agronegócio. Segundo a caracterização do SEBRAE, a Empresa A se encaixa como empresa de médio porte, contando com 78 colaboradores, além de seus 2 diretores, e tem como principais produtos softwares voltados para gestão das empresas do ramo bem como um software de CRM.

Dos softwares que a empresa desenvolveu, o software de gestão do negócio é o carro chefe da empresa. De acordo com o site da empresa o sistema pode conter vários módulos de diversas áreas como sementes, cereais, recursos humanos, financeiro, contábil, dentre outros, que podem ser adquiridos de acordo com a necessidade específica do cliente.

Em seguida temos software de CRM para equipe de vendas, que auxilia os clientes na performance do gerenciamento de suas vendas, além de monitoramento da equipe e relacionamento com os clientes.

4.1.1 Início do Projeto e Documentação

A entrevista com o representante da Empresa A foi feita presencialmente na sede da empresa. O entrevistado é o analista de sistemas sênior Brunno, 37 anos, que está na Empresa A há 2 anos e 9 meses participando em diferentes projetos dentro da empresa desde desenvolvimento a criação de painéis de Business Intelligence que auxiliam os clientes a visualizarem seu progresso e como andam as vendas e negociações.

Dando início as perguntas pré definidas no roteiro e para contextualizar como a Empresa A inicia seus projetos e atinge seus clientes, Brunno explica que a empresa possui basicamente duas formas de aproximação para que novos negócios sejam fechados. A primeira delas é a partir de um cliente principal com o qual o Software de Gestão foi desenvolvido em conjunto, que dentro do seu círculo de negócios possui outras empresas que podem vir a se interessar pelos produtos oferecidos pela Empresa A, sendo essa a principal forma de atração de novos negócios. Mas que também existem as empresas que ficam sabendo dos produtos

oferecidos por outros meios e que, assim, entram em contato para entender mais sobre os softwares e se as soluções se encaixam para se estilo de negócio.

Para iniciar o projeto, a Empresa A segue alguns processos padrão para análise do perfil do cliente. Nesse ponto, Brunno cita que *“Sempre que a proposta comercial é fechada, o próximo passo é a análise técnica, se houver a necessidade de módulo específico temos a parte de aderência técnica onde nós marcamos uma reunião com o cliente para ter maior acesso aos dados, tabelas, todas as estruturas que precisamos ter para alimentar o sistema/aplicação que temos.”* Para o desenvolvimento a partir do “zero” o analista aponta que uma outra reunião entre o ponto focal do cliente e o gerente de projetos é marcada para que haja a análise de dados a fim de encontrar todos os pontos que precisarão ser atendidos para que o projeto comece a avançar ou se ainda será necessário a análise de outros cenários envolvidos para que o desenvolvimento tome forma.

Sobre os requisitos do projeto, Brunno aponta que inicialmente o cliente tem uma ideia bem simplificada do que gostaria que o produto final fosse, mas que como a Empresa A já possui um *know how* consolidado na área existe um direcionamento durante a fase de definição de requisitos, *“De maneira geral o cliente tem uma ideia simplista do que ele precisa, tem uma noção, mas ainda não sabem o ponto em que querem chegar. Como nós temos um produto acabado na empresa muitas vezes ele norteia esse tipo de situação. Em casos específicos, a gente tem o produto pronto que juntamente com os módulos adicionais, são questões mais específicas para cada cliente, mesmo assim acabamos norteando o cliente quando ele não tem uma ideia formada sobre o assunto, nós trazemos esse ponto para ele, de como usar, tratar e fazer a apresentação dos dados. Mas há também os clientes que já sabem como querem o produto final, porém são casos mais raros.”*

Definidos os requisitos, para firmar acordo, a documentação gerada pela empresa é básica, porém seguida à risca, uma vez que o escopo do projeto é estabelecido e será seguido em todas as fases do desenvolvimento como um norteador do trabalho. Segundo a sexta edição do PMBOK (2017), o gerenciamento do escopo deve conter a definição do mesmo, a forma como será desenvolvido, monitorado e validado juntamente com os requisitos acordados com o cliente. No gerenciamento de projetos da Empresa A, essa definição é a base inicial para todo o processo, e é estabelecida desde o começo na documentação para que haja clareza nas entregas.

Brunno comenta que a empresa já notou uma deficiência na documentação, mas que já está trabalhando para melhorar pontos que acabam ficando em aberto, *“No ponto da documentação estamos tentando melhorar, ela já foi um pouco mais deficitária, mas temos sim uma documentação que já indica tudo o que teremos nesse projeto e quais são os pontos que serão atendidos. O que acaba ficando mais aberto são questões específicas da regra de negócio do cliente, porque por mais que tenhamos um produto acabado, algumas particularidades do cliente nós temos que adaptar.”* Mesmo com alguns pontos específicos de cada cliente ficando não tão claros, Brunno comenta que mesmo para adaptações é feito um levantamento do que o projeto irá conter colocando esse esboço juntamente com o escopo inicial e que qualquer iniciativa que fuja do escopo será alocada na fase subsequente, após a entrega do que foi previamente acordado.

4.1.2 Participação do Cliente

Partindo para o próximo tema e visando seguir um dos princípios do Manifesto Ágil, as perguntas se direcionaram para a participação e presença do cliente durante o desenvolvimento do projeto. Nesse ponto Brunno aponta que em todos os projetos a Empresa A procura ter um ponto focal dentro do cliente, que seria um colaborador responsável pela comunicação entre cliente e Empresa A para que o fluxo de informações entre ambas seja facilitado, *“Sempre buscamos ter um ponto focal dentro do cliente que será nosso canal de relacionamento e o que ficará balizando todos os processos. Então tudo o que a gente está desenvolvendo, fazendo e entregando, ele está presente como o primeiro contato entre a nossa equipe e o cliente”*.

Ainda sobre o ponto focal dentro da empresa, Brunno cita que mesmo que algumas coisas fujam de sua alçada, é responsabilidade do mesmo levar questões específicas para dentro da Empresa Cliente para que haja alinhamento das informações e que, caso seja necessário, haja uma reunião com a área de negócios envolvidas para que a definição do que deve ser adaptado ou desenvolvido fique clara para a equipe da Empresa A mantendo sempre o fluxo de informações alinhados entre ambas as partes.

4.1.3 Equipe, Desenvolvimento e Entrega

Com o escopo do projeto definido, é hora de definir a equipe de desenvolvimento e iniciar a etapa de desenvolvimento. Para esse quesito, o entrevistado cita que, visto que o produto acabado já existe, o desenvolvimento segue uma sequência lógica para a captação, transformação e apresentação dos dados dentro dos módulos escolhidos pelo cliente.

A Empresa A possui uma equipe de desenvolvimento e, segundo Brunno, a alocação de um ou mais analistas depende do cliente e do tamanho do projeto, *“A grosso modo colocamos um analista por projeto, mas em casos especiais alocamos mais de um analista fora o coordenador e gerente de projetos. E aí sim, quando há mais de um analista envolvido são definidas algumas tarefas para que cada um ataque uma frente diferente do projeto para que a entrega seja mais fluida.”*

Brunno também aponta que a comunicação entre o gerente de projetos e os membros da equipe é bem facilitada e que, como sugere o Manifesto, geralmente ocorre face a face para que todas as informações sejam passadas para os membros sem qualquer interferência, *“Internamente, quando os assuntos a serem tratados dizem respeito apenas a equipe da Empresa A, também priorizamos a conversa direta. Antes nós éramos 100% presencial, mas agora como adotamos o modelo híbrido também tivemos que melhorar e adaptar a nossa comunicação utilizando outras ferramentas para que todos estivéssemos alinhados”*. Para facilitar a comunicação com o *focal point*, é criado o grupo do projeto que contém não somente a gerência da Empresa A e o representante do cliente, mas também todos os analistas envolvidos no projeto para que todos saibam das informações ao mesmo tempo, uma vez que as mudanças podem ocorrer a qualquer momento e apesar da decisão de implementar a mudança ou não passar primeiro pela gerência os analistas já ficam cientes de que algo pode estar por vir ajudando na adaptação das tarefas caso a mudança seja validada e inserida na *sprint* que está em andamento.

Na sequência, também seguindo os preceitos do Manifesto Ágil, outro ponto importante comentado é a forma com que a equipe lida com possíveis mudanças durante a execução das *sprints* previstas no início do projeto. Sobre isso, Brunno comenta que, *“A equipe é bem aberta a essas mudanças, porque atualmente quase todas as mudanças que o cliente propõe não vão diretamente para o analista.”*

Primeiro é passado para o gerente de projetos que vai decidir se a mudança será executada ou não, sempre conversando com o focal point e internamente na nossa empresa.”

O analista aponta que hoje as entregas são divididas nas conhecidas *sprints* que são acordadas com o cliente já definidas quando e o que será entregue em cada uma, *“As entregas são divididas em sprints dentro do projeto que são acordadas com o cliente já meio que definido quando será entregue e o que será o conteúdo entregue. Em alguns momentos nós acabamos realizando duas sprints ao mesmo tempo, se for possível. Mas no geral seguimos as entregas de acordo com as sprints acordadas.”*. Esse método de divisão ainda é recente dentro da empresa e os membros mais antigos da equipe ainda estão se adaptando a essa forma de divisão.

Levando em conta o escopo inicial, o analista cita que, com o *know how* que a empresa já possui, a divisão das etapas em *sprints* facilitou o entendimento e sequenciamento das atividades que vão ser realizadas até a entrega final, e que, após sua definição, elas serão seguidas até o final, *“Como a maioria dos nossos projetos já tem um processo de maturação muito bem definido, as sprints ficam bem claras para a equipe. Então apresentamos a proposta para o cliente do caminhar e desenvolvimento das sprints e tentamos ao máximo seguir elas até o final. Como o nosso know how já é bem definido, acabamos não tendo muitas surpresas quanto a isso. É só quando entramos em alguma fase especial do desenvolvimento que colocamos o levantamento de horas de acordo com a capacidade do analista, como temos analistas júniores e sêniores, sabemos que o tempo será diferente de acordo com quem será alocado para o projeto e assim definimos essa sprint ou dividimos caso seja necessário.”*

4.1.4 Métodos Ágeis

A fase final da entrevista foi dirigida para um dos pontos centrais dessa pesquisa que é a adoção dos métodos ágeis pela empresa entrevistada. Nesse quesito a Empresa A está passando por uma mudança interessante. A gestão de projetos da empresa mudou há pouco tempo, e a nova gestão está no processo transitório de aplicar alguns métodos ágeis que tenham a recepção e incorporação facilitada pelos membros da equipe. Sobre isso, Brunno comenta *“Estamos vindo de*

uma mudança na gerência de projetos, saindo de uma filosofia e partindo para outra. Algumas coisas nós mesmos enquanto analistas sentíamos que precisava mudar, porque conseguimos sentir a deficiência no processo, e vendo alguém disposto a aplicar algo que melhorasse nosso processo animou bastante o pessoal para que o método fosse adotado e seguido.”

Como percebido na seção anterior um dos métodos que está sendo aplicado na empresa é o *Scrum*, que trouxe mais agilidade no processo e clareza nas entregas com a aplicação das *sprints*, saindo de um processo mais rígido para algo mais fluido “*Antes nosso processo era um pouco mais engessado e agora estamos melhorando e aplicando isso na empresa para que a gente possa crescer. Era até mesmo algo cultural, nem todos da equipe tinham esse conhecimento e envolvimento com essas metodologias para que fossem aplicadas.”*

A maior dificuldade que a empresa está enfrentando com relação a adoção dos métodos ágeis é pela questão cultural. Existem alguns analistas que estão a bastante tempo na empresa e estavam acostumados com o método antigo de desenvolvimento, alguns até mesmo não conhecem metodologias ágeis, mas Brunno comenta que com o amadurecimento da equipe, essa percepção de agilidade vem crescendo “*A nossa gerência de projetos tem batido bastante nessa tecla dos métodos ágeis, estamos nesse processo ainda cultural saindo de não ter nenhuma metodologia para ter uma metodologia aplicada. Nem todos os analistas estão adaptados ou tem o conhecimento dessas metodologias, mas as mais utilizadas aqui dentro é o Scrum e Kanban que estamos tendo mais facilidade na adaptação e as que estão sendo mais culturalmente aceitas na empresa. Temos um caminho longo a ser percorrido, até mesmo porque de uns tempos para cá tivemos um crescimento no número de analistas e isso gera uma adaptação tanto para os que já tem certo tempo de casa quanto para os novos que estão chegando. As noções sobre essas metodologias vem crescendo de acordo com o amadurecimento da equipe.”*

Brunno conta que a com a adoção do *kanban*, a equipe passou a enxergar melhor o fluxo do processo, se tornando algo mais didático e visual. Sobre o assunto, Vale (2014, p. 142), cita que “*Kanban* se utiliza desses elementos (fluxo e valor), comuns à maioria dos processos para projetar um mapa visual que represente o modelo de trabalho da forma como ele é atualmente”. O autor ainda comenta que esse instrumento visual “cria uma visão compartilhada de como “o

todo” do sistema de trabalho está organizado”. Com a adoção de quadros físicos de *kanban*, a equipe visualiza de forma mais prática em que ponto o processo está, quais serão os próximos passos e o que já foi finalizado, evitando que mais de uma pessoa desempenhe a mesma tarefa ao mesmo tempo e formando uma linha de raciocínio para o desenvolvimento.

O analista ainda comenta que com a adoção dessas duas metodologias um dos maiores gargalos do processo da empresa foi suprido, *“A princípio as metodologias que estamos aplicando já estão resolvendo o que era o nosso maior gargalo, que era organizar os projetos dentro da empresa tornando o processo mais visível o que com a adoção dessas metodologias estamos conseguindo melhorar e até mesmo mensurar coisas como tempo e desenvolvimento nos tornando mais assertivos nesse sentido.”*

Quando perguntado algum outro ponto de melhoria que poderia ser resolvido com a adoção de algum método ágil, Brunno é bem realista com a visão da empresa e o caminho a ser percorrido *“Estamos tentando amadurecer essas duas metodologias para sentir se há a necessidade de adotar alguma outra. Por enquanto essas duas estão atendendo bem as nossas necessidades. Para quem conhece as metodologias ágeis é importante saber como elas se encaixam no nosso cenário de desenvolvimento. Como trabalhamos com um produto base e não temos que a cada dia desenvolver algo novo, algumas metodologias não se encaixam no nosso processo.”*

4.2 Empresa B

A Empresa B está no mercado há mais de 50 anos ofertando diferentes soluções tecnológicas. Contando com uma rede nacional de concessionárias, a Empresa B já está presente em mais de 150 cidades ao redor do Brasil e tem mais de 17 mil usuários. As soluções giram em torno da Gestão Contábil e Gestão Empresarial. Hoje a Empresa Conta com mais de 250 colaboradores entre internos e externos, se enquadrando como uma empresa de grande porte.

De acordo com o site da empresa, um dos sistemas de gestão contábil que a empresa possui é um software que auxilia o cliente com todas as questões financeiras ligadas ao seu negócio tudo concentrado no mesmo software para que o

cliente tenha resultados e visão de como a empresa está seguindo com um software de qualidade de fácil entendimento.

Além de soluções contábeis, a Empresa B também conta com softwares para diferentes ramos, atacando a mercados distintos como do comércio atacadista no qual possui um software que disponibiliza ferramentas que auxiliam no processo de compras e tem integração com todos os setores também contando com o módulo de vendas em que possui metas por vendedor, análise de vendas por seguimento, comissão por vendedor, análises de orçamento e muito mais.

Outros produtos da empresa são seus softwares voltados a indústria, sendo eles de ERP, MRP 1 e 2 bem como softwares para PCP. Esses softwares são voltados as atividades do processo produtivo desde a engenharia de produtos, até a entrega dos produtos acabados. O sistema conta com planejamento de produção, análise da demanda e parâmetros para reposição e outras funções necessárias para o MRP, são sistemas completos para serem utilizados em indústrias de todos os ramos proporcionando uma gestão preparada para flutuações de mercado e integrada que conduza o processo produtivo de forma eficaz e com qualidade.

4.2.1 Início do Projeto e Documentação

A entrevista com o representante da Empresa B foi feita de forma on-line devido a agenda do entrevistado. O representante é Claudemir, 50 anos, gerente comercial da Empresa B Software fazendo parte da empresa há quase 17 anos. Possui mais de 20 anos de experiência com tecnologia da informação, é professor de informática, formado na área de TI com especializações em gestão de negócios, consultoria empresarial e atualmente faz parte do mestrado em engenharia de software da Universidade Tecnológica Federal do Paraná em Cornélio Procópio. A empresa não será identificada por questões de permissão.

Dando início a entrevista, sobre uma contextualização da Empresa B, Claudemir explica *“A Empresa B é uma software house que está no mercado a mais de 50 anos. E isso quer dizer que estamos no mercado antes mesmo do pc, somos da época do framework. Possuímos 2 grandes pilares de trabalho, o primeiro é a área contábil onde somos produtores de software contábil, ou seja, para empresas contábeis. A outra linha que nós trabalhamos é a nossa linha de parceria, BRIDcoligadas a nós onde cada empresa tem uma especialidade, por exemplo:*

temos uma empresa especialista em desenvolver softwares para postos de gasolina, uma especialista para software em shoppings, uma empresa especialista em desenvolver softwares para a web. Então para cada linha vertical tem um CNPJ parceiro, desenvolvedor.”

Quanto ao início do projeto, o gerente explica que eles partem de um processo chamado “aderência”, no qual há a análise do perfil do cliente, de suas necessidades e interesses, *“Nesse ponto nós envolvemos nossos consultores que são pessoas com habilidade com o sistema, e que vai implementá-lo na Empresa Cliente, dando treinamento e fazer o sistema rodar e funcionar. De acordo com o tamanho da empresa será definida a quantidade de pessoas envolvidas.”*

Sobre os requisitos, assim como na Empresa A, a Empresa B trabalha com produtos já desenvolvidos e testados pela empresa, que a partir da necessidade do cliente passará por ajustes até atender a real necessidade do consumidor. Neste ponto, Claudemir afirma que os clientes já sabem o que querem quando chegam a Empresa B para fechar uma possível parceria. Como a empresa já está há anos no mercado e suas soluções já são reconhecidas, os clientes já têm conhecimento sobre os softwares, então quando chegam até a empresa, Claudemir explica que

“Quando o cliente chega na Empresa B ele já te os requisitos que gostaria em seu produto, a gente analisa o que ele precisa, se nós tivermos o produto entregamos pra ele.”

Seguindo para documentação, o gerente explica que a empresa já possui uma documentação pré definida e que a partir da Aderência, tópicos serão adicionados de acordo com a necessidade que o cliente apresentar, *“A documentação é interna, que são basicamente questionários de entrevistas de registros de acordo com a necessidade do cliente, como a gente tem bastante conhecimento na demanda, nós fazemos pequenos questionários, por isso até chamamos de aderência, para entender a real necessidade do cliente, com o que o mercado exige.”* A partir das respostas obtidas com os questionários a documentação base é modificada para abordar os temas que concernem ao cliente em questão.

4.2.2 Participação do Cliente

No ponto da participação do cliente, Cludemir é bem sucinto na explicação. A consolidação da Empresa B no mercado torna as coisas mais fáceis para ambas as partes. O cliente procura a Empresa B pela confiança no *know how* que a empresa possui. O gerente comercial ainda comenta que o cliente está sempre presente em todas as fases do projeto, porém como o produto no geral já está pronto e apenas alguns incrementos e adaptações devem ser feitas até a entrega final, o contato é constante no que diz respeito a atualizações do processo e acordo das entregas.

Como na EMPRESA A, a Empresa B também adota a abordagem de ter um *focal point* dentro da Empresa Cliente para que todas as atualizações e cronogramas sejam repassados e validados de acordo com o tempo requisitado pelo cliente para a entrega do software. Na seção a seguir, poderemos perceber que o desenvolvimento é feito de forma facilitada justamente pelo *know how* da empresa e pelo tempo de mercado que a mesma possui. Sendo assim, a participação do cliente se restringe a buscar a Empresa B com o que ele precisa para a sua empresa e entender se os produtos ofertados se enquadram na sua necessidade apontando as alterações ou incrementos desejados para que a entrega ocorra quando planejado.

4.2.3 Equipe, Desenvolvimento e Entrega

Partindo para o entendimento de como a equipe é formada para a realização do projeto, Claudemir esclarece que dada a natureza dos softwares que a Empresa B oferece, a equipe é formada por colaboradores de diferentes áreas, *“Temos analistas de negócio que são os contadores, advogados, administradores, que são as pessoas que enxergam o mercado. Quando o mercado se altera, a equipe faz a análise, entregam essas análises de mercado para os analistas de sistemas onde os analistas de sistema entregam a instrução para o programador fazer, aí depois vem a rotina de desenvolvimento, depois o processo de testes de qualidade, a versão beta, para então gerar a versão final e ir para a distribuição.”*

O gerente ainda comenta que devido ao estilo do trabalho e com a vinda da pandemia, as equipes não estão mais concentradas em um lugar só, até mesmo porque a Empresa Atua em diferentes cidades do país. Quando perguntado sobre como ocorre a comunicação nesse estilo de trabalho Claudemir comenta, *“A comunicação é através de tecnologia pura, eles estão vivos na tecnologia. Temos*

uma plataforma que nós desenvolvemos, o CRM, em que entrega, atividades, tarefas, ficam contidos nela. Como nós temos vários cases eu não consigo encontrar no mercado essa tecnologia híbrido a que eu preciso, por isso nós mesmos desenvolvemos, a técnica também foi desenvolvida pela gente, por isso falei que adaptamos o que vem de encontro a nossa necessidade.”

Sobre os métodos utilizados pelos programadores no desenvolvimento do produto, Claudemir comenta que não há a utilização de um método em específico visto que os clientes e o trabalho em si são adaptativos, ou seja, o método utilizado para desenvolver um software para um posto de gasolina não necessariamente será o mesmo que para o desenvolvimento de um programa para gerenciamento de um shopping, por exemplo. O método que o desenvolvedor irá utilizar depende do cliente e da necessidade encontrada. Ainda sobre os métodos, Claudemir comenta que, para auxílio, o programador recebe uma ficha de desenvolvimento com as características requeridas para aquela etapa do processo e deve segui-la como base.

Sobre a ficha entregue ao programador como norteadora do processo, o gerente comercial explica *“A ficha de desenvolvimento contém todas as características que ele vai precisar implementar. Nós desenvolvemos uma tecnologia em que até mesmo o design e o desenho quem desenvolve pra gente é a própria ferramenta, ele só vai modelar, mas quem gera o código não é ele, ele só confere o código e faz as implementações necessárias. Nunca se pega nada do zero. Vamos dizer que a ferramenta faz 60% e ele faz 40%, porque a ferramenta faz toda a parte de padronização, regras e classes, tudo isso a ferramenta faz sozinha. Então ele pega aquela regra que o analista entregou, lê e coloca essa implementação naquilo que a ferramenta entregou para ele.”* Ou seja, a ferramenta faz a parte base do código e o entrega de forma padronizada para que o programador faça apenas a parte do que irá gerar a modificação no módulo ou a implementação necessária para atender a necessidade requisitada.

Ainda sobre o processo de desenvolvimento, Claudemir conta que a técnica utilizada foi feita a muito tempo e que até mesmo a linguagem de programação foi desenvolvida pela empresa, *“A própria linguagem e programação é desenvolvida pela Empresa B, um dos nossos diretores é formado no ITA, então quando veio toda essa parte de desenvolvimento, quando saímos do método antigo, nós acabamos criando a nossa própria linguagem. Como ainda não existiam profissionais nessa época, até porque não existiam nem os computadores, houve uma importação de*

profissionais para cá que foi quando percebemos que era necessário produzir nossa própria linguagem.”

Sobre as entregas do produto, Claudemir aponta que algumas vezes elas acontecem de forma imediata, porque no geral o sistema é entregue ao cliente sem muitas modificações, então, para a Empresa B, o que conta para entrega seriam as atualizações *“A entrega das nossas atualizações são automáticas, o sistema é automatizado, quando há alguma mudança na versão do software o cliente recebe a versão nova com as alterações e correções. Muitas vezes o cliente nem percebe essas alterações, nos contrata por causa disso.”*

O gerente ainda comenta que um dos pontos importantes da Empresa B e um dos porquês da empresa ser requisitada é o pós vendas ofertado ao cliente, *“O pós atendimento é contínuo. Depois da implantação o contato que a empresa vai ter com a gente é o suporte. Esse suporte existe de duas formas, tanto na matriz quanto nas filiais, mas é contínuo, porque como nós atendemos legislação, a lei muda e então nós temos que ir lá, fazer a alteração no sistema e entrega a atualização para o cliente sempre mantendo a qualidade de nossos serviços.”*

4.2.4 Métodos Ágeis

Chegando ao ponto dos métodos ágeis, Claudemir demonstra conhecimento sobre o assunto, não de métodos específicos como XP ou Scrum, mas sobre como o modelo adaptativo desses modelos se encaixam no dia a dia da Empresa B: *“Nós sabemos tudo de metodologias ágeis, aplicamos e fazemos a incorporação dos processos. Métodos ágeis são métodos que são aplicados na tecnologia, são formas de trabalho. Então esses métodos são criados conforme a demanda vai acontecendo, nós vivemos isso porque o mundo muda e as vezes nós não estamos preparados, temos que criar adaptações que nos provocam a questão de mudanças na regra de desenvolvimento, treinamento, entrega.”*

Além dos métodos apresentados anteriormente, existe a percepção de uma filosofia ágil para o gerenciamento de projetos que mudam tanto a forma de trabalho como os processos utilizados, o que pode ser percebido nos projetos da Empresa B. Sobre o assunto, Boerg (2018), explica que processos e métodos de trabalho considerados inovadores acompanham projetos inovadores de forma que a equipe

possa ser adaptada e remanejada para melhor atender a realidade do projeto e da demanda.

Para exemplificar como o método da Empresa B é adaptativo, Claudemir utiliza a pandemia do COVID-19, *“Um exemplo bem claro da nossa adaptabilidade é a pandemia, o mercado não estava muito preparado para o home office, nós, como somos da área de tecnologia temos muita habilidade com relação a isso, então tivemos uma adaptação muito rápida até mesmo com relação de tirar os programadores de lugar e colocar em outro, e isso quem nos proporciona é técnica, que é o que você chama de métodos ágeis. Nesses métodos existem várias regras, várias formas de aplicabilidade e a gente vai atualizando e trabalhando com elas.”*

A adaptabilidade é uma das principais características abordadas tanto pelo Manifesto Ágil quanto pelos métodos desenvolvidos a partir dele. Assim, Claudemir finaliza, *“Os métodos ágeis nascem de uma necessidade de mercado. O mercado que faz essa mudança, nós acompanhamos dois fatores, o desenvolvimento e o financeiro, para entregar valor tanto para gente quanto para o usuário. Se a metodologia não estiver atendendo nós temos que fazer adaptações.”*

4.3 Empresa C

A Empresa C é uma prestadora de soluções e serviços em tecnologia da informação que não terá seu nome divulgado.

Em sua gama de serviços alguns exemplos de seus produtos são plataformas de ERP em nuvem, sistemas de bancos e soluções de customer intelligence. A Empresa C conta com clientes de diversos setores como bancos, indústrias, empresas de telecomunicações, seguros, educação e serviços públicos.

Um dos produtos da Empresa C que tem como foco o desenvolvimento ágil é um software que permite que as empresas alinhem sua estratégia com a execução da equipe ofertando uma visão unificada de projetos e portfólios ajudando as partes interessadas e a equipe de negócios a trabalhar em conjunto de forma eficaz.

No Brasil, a Empresa C conta com diversos projetos com empresas de todos os ramos. Com um centro de suporte técnico, localizado em Londrina, projetos não apenas com empresas brasileiras são gerenciados, como o projeto da Exelon, por exemplo, empresa estadunidense de energia, que contém todo o suporte de Help Desk aqui na cidade.

4.3.1 Início do Projeto e Documentação

Como a Empresa C no Brasil se divide basicamente em projetos para diferentes clientes, a pesquisa na empresa se dirigiu a um projeto em específico, o de uma mineradora, que será chamada de “mineradora” para que não haja identificação.

Tendo uma estrutura tão grande, a “mineradora” conta com muitos sistemas e um enorme time de TI. É nesse ponto que a Empresa C entra em campo. Como uma parceira do grupo de TI da “mineradora” a Empresa C desenvolve sistemas e presta suporte técnico para os demais sistemas utilizados pela empresa.

O representante da Empresa C no projeto “mineradora” é Lucas, 38 anos, líder técnico da Corporate, projeto de sustentação do TI da parceira. Como forma de iniciar a apresentação do projeto, Lucas explica *“Estou no projeto a um ano como líder técnico, nosso projeto conta com 46 colaboradores divididos em “torres” que são as subdivisões dos colaboradores envolvidos. Nós contamos com mais de 60 sistemas como SharePoint, Dotnet, Java, VB6 dentre outros. Nosso trabalho tem basicamente dois tipos, o primeiro deles é a prestação de suporte técnico para sistemas de diferentes áreas da “mineradora”, então quando algum desses sistemas dá problema o usuário entra em contato com a gente pra abrir um ticket. Se estiver no nosso escopo resolver o problema nós resolveremos, se não estiver, a gente encaminha para o time responsável. E a segunda forma de trabalho é o desenvolvimento de sistemas propriamente ditos, desde uma simples alteração em um sistema existente quanto ao desenvolvimento de acordo com o pedido do cliente.”*

Como a Empresa C tem um acordo fechado para essa prestação de serviços junto a “mineradora”, o início dos projetos de desenvolvimento (foco dessa pesquisa), se dão de uma forma um tanto informal. Lucas comenta que como um processo “padrão” para o início do projeto a equipe do cliente entra em contato com a ideia do que gostaria para o sistema e dá um prazo para a elaboração da proposta. Nesse meio tempo, os desenvolvedores da Empresa C elaboram a forma que o produto irá tomar e buscando apresentar esboços de orçamentos financeiros e estimativas de entrega. Sobre o assunto Lucas comenta *“Trabalhamos para*

desenvolver módulos de alto nível, porém muitas vezes no começo o cliente só tem a ideia do que precisa, seja um sistema para barragens, ou uma melhoria em um sistema que já entregamos. Muitas vezes o pedido do time da “mineradora” tem só uma linha, só a ideia principal mesmo, e fica sob a nossa responsabilidade decidir como o desenvolvimento vai acontecer, essa ideia inicial na maioria das vezes vem com requisitos que nem vão ser utilizados no produto, eles dão apenas um norteamento do que deverá ser entregue. Assim eles dispõem um tempo para que a gente elabore a proposta, colocamos o escopo e a descrição que nosso time pensou bem como as horas que vamos utilizar no projeto. Depois disso geramos o cronograma e levamos para validação.”

Após a validação da proposta apresentada pelo time da Empresa C o sistema começa a ser desenvolvido. Sobre formalização do projeto e documentação utilizada para tal Lucas aponta que esse é um ponto que precisa ser melhorado, “Sobre documentos, como a “mineradora” já tem uma parceria de 7 anos fechada com a Empresa C não há nada formal ou padrão que a gente usa, nem por parte da Empresa C nem por parte da “mineradora”. A proposta fica documentada, mas não há um contrato assinado por ambas as partes para o desenvolvimento começar, isso já está “incluso” no contrato inicial entre as duas companhias, mas é algo que precisa ser melhorado.”

4.3.2 Participação do Cliente

Sobre a participação do time da “mineradora” durante o projeto, Lucas comenta que a relação entre o time de TI da “mineradora” e o da Empresa C acontece de forma facilitada, “O contato com o time deles (“mineradora”) ocorre de uma forma muito fácil, a gente sempre está em contato com quem solicitou o sistema, geralmente é alguém do TI deles que pegou a informação com o departamento que está pedindo o produto e aí eles entram em contato com a gente. Com o TI responsável por aquele projeto em específico a gente tem contato quase que diário para troca de informações. Também procuramos agendar uma reunião semanal com o pessoal do departamento que vai utilizar o sistema para apresentar em que ponto estamos e se o que a gente pensou é realmente aquilo que eles precisam. Como os requisitos trazidos para a gente são meio rasos, a gente procura ter esse contato pra não começar a desenvolver algo que não é o que eles queriam.”

4.3.3 Equipe, Desenvolvimento e Entrega

No projeto da “mineradora”, Lucas explica que o desenvolvimento segue o “básico” de todo projeto do ramo, começando pela análise, em que a equipe analisa os requisitos iniciais solicitados pelo cliente dando forma ao sistema que será desenvolvido e até mesmo incrementando a ideia inicial para que dali saia um sistema funcional. Após a análise e classificação do que será feito, o desenvolvimento começa. Os programadores alocados no projeto dependem da linguagem que será utilizada para a programação, *“A equipe de desenvolvimento depende da linguagem que será utilizada. Se o sistema tiver que ser desenvolvido em Java, programadores com domínio nessa linguagem serão alocados no projeto, se for um desenvolvimento em VB6, serão alocados pessoas com mais facilidade para ela e assim por diante. Depende muito do que vai ser desenvolvido mesmo, um sistema para web, um software para barragens, tudo depende do produto em questão.”*

Com o desenvolvimento em andamento, o próximo passo são os testes do sistema, *“Conforme os módulos são desenvolvidos e o sistema vai tomando forma nós começamos os testes do que foi feito para ver se vai atender aos requisitos, se está realmente funcionando e o que precisa ser melhorado. Assim, no teste a gente tem uma visão de como o sistema está e o que falta para as próximas etapas.”* Dessa forma, com o andamento do projeto, Lucas comenta que outras reuniões com o time de TI e do respectivo departamento são agendadas para apresentação de protótipos do sistema para validação do cliente para que se passe para a próxima fase ou para o próximo módulo a ser desenvolvido, ou seja, as entregas vão ocorrendo parcialmente para que o que está sendo desenvolvido e o que cliente precisa estejam alinhados.

Sobre a comunicação entre a equipe, Lucas comenta que os membros estão sempre em contato, *“Geralmente cada um da equipe vai atacar uma frente diferente do projeto, mas a gente precisa e procura sempre manter o contato até mesmo porque nós vamos pensando e desenvolvendo os requisitos e aí mais que nunca a equipe precisa estar alinhada para que dois programadores não comecem pelo mesmo requisito e que o que um esteja desenvolvendo complemente o processo do*

outro e assim a gente chegue em um software funcional com um desenvolvimento harmonioso.”

Sobre mudanças no escopo e como a equipe recebe isso, Lucas é bem realista *“Como na maioria das vezes os requisitos são bem básicos, a gente já espera que haja mudanças durante o processo de desenvolvimento, o que não quer dizer que a equipe goste disso. Muitas vezes em algumas apresentações que marcamos, o cliente percebe que estamos no caminho, mas que não era exatamente aquilo que eles precisavam, a equipe aceita as mudanças sim, mas essas mudanças podem gerar retrabalho, o que ninguém gosta. Outro ponto que a gente é afetado são nas entregas, porque precisamos mudar o planejado para adaptar o que o cliente pediu.”*

4.3.4 Métodos Ágeis

Com o final do tempo disponível para a entrevista chegando, Lucas acaba sendo bem sucinto e realista com relação a adoção dos métodos ágeis por parte de seu projeto *“Sobre os métodos ágeis é claro que a gente já ouviu falar, a Empresa C em si procura ter essa visão ágil para desenvolvimento de projetos, mas aqui no nosso projeto acabamos não aplicando muitos os conceitos. A gente até vê em outros projetos que o pessoal pega bem forte nessa questão, principalmente com o Scrum, Kanban e alguns seguem também o FDD, mas nós nunca conseguimos implementar algum desses no projeto da “mineradora”, não até o momento pelo menos. Em julho nosso projeto vai ter uma virada, a gente vai mudar a estrutura das torres para ter um time de desenvolvimento unificado com uns 20 programadores para atender as nossas demandas e aí vamos conseguir trazer algo mais estruturado, talvez começar a aplicar o Scrum muito mais do que apenas dividir nossos processos em sprints e fazer a nossa gestão ter uma pegada mais ágil também, mais adaptativa para que a gente consiga ser mais assertivo com os nossos orçamentos e prazos.”*

Lucas explica que essa “virada” que o projeto da “mineradora” vai ter parte da Empresa C, para que o projeto se estruture de forma diferente, uma vez que a “mineradora” é um dos maiores clientes que a empresa tem no Brasil e que é necessário organizar melhor como o projeto está definido. O entrevistado ainda aponta que enquanto prestadora de serviços é função da Empresa C entender quais

os GAP's o serviço tem no momento, e estudar questões de melhoria e qualidade do serviço ofertado, já que a parceria ainda tem cerca de 6 anos pela frente e que com a reestruturação que o projeto vai passar, os métodos ágeis vem como auxílio tanto para o desenvolvimento quanto para a gestão.

4.4 Empresa D

A Empresa D foi fundada em 2019 como uma parte do grupo Empresa Principal, criada para trabalhar o desenvolvimento ágil de plataformas digitais voltadas para o setor do agronegócio. A Empresa D conta com 15 colaboradores, o que segundo o SEBRAE, a caracteriza como uma empresa de pequeno porte.

Os principais produtos desenvolvidos pela Empresa D são um aplicativo voltado para o produtor que conta com a conta corrente do cliente, os pagamentos a receber, o que deve ser pago, os grãos comprados, os insumos necessários e tudo mais que o produtor precisa ter em mãos sobre a sua safra para que o controle seja feito de forma centralizada, esse app também conta com a versão web que é o Portal do Produtor.

O outro produto é o software de vendas, um aplicativo CRM voltado para o setor de vendas dos clientes, que conta com lista de preços, simulador de vendas, registro de visitas técnicas e comerciais, plano de cultivo e tudo o que ele acarreta, ou seja, o que precisará ser comprado de sementes, insumos e assim por diante desde o plantio até a colheita.

Além dos dois principais produtos citados, a Empresa D também conta com iniciativas de inteligência artificial, bem como softwares voltados para seguro agrícola e gestão comercial.

4.4.1 Início do Projeto e Documentação

A entrevista com o representante da Empresa D também foi feita de forma online visto a disponibilidade de agenda tanto do entrevistado quanto da entrevistadora. O representante é o diretor de tecnologia André, 42 anos, que trabalha na Empresa D desde a sua fundação além de já estar no grupo Empresa Principal há 9 anos dos seus 25 anos de experiência na área.

Iniciando a conversa, o diretor começa explicando como funciona a Empresa Principal, empresa da qual a Empresa D faz parte. O Empresa Principal é uma empresa tradicional do ramo do agronegócio e é uma distribuidora de insumos agrícolas e também uma cerealista que compra e vende grãos. André era gerente da parte de tecnologia na Empresa Principal, responsável por manter e implementar sistemas. Assim quando a ideia da Empresa D foi colocada em prática, passou a CTO (*Chief Technology Officer*) da nova empresa.

Sobre a Empresa D, André explica *“Há três anos nós decidimos montar a Empresa D para trabalhar com algumas tecnologias que a gente não trabalhava na Empresa Principal, como por exemplo Java, com algumas plataformas de JavaScript e principalmente com plataformas para desenvolver aplicativos mobile que a Empresa Principal até tinha alguns, mas comprados de terceiros e nenhum tinha um fit muito alto para o negócio.”* André também explica que a Empresa Conta com dois produtos principais, já citados na apresentação da empresa, que são aplicativo para o produtor rural e o de vendas, pensados para duas áreas distintas do agro, mas que se complementam, visando atender tanto o produtor em si quanto empresas de porte maior. Sobre esses dois produtos o diretor comenta, *“Esses dois produtos em específico são por encomenda porque a gente conversa continuamente com a área de negócios para entender o que a gente vai colocar nas próximas sprints. Temos projetos e longo e curto prazo, esses de curto prazo as sprints são semana a semana mesmo em que cada semanada é diferente, podendo mudar de acordo com a priorização.”*

Sobre os processos padrão para o início do projeto André comenta que os processos dependem do cliente e do tamanho do projeto e nesse ponto começa-se a identificar o conhecimento sobre os métodos ágeis por parte do entrevistado, pois André aponta um pequeno problema na utilização dessas metodologias, *“Depende muito do tamanho do projeto, para projetos menores a gente procura trabalhar só com a Mock, que são essas telas tentando simular a experiência do usuário. Para projetos maiores, temos outros artefatos e metodologias envolvidas, temos que ter uma agenda de prazos, e aí temos uma grande dificuldade dos métodos ágeis, porque partindo da premissa que os métodos ágeis têm pouca documentação e de que eu aceito mudanças e isso é algo bem-visto dentro dessa metodologia, como eu vou trabalhar prazos nesse processo? Se for algo de curto prazo a gente distribui as sprints, de 1 a 4 semanas, mais que isso também já é demais, e aí eu consigo*

trabalhar com algo de curto horizonte. Agora, quando chega a diretoria e o cliente querendo saber o prazo total do projeto, já temos mais dificuldade em dar essa data, porque o ambiente é mutável e dinâmico.” Sobre isso, André ainda comenta que se começa traçar uma linha entre trabalhar com métodos ágeis e acabar transformando o projeto em um modelo Cascata, *“Trabalhando com métodos ágeis como a gente faz, não temos essa de “data final”, temos datas de entregas e essas entregas a gente consegue ir seguindo o cronograma, mas quando o cliente chega e me fala, quero esse projeto fechado em 1 ano, por exemplo. Isso já não é mais método ágil, é Waterfall, você vai entregando partes “prontas” e cascadeando o processo. Então assim, tentamos não trabalhar projetos de longo prazo para que a gente consiga ter o controle do que conseguimos desenvolver.”*

Partindo para o ponto de requisitos André explica que o cliente geralmente tem algo em mente, porém a ideia acaba sendo um pouco rasa, e para que haja conhecimento o suficiente para desenvolver algo funcional para o usuário a Empresa D usa uma metodologia específica para tal *“Os nossos clientes, vou dar um exemplo, o departamento de marketing da Empresa Principal, eles chegam com alguns requisitos, mas normalmente é algo mais raso então precisamos de algo mais complexo e com mais detalhes. Precisamos de uma metodologia, antigamente a gente chamava de engenharia de software ou engenharia de requisitos, trabalhando com o que a gente chama de UX Design (User Experience Design), que é a etapa de entender a experiência do usuário, desenhar, estudar e pesquisar sobre, já fizemos até mesmo pesquisas com produtores agrícolas, time de vendas etc. para entender realmente o que eles precisam e não só pegar requisitos e sair desenvolvendo.”* Sobre o assunto, Pereira (2018), explica que a experiência do usuário (UX), atinge todos os aspectos da iteração que o consumidor pode ter com uma marca e vai desde o primeiro contato até a aquisição dos produtos. O autor cita que grandes marcas já perceberam a necessidade de colocar o usuário no centro de seu desenvolvimento para melhorar percepção sobre o produto além de garantir bom retorno financeiro.

Assim, o diretor de tecnologia esclarece que a empresa recebe os requisitos, mas sempre procura casá-los com conhecimento na experiência do usuário unindo aos desdobramentos que aquele requisito pode trazer para a equipe transformando-os em especificações técnicas. É nesse ponto que o tema documentação entra na conversa. André explica que as especificações técnicas são definidas, mas não são

amplamente documentadas, *“Por mais que a gente não escreva essas especificações técnicas de forma longa etc. A gente sempre procura tê-las em registro, mas não priorizamos a descrição super detalhada, porque, seguindo o Manifesto Ágil, acho que você já leu sobre isso, não é que possa ter documentação é que a gente prioriza mais o desenvolvimento do código do que descrever tudo em documentações.”* Sendo assim, André explica que há documentação padrão, para fins de fechamento de parceira etc., mas que o foco do projeto não é a descrição detalhada em páginas que muitas vezes nem serão lidas, mas sim um código funcional que o cliente enxergue valor.

4.4.2 Participação do Cliente

Na questão de participação do cliente, André começa explicando a forma com que os clientes chegam Empresa D, *“No nosso caso não tem uma abertura para o mercado, são apenas clientes do próprio grupo. Então temos a Empresa Principal como o principal, a Empresa de Sementes, uma empresa de seguros, uma portadora de insumos da China. São basicamente todas as empresas do grupo, então não temos um trabalho comercial. A Farmbtis não tem nem site para receber encomendas, focamos nesses clientes que estão no grupo.”* Nesse ponto a Empresa D se assemelha a Empresa A, já apresentada nessa pesquisa, pois ambas trabalham atendendo clientes vindos de um cliente maior em específico, com a diferença que a Empresa A não trabalha apenas com clientes vindos dessa área, mas também abre portas para demais empresas.

Outro ponto de semelhança entre as duas, e com a outras empresas estudadas, é que a Empresa D também trabalha com pontos focais dentro do cliente, porém com a diferença é que a Empresa D procura trabalhar com mais de um *focal point* dentro da Empresa Cliente, *“Procuramos trabalhar com pontos focais. Toda demanda temos pessoas fixas para que a gente mantenha a conversa e o fluxo de informações seja feito de uma forma mais fácil. Sobre questões de preço, mantemos um ponto no marketing, sobre questões mais ligadas ao produtor rural, partimos para alguém de outra área e assim por diante. Estamos conversando semanalmente com o cliente.”*

O contato e troca de informações entre Empresa D e cliente ocorre semanalmente devido a outra questão que será abordada no tópico seguinte, as

entregas. A Empresa D procura trabalhar com entregas semanais e quinzenais, o que torna a presença e participação do cliente constante no que diz respeito ao desenvolvimento, uma vez que toda semana haverá uma reunião para apresentação do que a sprint da semana tinha por objetivo.

4.4.3 Equipe, Desenvolvimento e Entrega

Começando pela equipe do projeto, semelhante a Empresa C, a Empresa D procura alocar os desenvolvedores de acordo com o background de cada um, para que o desenvolvimento ocorra de forma fluida, porém essa divisão é feita de forma dinâmica e com finalidade de otimizar o processo, *“As equipes são dinâmicas, ter pessoas fixas que trabalham só com um produto, até tenho, mas no geral a equipe é definida a cada projeto de acordo com os conhecimentos de cada um. Temos o pessoal de back-end que é mais voltado para Java assim como temos o pessoal de front-end indo para o lado do Java Script, temos o pessoal do aplicativo Flutter etc. Então de acordo com o entendimento do projeto a gente observa a demanda e aloca o time.”*

André ainda explica que devido a pandemia equipe se adaptou para o modelo híbrido o de trabalha, assim a equipe não fica toda no escritório, o que traz adaptações para reuniões que geralmente são realizadas através do Microsoft Teams, *“A equipe não fica toda no mesmo lugar, tem gente que vem mais para o escritório, tem gente que fica mais em casa, desde a pandemia a gente adotou o modelo híbrido o, então não temos uma regra de que todos devem vir. Pouco a pouco com a amenização da pandemia a gente deve trazer mais as pessoas para o escritório, mas dificilmente vamos ter aquela regra de que todos devem estar aqui. Então a comunicação é feita pelo Microsoft Teams mesmo, marcamos as reuniões e vamos nos adaptando. As vezes temos apenas duas pessoas de casa, então começamos a reunião, colocamos no viva voz e assim vamos.”*

Seguindo para o ponto do desenvolvimento, o entrevistado explica que a arte de UX acontece como primeiro passo para o desenvolvimento do produto, *“Procuramos muito desenhar as telas dos produtos e não só desenho o photoshop em si, usamos o Adobe XD e outras ferramentas para ir desenhando a experiencia do usuário clicando nas telas, entrando em outras páginas, simulando como o cliente vai receber e usar o nosso produto para que a gente possa refinar melhor os*

requisitos. Então primeiro a gente tem todo esse trabalho de UX para depois passar para o time de desenvolvimento propriamente dito.”

André ainda explica que não há um processo padrão para o desenvolvimento, mas que algumas etapas acabam sendo seguidas com algumas variações de acordo com o tipo do projeto. Para a organização das tarefas a Empresa D usa o software Gira, que é uma plataforma de gestão de tarefas e desenvolvimento feito justamente para trabalho em projetos ágeis, em que há espaço para que todas as tarefas contidas dentro daquela sprint sejam cadastradas e divididas entre os membros da equipe. Com o cadastro das tarefas concluído a equipe começa a dividir quem fará o que, lembrando que há desenvolvedores de front-end e back-end o que por si só já difere quem realizará cada *task*. *Assim, com a separação de atividades e desenvolvimento inicial do produto a equipe passa para outra etapa que são os testes, “Antes de colocar o produto no ar, sempre temos os testes unitários, automatizados. Temos também o code review, ou seja, se eu desenvolvi outra pessoa vai revisar o meu código, depois de aprovado esse código vai para homologação, em um ambiente específico para isso. Então, nossas etapas seguem essa sequência, fazemos os testes, revisamos o código e enviamos para esse ambiente de homologação em que é feito um teste funcional, e só depois de aprovado posso mandar o produto para a produção real.”*

E assim chega-se ao ponto das entregas. André comenta que a equipe procura fazer de forma semanal, com *sprints* mais curtas para que o cliente entenda como o produto está ficando e para que possíveis mudanças nos requisitos aconteçam no fim de cada sprint e não no meio, o que pode acontecer quando entregas são estipuladas com prazos mais longos. Ainda sobre mudança nos requisitos o diretor comenta, *“Se está no meio de uma sprint e algo muda, acaba sendo ruim, não é um negócio super bem aceito, porque você está no meio de uma sprint que geralmente é curta e se você precisar mudar algo no meio atrapalha o trabalho do time e o planejamento feito. Nesse caso tentamos segurar e jogar para a próxima sprint. No caso de ser um bug, a gente para e corrige, mas se for um erro na especificação que inicialmente o cliente achou que queria algo, mas no final era outra coisa, sempre tentamos alocar na próxima sprint para não atrapalhar o trabalho iniciado. Se terminamos uma sprint, entregamos e o cliente sugere mudanças, aí não tem problema algum, mudou a gente se adapta para que na próxima a gente já comece trabalhando nessa mudança.”*

4.4.4 Métodos Ágeis

Chegando ao ponto principal da pesquisa, a utilização dos métodos ágeis, André demonstra certo conhecimento sobre o assunto, principalmente no que diz respeito ao Manifesto Ágil e suas considerações sobre como um projeto de desenvolvimento de software deveria ser. Saindo um pouco do roteiro estipulado e para entender melhor sobre o conhecimento nesse assunto, André é perguntado sobre o que ele vivencia do Manifesto Ágil no seu dia a dia e responde, *“Os conceitos do manifesto a gente segue todos, vou até abrir aqui (na internet) para ler de novo, “indivíduos e interações mais que processos e ferramentas”, sim, seguimos, no Scrum por exemplo, nós temos as Scrum Dailys, a gente vive com reunião diária, não muito longas, mas nossas iterações são muito importantes, não é como em um negócio que eu trabalhei anos atrás em uma fábrica de softwares, que a gente recebia a especificação e ficava lá fazendo o código. “Software em funcionamento mais que documentação abrangente”, sem dúvidas, já até comentamos sobre isso, na nossa visão uma documentação é um código bem-feito, bem estruturado, muito melhor que você escrever uma documentação que ninguém vai ler. “Colaboração com cliente mais que negociação e contratos”, sim, sem dúvidas, e por fim “Responder a mudanças mais que seguir um plano”, acabamos de comentar sobre isso também, que é aí que mora uma dificuldade, queremos ser ágeis para responder a mudanças, mas o cliente em si quer um plano com início, meio e fim, orçamento exato e aí mata o “ágil”. E ainda complementa, “As pessoas costumam romantizar os métodos ágeis achando que só tem vantagem, mas de vez em quando você precisa ter prazo, escopo e custo bem declarados, casando um pouco com as metodologias do PMBOK, mas quando você casa esses métodos você acaba perdendo a agilidade. Quero fazer um projeto estilo Cascata, mas com reuniões todo dia como faz o Scrum... No fim você pode chamar como quiser, mas Scrum não acaba sendo.”*

Com as considerações feitas com relação ao Manifesto, André explica que sobre os métodos ágeis propriamente ditos, o único que a empresa procura seguir mais a risca é o *Scrum*, mas que semelhanças com *XP* e *Kanban* também são encontradas, *“A gente procura seguir o Manifesto sempre. O Scrum seguimos no geral, procuramos seguir os artefatos, reuniões, temos reuniões de planejamento,*

temos as Dailys, e em alguns casos temos a review da sprint, mas não sempre. Então seguimos o Scrum de maneira geral que é o mais aderente a nossa realidade. Sobre a XP temos algumas características, como a parte de revisão de código que se aproxima um pouco, o código é universal, então todos temos acesso, nosso pessoal trabalha bem integrado, então assim, temos características, mas não é algo implementado. E como kanban a gente usa o Gira, sempre colocamos nossas tasks lá então isso se torna muito visual e ajuda bastante no nosso desenvolvimento.”

Com um tempo sobrando para o final da entrevista e devido aos anos de experiência no assunto, André é perguntado sobre o que ele enxerga de diferença entre a utilização dos métodos ágeis e pesados, e qual forma de trabalho ele preferiu trabalhar, *“Na minha visão os métodos ágeis são melhores porque proporcionam entregas contínuas, mas depende do cliente também. Se você tem um cliente interno, ele demanda informações de escopo, prazo e preço muito claras, o PMBOK chama de tríplice restrição, inclusive. Esses pontos acabam dificultando que o projeto seja feito de forma ágil. Se me perguntarem que traz mais resultado e o que eu prefiro, claramente a resposta será sobre métodos ágeis e os conceitos do manifesto. Para projetos muito grandes em empresas muito grandes, com as quais eu já trabalhei é muito difícil conseguir encaixar os conceitos ágeis, principalmente na questão das entregas, você até pode entregar semanalmente, mas a visibilidade de prazo etc. é muito importante e é necessário ser bem claro quanto a isso. Então claro que depende do tipo do projeto que você está inserido, mas no geral os métodos ágeis proporcionam um ambiente mais próximo do que a gente precisa viver no dia a dia, algo mais dinâmico e que nos permite mudar de acordo com as necessidades que vão surgindo.”*

4.5 Compilado dos dados das empresas estudadas

Tabela 4 - Resumo dos resultados obtidos

Empresa	Empresa A	Empresa B	Empresa C	Empresa D
----------------	------------------	------------------	------------------	------------------

Início do projeto e documentação	Reunião técnica para análise do perfil do cliente; Auxílio ao cliente quanto aos requisitos; Documentação simplista;	Reunião de Aderência para análise do perfil do cliente; Requisitos definidos; Documentação padrão	Início do projeto feito de forma informal; Requisitos vagos ou inexistentes Sem documentação	Reunião com o cliente para análise do produto a ser desenvolvido; Requisitos conhecidos, mas não tão claros; Foco no desenvolvimento acima da documentação
Participação do cliente	Adoção de um ponto focal nas empresas clientes	Adoção de um ponto focal nas empresas clientes	Adoção de um ponto focal nas empresas clientes	Adoção de um ponto focal nas empresas clientes
Equipe, Desenvolvimento e Entregas	Equipe alocada de acordo com o background de cada desenvolvedor; Desenvolvimento linear e conhecido; Entregas em sprints definidas no escopo inicial	Equipe alocada de acordo com o background de cada desenvolvedor; Desenvolvimento com auxílio do software desenvolvido pela empresa; Entregas imediatas ou feitas de forma rápida devido ao produto já estar acabado	Equipe alocada de acordo com o background de cada desenvolvedor; Desenvolvimento por partes e com acompanhamento do setor em questão Entregas em sprints acordadas nas reuniões de entrega	Equipe alocada de acordo com o background de cada desenvolvedor; Desenvolvimento de Mocks ou telas funcionais a partir do UX Design; Entregas semanais ou quinzenais de acordo com as sprints
Métodos Ágeis	<i>Scrum e Kanban</i>	Não há	Não há	<i>Scrum e Kanban</i>

5. ANÁLISE DOS RESULTADOS

5.1 Empresa A

5.1.1 Início do Projeto e Documentação

Com as respostas obtidas na entrevista com o representante da empresa, pode-se notar que o *know how* que a Empresa A possui dentro do ramo em que está atuando é grande e foi consolidado com as mudanças que a empresa passou durante os anos. A Empresa A tem uma metodologia definida para o desenvolvimento dos produtos que vem justamente a partir do *know how* apresentado pelo entrevistado que possivelmente é compartilhado com outros membros da equipe, tanto analistas quanto gestão.

Como a empresa possui um produto acabado e as mudanças são desenvolvimento de módulos específicos ou outras modificações mais simples, o *know how* da empresa trabalha para auxiliar o cliente a entender o que ele realmente precisa para o produto que será desenvolvido norteando o processo decisório. Assim, a Empresa A consegue utilizar da metodologia desenvolvida por eles para análise técnica e coleta de dados junto ao cliente. Pode-se perceber que a empresa tem um método que se encaixa para todos os possíveis clientes que deverá ser adaptado apenas para situações particulares.

No que consta sobre documentação a Empresa A ainda se encontra em um estágio simplista. Os documentos gerados são basicamente os mesmos para todos os clientes, o que de certa forma traz um padrão para empresa, porém a documentação em si é básica e precisa ser melhorada. Ponto que até mesmo Bruno comenta dizendo que a empresa notou essa deficiência e está tentando melhorar.

Na análise do Manifesto e seus conceitos, pode-se notar que existe a priorização de um trabalho bem-feito antes de uma documentação detalhada e extensa. A Empresa A tem total condição de se adaptar a esse quesito. Como comentado, a empresa entende muito do que está fazendo e até mesmo ajuda o cliente no entendimento do que ele precisará para seu produto, assim, pode-se adaptar a documentação que a empresa possui, que já é simples, para que continue

sendo algo conciso, porém focado no que o cliente apresentou e o que será feito. Simples, conciso e assertivo, pontos que o Manifesto Ágil aponta para esse processo.

Um ponto positivo que a Empresa A possui com relação a documentação é que o escopo inicial gerado tende a ser seguido à risca, ou seja, tudo o que for acordado inicialmente com o cliente será seguido e possíveis adaptações serão realizadas se o tempo disponível for suficiente ou se o cliente estiver disposto a aumentar a quantidade de horas do projeto. Isso traz segurança tanto para a equipe de desenvolvimento quanto para o cliente que sabe, desde o fechamento do negócio, as fases que o desenvolvimento vai passar e o que será entregue.

5.1.2 Participação do Cliente

No que consta sobre participação do cliente a Empresa A se assemelha muito com as outras empresas estudadas nessa pesquisa. Nota-se que todas as empresas desse estudo adotam o ponto focal dentro do cliente para contato e repasse de informações. Isso é positivo para ambas partes e importante para o andamento do projeto.

A utilização de pontos focais, para o cliente, traz a percepção de que a Empresa Contratada está presente e preocupada com o desenvolvimento correto do que foi acordado, e como citado pelo entrevistado, deixa uma sensação maior de presença e segurança para o cliente que sabe que qualquer dúvida ou problema será relatado a pessoa responsável pelo contato entre ambas as partes o que faz com que o cliente acompanhe o processo e saiba das possíveis dificuldades encontradas pela Empresa A, em que ponto o desenvolvimento se encontra e se a adequação do software vai ser funcional ou não.

Para a Empresa A, a adoção de um ponto focal faz com que o canal de comunicação com o cliente seja constante. As informações e mudanças são repassadas com facilidade, o que aumenta o valor do produto que eles estão desenvolvendo, porque trazer o cliente para dentro de seu processo faz com que seu desenvolvimento seja feito de forma sustentável e acompanhado, fazendo com que empresa saiba que as entregas serão assertivas trazendo valor ao que foi desenvolvido.

Ambos os pontos citados, desenvolvimento sustentável e entregas contínuas, que são encontrados no processo da Empresa A se assemelham ao Manifesto Ágil nos princípios 1 e 8, que falam respectivamente “*nossa maior prioridade é satisfazer o cliente através da entrega contínua e adiantada de software com valor agregado*” e “*os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.*”

Assim, o método adotado pela Empresa A para trazer o cliente mais próximo das etapas que serão realizadas, mesmo que inconscientemente faz parte de princípios descritos no Manifesto e ainda de um de seus conceitos que diz que a colaboração com o cliente deve ser maior que negociação e contratos, o que também pode se encaixar no que foi descrito na seção sobre documentação.

5.1.3 Equipe, Desenvolvimento e Entrega

Sobre a equipe, o *know how* da Empresa A traz novamente mais facilidade no processo. A empresa entende que analistas juniores precisam de mais tempo para entender e começar a desenvolver do que analistas seniores, assim quando o projeto necessita de mais de um analista a gestão sabe que a quantidade de horas deverá ser revisada junto a equipe e cliente para que o fechamento desse acordo seja feito de forma mais assertiva. Nesse ponto o que se assemelha ao Manifesto é que a gestão deve confiar que aquele analista conseguirá produzir o que o projeto precisa além de fornecer o suporte necessário para que o trabalho seja realizado.

Quando mais de um analista é alocado no projeto, a equipe decide quem atacará cada frente, levando em consideração o *background* de cada um, ponto citado pelo décimo primeiro princípio do Manifesto que diz “*as melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.*”

Para finalizar o tema sobre equipes também é importante citar que a equipe da Empresa A é bem aberta a receber mudanças, o que se diferencia de outras empresas estudadas. Isso se deve por outro método utilizado pela Empresa A, em que as mudanças não vão diretamente para a equipe de desenvolvimento e sim para o gerente de projetos que vai analisar juntamente ao cliente se aquela mudança é válida para então passar a informação para os membros envolvidos. Isso gera menos estresse para quem está desenvolvendo, uma vez que se a mudança

não for validada a equipe até ficará sabendo, mas não vai impactar o trabalho desenvolvido e caso a mudança seja aceita, o trabalho entre equipe e gestão será para que seja alocada de forma a não atrapalhar o desenvolvimento da sprint em questão.

Sobre o desenvolvimento a Empresa A traz consigo muitas características apontadas pelo Manifesto, mesmo que a empresa não se dê conta disso. Começando pela comunicação face a face entre desenvolvedores e gestão, um dos pontos importantes levantados pelo Manifesto. A Empresa A se enquadra como uma empresa de médio porte, mas seu escritório tem um modelo aberto para que todos os membros da equipe possam se sentar próximos e uma conversa seja estabelecida durante o desenvolvimento. Isso também torna o ambiente mais agradável fazendo com que os colaboradores se sintam mais motivados e confortáveis no local de trabalho, outro ponto que pode ser encontrado no Manifesto.

A empresa está tempo o suficiente no ramo para entender facilmente as fases em que o desenvolvimento irá passar e a equipe consegue definir quem cuidará de cada parte. Os analistas que cuidam de *front-end* conversam entre si para chegar a um ponto de acordo, o programa utilizado para montar tabelas, gráficos e outros componentes que serão visíveis ao cliente é o *Qlik Sense*, assim como os analistas que cuidam do *back-end* que trabalham com *Qlik View* e outras linguagens. O método da empresa é simples, mas consolidado, o que faz com que a autonomia da equipe seja maior e dessa forma, os membros da equipe trabalham em conjunto para entregar a sprint.

O desenvolvimento e as entregas são ligados diretamente. Ainda no início do processo a Empresa A procura definir as sprints e o que cada uma vai entregar para que isto conte já na documentação inicial, assim, o processo de maturação de cada sprint é bem conhecido pelos membros da equipe o que torna o processo de desenvolvimento definido e podendo ser seguido como acordado inicialmente.

No que consta sobre entregas, Empresa A tenta seguir as sprints acordadas do começo ao fim, assim as entregas são feitas parcialmente seguindo o cronograma pré-estabelecido, dessa forma o primeiro ponto abordado no Manifesto é seguido pela empresa, que faz suas entregas serem contínuas. Outro ponto que a EMPRESA A leva muito em consideração e que podemos encontrar no manifesto é que a cada entrega o software deve ser funcional para que a avaliação do cliente seja assertiva e que as mudanças encontradas sejam percebidas na reunião de

entrega da sprint para que a solicitação seja atendida na próxima fase do projeto caso necessária.

5.1.4 Métodos Ágeis

Sobre um dos pontos principais dessa pesquisa, os métodos ágeis, a Empresa A se encontra em um momento interessante. Durante a conversa, o entrevistado comentou que a empresa está passando por um processo de transição na gestão de projetos, saindo de não ter uma metodologia, para tentar encaixar algumas iniciativas ágeis no processo.

Nesse ponto, a empresa encontra dificuldades no quesito cultural para a adoção de métodos como *Scrum* e *Kanban*. A equipe de analistas vem crescendo com o aumento da demanda e muitos deles não sabem o que são os métodos ágeis e no que eles poderiam interferir no trabalho a ser realizado. No entanto a nova gestão de projeto vem trabalhando para que a cultura da empresa mude com o passar do tempo.

Os dois métodos que Empresa A começou a adaptar para a sua realidade, como citado, são *Scrum* e *Kanban*. Na parte relacionada ao *Scrum*, o que a empresa está adotando no momento é a divisão das etapas do processo nas sprints, para facilitar o desenvolvimento e controle do que será entregue em cada etapa. No entanto, outros pontos abordados por esse método ainda não fazem parte do dia a dia da empresa, como as reuniões diárias que a *Scrum* utiliza para que os membros da equipe fiquem alinhados sobre o que foi desenvolvendo até aquele momento e o que falta ser feito. Esse é um ponto que poderia ser adotado e que seria facilmente incorporado na empresa, visto que o próprio ambiente é propício para a realização. Como os membros da equipe tem a possibilidade de se sentarem juntos no mesmo ambiente, os 15 minutos referidos como suficiente pelo método poderiam ser aplicado ainda no começo do dia, quando todos chegam, para que as informações sejam compartilhadas e a partir daí o dia de desenvolvimento tenha início.

Dada a realidade da empresa, pontos como *Product Owner* e *Scrum Master* talvez não sejam aplicáveis ao modelo de trabalho que a Empresa A possui hoje. O *Product Owner* acaba sendo próprio cliente que entrou em contato para ter um orçamento ou o escopo do projeto requisitado e as funções que seriam de um Scrum master, como conhecimento do método e resolução de conflito acaba sendo

englobado pelo *know how* possuído pela equipe. No entanto dadas as conversas obtidas para o estudo e demais pesquisas feitas, é difícil encontrar uma empresa que tenha todos os pontos do *Scrum* totalmente definidos, o que se acaba se encontrando são adaptações do modelo para a realidade da empresa em questão, ponto que Empresa A vem atendendo e incrementando de acordo com as percepções de necessidade que vem surgindo e que o *Scrum* pode suprir.

Sobre o *Kanban*, pode-se dizer que foi o método de mais fácil incorporação que a Empresa Adotou, porque não depende tanto da adaptabilidade dos novos membros a uma forma de trabalho como o *Scrum*. A Empresa A possui quadros de *Kanban* físicos de acordo com cada projeto, colocados nas paredes do ambiente aberto que a empresa possui para que a visibilidade do processo seja conhecida por todos e que caso algum membro da equipe encontre dificuldade em desenvolver determinada tarefa, outra pessoa possa o ajudar de acordo com as tarefas da *sprint* delegas a ele e o prazo de conclusão remanescente. Sobre isso, o entrevistado comentou que a adoção do *Kanban* em conjunto com o *know how* da empresa possui ajudou a organizar as prints de maneira sustentável de forma que nenhum analista fique sobrecarregado. A equipe procura destrinchar as tarefas da *sprint* em pequenas atividades que serão divididas em comum acordo entre os membros e colocas de cores diferentes no *kanban* de acordo com o que a tarefa se refere. Dessa forma, a equipe tem visão do processo e do progresso que vem tendo de acordo com o que foi terminado e retirado do *kanban*.

Além desses dois métodos que aos poucos vem sendo incorporados na empresa, a Empresa A também possui traços de outra metodologia no seu desenvolvimento, mesmo que essa não seja conhecida pela empresa, o FDD (desenvolvimento dirigido por funcionalidade). Como em muitos casos o cliente necessita apenas que um novo módulo específico seja incorporado ao Software de Gestão, como financeiro, vendas, cereais, sementes etc, por exemplo, a equipe acaba desenvolvendo o módulo de acordo com a nova funcionalidade requerida, traços muito semelhantes ao do FDD, uma vez que, de acordo com o entrevistado quando algum módulo especial é solicitado a equipe que irá desenvolvê-lo deve construir a lista de tudo o que o módulo deve conter, o que se assemelha a lista de funcionalidades gerada pelo modelo em questão. Assim, etapas como DPF (detalhar por funcionalidade) e CPF (construir por funcionalidade) são adotados pela equipe

de módulos especiais e direcionam o desenvolvimento dirigido para aquela nova característica que o software deverá possuir.

5.2 Empresa B

5.2.1 Início do Projeto e Documentação

A Empresa B é uma Empresa Com resultados e análise sucintos. Nesses mais de 50 anos no mercado a empresa passou do framework ao software e vivenciou a construção desse novo modelo. Dessa forma, diferente das outras empresas estudadas a Empresa B possui um método próprio para cada um dos pontos analisados nessa pesquisa.

A empresa está presente em diferentes ramos de atuação e já tem produtos prontos para supriar as necessidades de seus clientes, questões como documentação e inicialização do projeto são tidos como simples pela percepção de Claudemir, o tempo de experiência não só da empresa, mas também do entrevistado faz com que todas as perguntas tivessem respostas simples sendo a entrevista com menor tempo de duração.

Sobre o início do projeto, há a percepção de que o cliente chega até a Empresa B devido a confiança que a empresa passa devido a marca estabelecida no mercado. Nesse ponto o que se pode encontrar de semelhanças com o Manifesto é relaciona aos seus conceitos e não princípios. Essa visão que a Empresa B construiu ao longo dos anos faz com que a colaboração com o cliente seja de fato maior que negociações e contratos, visto que a empresa domina todos os aspectos envolvidos e até mesmo negociações elaboração documentos sejam tarefas facilmente resolvidas pela equipe de negócios.

A documentação utilizada pela Empresa B já está consolidada e tem mudanças básicas de acordo com a necessidade apresentada, assim como a Empresa A, a empresa já possui documentos padrão que vão sendo alterados de acordo com o cliente. O que diferencia a documentação de cliente para cliente é o questionário que é aplicado na empresa que aborda pontos específicos de cada ramo, para que a Empresa B entenda qual estágio a Empresa Cliente se encontra e como o software deverá ser implementado e se realmente irá atender as necessidades levantadas.

Outra coisa que se pode perceber é que a empresa preza muito por implementar um software funcional e treinar os envolvidos na Empresa Cliente do que realmente focar na documentação do processo, que como já citado, é padrão, outro ponto que pode ser observado nos conceitos do Manifesto.

5.2.2 Participação do Cliente

Sobre a participação do cliente, Claudemir deixa claro que um dos motivos dos clientes irem até a Empresa B é pela consciência de que a empresa sabe o que está fazendo e que irá entregar um software com valor para o cliente. Ponto que pode ser observado no primeiro princípio do manifesto que aponta que a prioridade da empresa é satisfazer o cliente com entregas adiantadas com valor agregado. Assim, o cliente entra em contato com a Empresa B, que apresentará os produtos que a empresa possui e caso um deles atenda a necessidade que o cliente busca suprir a Empresa B se encarrega de todo o processo até que a entrega e implementação possa ser feita.

Assim como as outras empresas estudadas, a Empresa B também adota o método de ter um *focal point* dentro da Empresa Cliente. Logo, quando a Empresa B faz as pequenas alterações necessárias ou quando já tem a equipe de implementação preparada para ir a Empresa Cliente, o primeiro a saber é o *focal point* que juntamente com a gestão da Empresa Agendará a data para entrega e implementação do software na empresa.

Além dos pontos citados, não se encontram maiores semelhanças com os princípios e conceitos do Manifesto Ágil, até mesmo porque a Empresa B desenvolveu alguns de seus métodos antes da elaboração do Manifesto que ocorreu no início dos anos 2000, quando a empresa já possuía 30 anos de experiência.

5.2.3 Equipe, Desenvolvimento e Entrega

Como a Empresa B possui as chamadas “concessionárias” espalhas pelo país, muitas vezes a equipe que vai desenvolver ou atualizar o software em questão não está na mesma cidade ou até mesmo no mesmo estado. Porém para suprir esse afastamento dos membros, a Empresa B desenvolveu uma plataforma própria para a comunicação, um CRM, em que não só a comunicação é feita, mas também

todas as atividades são registradas e tem seu progresso marcado de acordo com o decorrer do desenvolvimento. Nesse ponto pode-se dizer que a Empresa B realmente vive a tecnologia e usufrui de seus benefícios para que não apenas desenvolvedores, mas analistas de negócios, pessoas do jurídico e financeiro fiquem sempre alinhadas sobre os procedimentos e sempre pensando em formas de otimizar o trabalho. Assim, pode-se encontrar uma semelhança com o último princípio do Manifesto que aponta que a equipe deve sempre refletir sobre como tornar o processo mais eficaz e refina seu comportamento para que isso seja alcançado.

Para o desenvolvimento a Empresa B tem pontos interessantes. A empresa tem um *know how* tão grande do mercado de software que desenvolveu sua própria linguagem de programação além de ter um programa que gera a parte inicial do código para os programadores. Então questões de padrão e estética do código já vem prontos e cabe ao desenvolvedor apenas refinar o código e fazer implementações de comandos específicos para que o módulo seja finalizado ou a atualização do software feita. Dessa forma, a Empresa B se diferencia das outras empresas estudadas porque mesmo que a Empresa A tenha um produto pronto, o desenvolvimento de novos módulos é recorrente, o que não acontece na realidade da Empresa B que tem como maior parte do desenvolvimento a atualização dos softwares que a empresa possui, adequando-os as leis que mudam de tempos em tempos e necessitam de adequação por parte do software. Assim, observa-se que a Empresa B atende ao nono princípio do Manifesto que diz que a excelência técnica e bom design (que nesse caso já está pronto) aumentam a agilidade, o que de fato é percebido no contexto da empresa.

Sobre as entregas, Claudemir aponta na entrevista que muitas vezes essas são feitas de forma imediata, visto que os softwares já estão desenvolvidos e atuantes em muitas empresas, então se a necessidade que o cliente busca suprir está presente dentro do software que a Empresa B já possui, a única coisa a ser agendada é a implementação do mesmo por parte da equipe da empresa e posterior acompanhamento para observar se o software irá rodar como deve. Logo, o primeiro princípio do manifesto que aponta que a prioridade é satisfazer o cliente com entregas contínuas e com valor agregado é novamente observado no processo.

5.2.4 Métodos Ágeis

Assim, sobre os métodos ágeis, as características encontradas na Empresa B são diferentes das encontradas nas outras empresas estudadas. A Empresa B não conta com métodos ágeis em seu desenvolvimento, nem mesmo características de *Scrum*, *OpenUP* ou outros métodos mais utilizados pelo fato de que a maioria dos processos utilizados hoje na empresa foram desenvolvidos internamente para suprir as necessidades da Empresa B ao longo dos anos.

Na entrevista, Claudemir cita que a empresa “sabe tudo sobre métodos ágeis”, o que pode ser dito como verdade visto que “métodos ágeis” não são apenas as metodologias específicas apresentadas nesse trabalho, mas também uma filosofia que pode ser seguida tanto na parte de desenvolvimento quanto na parte de gestão.

O que se pode perceber, é que na Empresa B os métodos ágeis podem ser encontrados na forma com que o trabalho é conduzido, a gestão entende que é necessário se adaptar o mercado as demandas que são mutáveis então a percepção de agilidade é encontrada na forma com que a empresa se adapta as mudanças na economia e mercado e até por isso consegue se sustentar a mais de 50 anos mantendo seus processos e produtos atualizados e requisitados. Dessa forma, como a gestão entende dobre como a adaptabilidade é importante essa percepção é repassada a todos os colaboradores, que se engajam para transformar o processo e adaptar sua forma de trabalho para que tudo seja condizente com a proposta da empresa.

5.3 Empresa C

5.3.1 Início do Projeto e Documentação

A Empresa C possui resultados um pouco diferentes das demais empresas estudadas pois tem foco em apenas um cliente, A “mineradora”. Sendo assim, com um contrato firmado entra as duas empresas pelos próximos 7 anos algumas características encontradas na Empresa B, Empresa A e Empresa D não se encaixam para a realidade da multinacional.

O primeiro ponto de diferença diz respeito a documentação. Como apresentado por Lucas na entrevista a Corporate (nome do serviço de TI prestado a “mineradora”), não possui qualquer documentação com relação aos sistemas desenvolvidos. Nem por parte da Corporate nem por parte da “mineradora”. Levando em conta o contrato prévio entre as duas empresas é até de se esperar que não haja demais documentações, porém isso não quer dizer que seja o indicado.

A Corporate já desenvolveu mais de 60 sistemas para a “mineradora” e continua nesse processo. Seria interessante que a empresa mantivesse registro do que foi feito, seguindo o que o Manifesto Ágil propõe, por exemplo, que é ter uma documentação mesmo que ela não seja o foco do projeto, e sim o desenvolvimento.

O registro do serviço prestado é importante até mesmo para a segurança das duas empresas, e, para a Empresa C, pode ser uma forma de provar e talvez apresentar a futuros clientes o que foi desenvolvido durante o projeto com a “mineradora” como um exemplo do que pode ser feito para o novo cliente.

O início do projeto também ocorre diferente para a multinacional, com o cliente estabelecido, o início dos projetos nascem das necessidades dos diferentes departamentos que a “mineradora” possui, cada departamento tem autonomia para acionar o time de TI e requisitar uma atualização de sistema ou desenvolvimento de algo complementar.

Nesse ponto, pode-se destacar que a Empresa C segue um dos conceitos do Manifesto que fala que a colaboração com o cliente é mais importante que negociações e contratos, uma vez que a equipe de desenvolvimento colabora continuamente com o cliente para entender a necessidade apresentada no primeiro contato entre as partes para determinar o que o programa deverá conter e como as etapas desse desenvolvimento deverão ocorrer. No mais, a Empresa C não tem muitas semelhanças com os princípios citados no Manifesto no que diz respeito a documentação e iniciação do projeto.

5.3.2 Participação do Cliente

Sobre a participação do cliente pode-se notar que a Empresa C procura manter o ponto focal do departamento ou da pessoa do TI da “mineradora” sempre por perto. De acordo com o relato do entrevistado, muitas vezes a descrição do sistema a ser desenvolvido vem apenas com uma linha de referência básica, o que

faz com que a participação do cliente seja ponto chave para a o desenvolvimento de um sistema com as funcionalidades desejadas. Dessa forma, pode-se observar que a Empresa C possui semelhanças com o Manifesto no que diz respeito a prioridade em satisfazer o cliente com entregas contínuas com funcionalidades desenvolvidas e que gerem valor para o cliente, uma vez que em muitos casos, nem mesmo o departamento que pediu o desenvolvimento do sistema sabe qual será o produto desenvolvido em sua totalidade. Assim, se naquela entrega a funcionalidade apresentada for aprovada pelo cliente, gera valor ao que foi desenvolvido.

Nesse ponto também pode-se destacar que o quarto princípio descrito na Manifesto também se aplica aos processos da multinacional, pois nele o Manifesto descreve que pessoas de negócio e desenvolvedores devem trabalhar em conjunto por todo o projeto, o que se encaixa perfeitamente na realidade da empresa, visto que as pessoas de negócio, para esse caso específico, podem ser as que requisitaram que aquele sistema fosse desenvolvido, e é de suma importância a colaboração com o time de desenvolvimento para que o produto entregue esteja dentro das expectativas.

5.3.3 Equipe, Desenvolvimento e Entrega

No que diz respeito a equipe, assim como na Empresa A e na Empresa D, a Empresa C procura alocar seus desenvolvedores de acordo com o background de cada um para que o processo de desenvolvimento seja facilitado para todas as partes. Nesse quesito pode-se dizer que a Empresa C atender parcialmente ao sexto princípio do manifesto que diz que a melhor forma de se transmitir informações é com a conversa face a face, o que é muito prezado pela equipe da Corporate, mesmo que esse contato seja feito via Teams.

Para Empresa C o bom desenvolvimento do sistema depende diretamente da boa comunicação entre os membros da equipe, pois como citado na seção anterior e durante os resultados já apresentados, muitas vezes a equipe que deverá se desdobrar para entender as necessidades do cliente e descobrir por si só como aquele sistema será desenvolvido, visto que os requisitos iniciais são quase nulos e existem mudanças constantes sobre a percepção do sistema por parte do cliente. Logo pode-se dizer que a equipe da Empresa C também atende ao décimo segundo princípio descrito no Manifesto que relata sobre como a equipe deve refletir como se

tornar mais eficaz e ajustar seu comportamento de acordo, uma vez que a cada reunião com o cliente a equipe deve se adaptar ao que foi requisitado e ser eficaz o suficiente para suprir o que for pedido para a próxima entrega.

No que consta sobre o desenvolvimento, a Empresa C também trouxe resultados diferentes das demais. O entrevistado relatou que a empresa segue as etapas “básicas” para que o desenvolvimento seja completo que seriam análise, desenvolvimento, teste e implementação, porém o que a diferencia das demais são os requisitos. A Corporate trabalha com requisitos enxutos e quase inexistentes e cabe a equipe definir o que e como será desenvolvido a partir da ideia simplista que chega até eles. Assim, a empresa não possui um método padrão para seguir as etapas de desenvolvimento certas porque o decorrer do projeto vai depender diretamente do que foi requisitado e como foi requisitado. Um projeto em que o departamento sabe melhor o que precisará ser realizado e com maiores informações a respeito demanda menor tempo de análise, tempo que pode ser empregado na função de desenvolver o sistema. A requisição de um sistema sem muitas informações a respeito faz com que a equipe tenha que despender mais tempo entendendo o que será feito e pesquisando mais sobre o assunto, fazendo com o que o tempo estante acordado tenha que ser dividido entre desenvolvimento, entregas, ajuste nas mudanças e implementação.

O que traz outro elemento importante. Das empresas entrevistadas e pelos relatos conseguidos, a Empresa C é a empresa que mais faz ajustes no que está sendo desenvolvido. É claro que o comparativo não é tão justo uma vez que Empresa B e Empresa A trabalham com produtos prontos e os ajustes necessários vem de acordo com a necessidade do cliente, mas a falta de informações para início do projeto faz com que a cada reunião de entrega a equipe da Empresa C se depare com algum ajuste sendo necessário visto que até mesmo o cliente vai entendendo a forma com que o produto vai ficar de acordo com as apresentações do time. As mudanças são esperadas e o time consegue se adaptar para que elas sejam atendidas, isso leva ao conceito de que a equipe se utiliza das mudanças, mesmo que tardias para gerar vantagem competitiva ao cliente.

Sobre as entregas, a equipe de desenvolvimento da Corporate procura fazê-las constantes com o que já foi desenvolvido justamente por já esperar que mudanças sejam requisitadas. Nesse ponto pode-se destacar que a empresa traz consigo novamente o primeiro princípio do Manifesto, pois as entregas são feitas de

forma contínua visando entregar um sistema com valor agregado para que o cliente analise se a equipe está no caminho certo ou se o que foi desenvolvido não atende necessariamente o que o departamento pensou.

5.3.4 Métodos Ágeis

Após os pontos abordados e com as informações conseguidas na entrevista pode-se notar que no momento a Empresa C não possui aplicação de nenhum método ágil no seu processo de desenvolvimento de softwares e sistemas para a “mineradora”. O entrevistado até demonstrou saber do que se tratam os métodos, mas também concorda que nada está sendo aplicado até então. A esperança do time é que com a virada que o projeto vai passar a partir no mês de julho a equipe consiga se adaptar melhor para enfim conseguir ter alguma metodologia no dia a dia de desenvolvimento.

5.4 Empresa D

5.4.1 Início do Projeto e Documentação

A Empresa D foi a empresa que mais apresentou resultados no que diz respeito sobre conhecimento em métodos ágeis e é a que mais segue os princípios do Manifesto Ágil. Desde o início da entrevista o domínio do entrevistado sobre o assunto se fez claro, o que fez com que os resultados obtidos fossem muito satisfatórios.

Para dar início ao projeto, o cliente do grupo Empresa Principal entra em contato com a empresa para requisitar o desenvolvimento de um sistema ou aplicativo, de acordo com o entrevistado, muitas vezes os requisitos são básicos, o que torna a necessidade de uma análise mais profunda recorrente para o início do projeto. Nesse ponto, a Empresa D segue dois caminhos, o primeiro deles diz respeito a pesquisa de mercado e a aplicação do *UX Design*, que visa entender como será a experiência que o usuário vai ter na utilização daquele produto para que o desenvolvimento se baseie nisso. Para tanto a empresa foca na pesquisa de mercado, com o público-alvo selecionado para coleta de informações e

entendimento do processo. O outro caminho, consiste em analisar os requisitos de acordo com o *know how* que os colaboradores da Empresa D possuem para início da divisão de tarefas em especificações técnicas para o software e partir para o desenvolvimento.

É nesse momento que a documentação começa a ser elaborada. Porém como relatado pelo entrevistado e descrito como um dos conceitos do Manifesto Ágil documentação não é o ponto principal do desenvolvimento da Empresa D, que foca principalmente no desenvolvimento do programada em detrimento de uma série de páginas que podem nem ser lidas.

As especificações técnicas são as primeiras a serem detalhadas no que se tornará o documento formal para que o projeto se inicie. Após validação dessas especificações junto ao cliente a empresa decide se irá partir para a pesquisa de mercado ou se já possui algo semelhante desenvolvido que poderá se basear juntamente com o *know how* que os membros do projeto possuem.

Dessa forma, além do conceito que diz respeito a documentação, pode-se notar que a empresa também possui características voltadas a colaboração com o cliente mais que negociações e contratos, outro ponto descrito nos conceitos do Manifesto, uma vez que o entendimento do processo e a pesquisa de mercado são pontos pensados estrategicamente para gerar vantagem competitiva para o cliente que requisitou o sistema, resultado da colaboração da empresa prestadora do serviço para com o requisitante.

5.4.2 Participação do Cliente

Sobre a participação do cliente, segundo o relato do entrevistado são feitas reuniões semanais para apresentação do que foi desenvolvido na *sprint* em questão. Assim, o cliente acompanha de perto todas as fases do desenvolvimento opinando sobre o que foi desenvolvido até o momento. Desta maneira, a equipe consegue transformar os requisitos básicos recebidos em futuras especificações que poderão ser enquadradas na próxima fase do desenvolvimento.

Nesse ponto, assim como na Empresa C, pode-se destacar as semelhanças com o quarto princípio do Manifesto Ágil, que diz que pessoas de negócio e desenvolvedores precisam trabalhar em conjunto para que o projeto saia como o esperado, o que acontece frequentemente nos projetos da Empresa D que sempre

procura alinhar as necessidades do cliente com o que será desenvolvido nas próximas etapas, sempre analisando o que seria interessante não só para o dono do futuro sistema, como para o consumidor final daquele produto. Esse processo se liga com o oitavo princípio do Manifesto, em que patrocinadores, usuários e desenvolvedores deve manter um ritmo constante proporcionado pelos processos serem realizados de forma ágil.

Assim, quando a realizada a reunião semanal com o cliente, tudo o que aconteceu nas reuniões anteriores devem ter sequência lógica e fazer sentido para que a linha de desenvolvimento seja entendida pela parte de negócios do cliente, mas também quem fará o papel de consumidor.

Além das reuniões semanais, ter um ponto focal também é uma estratégia adotada pela Empresa D, assim como para as outras empresas estudadas. Isso se deve ao fato de que a comunicação também é base para o desenvolvimento na empresa que visa ter um fluxo de informações contínuo com o cliente para que as reuniões sejam claras e objetivas fazendo com que possíveis dúvidas sejam sanadas nas interações com a pessoa escolhida para ser o focal point do projeto. Assim, pode se dizer que a Empresa D também segue o conceito do Manifesto que se refere a indivíduos e interações serem mais relevantes que as ferramentas e processos utilizados.

5.4.3 Equipe, Desenvolvimento e Entrega

A equipe da Empresa D também é dividida de acordo com as aptidões de cada membro alocado no projeto. Assim como nas outras empresas estudadas, o tamanho da equipe é diretamente proporcional ao tamanho do projeto e dos conhecimentos de cada um. Como descrito na seção de início do projeto e documentação, a equipe se reúne para detalhar os requisitos obtidos em especificações técnicas para logo após a divisão das tarefas ser realizada e o projeto ter início. Esse processo se assemelha ao décimo primeiro princípio do Manifesto que aponta que requisitos, *design* e arquiteturas bem definidos se derivam de equipes auto-organizáveis, que conseguem entender o que deverá ser desenvolvido na sprint em questão e se organizam para que de acordo com o *know how* de cada um, as diferentes frentes do projeto sejam atacadas e o proposto para a entrega seja realizado sem atrasos.

A comunicação entre os membros da equipe é feita da forma como descrito no sexto princípio do Manifesto Ágil, que aponta que a melhor forma de troca de informações dentro da equipe é pela conversa cara a cara, mesmo que essa seja realizada via Microsoft Teams. Nesse ponto André comenta que a equipe procura marcar reuniões com todos os membros do time diariamente para que todos estejam alinhados, até mesmo porque a sprints tem prazo de uma semana, no máximo quinze dias na Empresa D e com entregas feitas de forma tão frequente é extremamente importante que todos saibam em que ponto o projeto se encontra até mesmo para oferta auxílio de partes que estejam atrasadas.

No que conta sobre desenvolvimento não há processo padrão, o que determina como será realizado é essencialmente a divisão das tarefas visando entregar um sistema em funcionamento em intervalos curtos de tempo para avaliação do cliente, ponto encontrado no terceiro princípio do Manifesto Ágil. Além disso, para a Empresa D há outro princípio se encaixa quase que perfeitamente no processo de desenvolvimento, o sétimo princípio, que diz que um software em funcionamento é a medida primária para o progresso. O que se pode observar é que visto as entregas serem feitas de forma tão rápidas e contínuas a melhor forma da equipe mensurar o que foi realizado até o momento é a apresentação da *Mock* funcionando até onde se permite para avaliação do cliente em questão, que é a medida do progresso do time.

No decorrer da reunião de apresentação do protótipo algumas mudanças começam a ser sugeridas, e nesse ponto a Empresa D tenta seguir o último conceito do Manifesto que diz que as empresas devem se adaptar as mudanças mais que seguir um plano, no entanto, na realidade da empresa mudanças no meio de uma sprint não são fáceis de lidar, uma vez que com entregas programadas para serem curtas, mudanças feitas em um desenvolvimento basicamente semanal podem causar dificuldades. Porém, o entrevistado relata que a maioria das mudanças ocorrem ao final da sprint, na reunião realizada com o cliente, que após validação consegue enxergar melhor a forma em que o software se encontra e a partir daí determinar se algo deverá ser mudado ou não. Dessa maneira, também pode-se observar que a Empresa D trabalha para se encaixar no segundo princípio descrito no Manifesto que aponta que mudanças devem ser bem-vindas mesmo que tardiamente, pois podem gerar vantagem competitiva ao cliente.

5.4.4 Métodos Ágeis

De todas as empresas estudadas a que aplica o *Scrum* mais fielmente é a Empresa D, que possui reuniões de planejamento da sprint, reuniões diárias para acompanhamento e revisão do processo antes da apresentação fazendo com que o processo seja organizado e assertivo.

Outro método encontrado no desenvolvimento da empresa é o *Kanban*, feito de forma digital dentro do software de divisão de tarefas, o Gira. Como citado pelo entrevistado, o software é utilizado para que todas as tarefas sejam registradas e possam ser vistas por todos os membros da equipe. Quando a tarefa é terminada pode ser marcada como concluída e assim não aparece mais no quadro de *tasks* em andamento ou não iniciadas, exatamente como um Kanban funciona.

De acordo com a descrição do desenvolvimento realizado pela empresa, também pode-se notar características da XP, mesmo que o método não seja aplicado em sua totalidade. O *code review* realizado ao final de cada *sprint* pode ser enquadrado como traço da XP, bem como o código do software ser de propriedade coletiva em que todos os membros têm autorização para melhorá-lo ou alterá-lo caso haja necessidade. Outro ponto de semelhança com a Extreme Programming é que no desenvolvimento da Farmbtis muito mais que a documentação de forma escrita, o que também se considera uma documentação é o código em si e os testes automatizados feitos para a conferência do que foi realizado até o momento, característica encontrada no desenvolvimento da empresa e nos princípios da XP.

No que consta sobre métodos ágeis, outro ponto que se pôde observar foi que a Empresa D é a empresa que mais demonstrou conhecimento tanto das partes positivas quanto das partes negativas da adoção de tais metodologias.

O entrevistado destaca que muitas vezes o cliente quer obter informações de escopo, prazo e custo bem definidos, o que acaba “matando” a agilidade do processo, visto que mudanças que devem ser aceitas e bem-vindas mesmo que tardiamente geram alterações no escopo e no orçamento, bem como no prazo de entrega. Assim, em projetos em que o cliente pede o escopo bem definido, a empresa tenta ao máximo seguir os preceitos do Manifesto Ágil, mas também acaba partindo para os princípios descritos no PMBOK, o que tira um pouco o foco da agilidade.

No entanto, o entrevistado deixa claro que das formas de trabalho, a que utiliza metodologias ágeis com entregas contínuas e *feedback* constante é a que melhor gera resultados e a que se encaixa melhor no dia a dia de trabalho da maioria das empresas, pois auxilia na adaptação do trabalho em um ambiente dinâmico e mutável, que é o ambiente em que maioria das empresas de software estão inseridas.

6. CONCLUSÃO

Levando em consideração os resultados obtidos com as entrevistas realizadas nas empresas de tecnologia da cidade de Londrina, pode se concluir que as empresas pesquisadas conhecem os Métodos Ágeis, porém há certa dificuldade de inseri-los no dia a dia, seja por questões culturais ou de aplicabilidade na realidade de cada empresa.

A Empresa A passa por um momento de transição em que a maior dificuldade encontrada é de adaptar a equipe ao método de trabalho ágil, porém conta com uma equipe motivada a participar do processo, pois as melhorias já podem ser sentidas pelos colaboradores. Aos poucos caso a Empresa Consiga estruturar o Scrum de maneira mais concreta o uso da ferramenta poderá trazer assertividade nas entregas e organização ao processo que já é bem conhecido.

Com um longo caminho na indústria de software, a Empresa B acaba sendo a mais independente das empresas no que consta sobre métodos ágeis uma vez que possui métodos próprios criados unicamente para adaptação da Empresa Ao mercado. Dessa foram, há conhecimento sobre o Manifesto e Métodos Ágeis, porém no dia a dia o que se percebe que a agilidade foi inserida nos programas desenvolvidos para padronizar o código, na plataforma criada para comunicação entre os membros da equipe e nos outros processos desenvolvidos ao longo dos anos, não necessariamente se encaixando em uma metodologia específica como Scrum, XP, DSDM etc.

A multinacional, Empresa C, é a que tem um maior caminho a ser percorrido na implementação de métodos ágeis no dia a dia do desenvolvimento. Assim como a Empresa A, a Empresa C faria bom uso de fermentas como Kanban e Scrum, pois além e serem as de maior facilidade de implementação, são os métodos que mais auxiliam a organizar o processo e trazendo uma visão. De acordo com a descrição do processo, ter um kanban mesmo que virtual, auxiliaria a equipe a enxergar quais requisitos estão sendo avaliados, quais atividades precisam ser realizadas para que seja atendido. Assim como para visualizar as mudanças que serão necessárias para a próxima etapa.

O Scrum também seria de grande aplicabilidade para a empresa. A definição das sprints daria ao cliente uma melhor visão do processo além de uma melhor estimativa de tempo e orçamento. Além do fato de que a equipe já tem a adaptabilidade necessária, o que faria com que o Scrum fosse aceito culturalmente. Como a equipe já trabalha com grande fluxo de informações, as reuniões diárias trariam maior assertividade para os membros, e a reunião de revisão da sprint seria proveitosa para que todos analisassem o que foi feito em conjunto avaliar ali mesmo o que poderia ser mudado, levando para o cliente soluções para as possíveis dúvidas.

Características da XP também poderiam ser incorporadas ao projeto como desenvolver um design simples, para que o sistema seja amigável aos olhos do cliente e de fácil entendimento quanto a sua utilização. Outras características como o código ser uma propriedade coletiva e a programação ser desenvolvida em pares, de acordo com o tamanho do sistema em questão também poderiam ser incluídas no processo para que todos saibam o que cada parte da equipe está fazendo e se caso um dos desenvolvedores vier a sair do projeto, que o substituto tenha total capacidade de entender o que está sendo feito e se juntar ao time para suprir as necessidades encontradas.

Por fim, no que diz respeito a Empresa D, pode-se notar que é a empresa que possui maior estruturação dos métodos ágeis no desenvolvimento e no cotidiano. A empresa busca trabalhar com essas metodologias pela eficácia que traz ao processo e para tanto possui uma equipe integrada e conhecedora dos métodos.

Dessa forma, percebe-se que a adoção das metodologias ágeis e dos princípios do Manifesto podem ser inseridos nos processos das empresas, mas é necessário que gestão e equipe trabalhem em conjunto para que o método saia do abstrato e seja realmente utilizado no dia a dia, para que assim os efeitos sejam notados e a Empresa consiga se adaptar à realidade do cliente trazendo entregas contínuas e desenvolvendo produtos com assertividade e valor agregado não apenas para o cliente mas também para o quem se tornará o consumidor final do sistema/produto.

7. REFERÊNCIAS

ABRAHAMSSON, Pekka.; WARSTA, Juhani.; SIPONEN, Mikko T. & RONKAINEN, Jussi. (2003) **New directions on agile methods: a comparative analysis**. IEEE Computer Society, p. 244-254.

AGILE. **Manifesto for Agile Software Development**, 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 28 set. 2021.

AITKEN, Ashley; ILANGO, Vishnu **"A Comparative Analysis of Traditional Software Engineering and Agile Software Development,"** 2013 46th Hawaii International Conference on System Sciences, 2013, pp. 4751-4760, doi: 10.1109/HICSS.2013.31.

ALMEIDA, Samyra L. F; RODRIGUES, Nadja N; LIRA, Heremita B; ET. AL. **Aplicação e análise de processo de desenvolvimento de software: um estudo de caso no GPES-IFPB**. Revista Principia, João Pessoa, nº43 p.152-165, abril, 2018.

BALDUINO, R. **Introduction to Open UP** (Open Unified Process). Eclipse Foudation, 2007. Disponível em:< <https://www.eclipse.org/epf/general/OpenUP.pdf>> . Acesso em: 06 abr. 2022.

BASSI, Dairton. Programação Extrema (XP). In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>>. Acesso em: 23 out. 2021.

BECK, Kent. **Extreme Programming Explained**. Addilson Wesley, Reading,1999.

BEZERRA, Eduardo. **Princípios de análise e Projeto de Sistemas com UML**. São Paulo, Campus, 2002.

BOEHM. Barry. **A Spiral Model of Software Development and Enhancement**. IEE Computer, v.21, n.5, 61-72,1988.

BOERG, J. (2018). **Kanban em 10 passos**. InfoQ Brasil, <<http://www.gianfratti.com/WP/wp-content/uploads/2018/04/InfoQBrasil-Kanban10Passos.pdf>>. Acesso em: 11/05/2022

DUARTE, Jorge. Entrevista em profundidade. In: LEMOS, Márcio. **Entrevista em profundidade**. Florianópolis, 2005. Disponível em: <https://d1wqtxts1xzle7.cloudfront.net/60586395/Entrevista_em_profundidade20190913-12365-1kjb1f2-with-cover-page-v2.pdf?Expires=1638053257&Signature=Oj1AUOoNgg2VNjoOBTA3IDsXH5roc5rvxvboPtbfw-PqRD2z3fbuEYHawcwApSTuctzsFBcfjR2sKUg9SVAUwpBM2lxLfk9jXq2~~aZKuXtjN5A7ltrb2ONixtQXmMOMJB6Bik7HxZ02wA-rVtwNle276OK2DyOWRbR-Bkysqh-glFJ2DSi8jVAC6wZE-XeC45j1A~iYLcG-QdfzKpjQ8~tv3OCmWs5SPYBpFo6tJ3XSr80kxe5CUuUgNF0RszLc5-3fGFvZvEwFQIADKUL4ZLKBRWYbSFcFQtigJ36K12tCeCkVuazP4XvOYyblkssEKKLz5MIhIKOI5kyYcaps9Q_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA> . Acesso em: 27 nov. 2021.

ELLIS, R. **Project Management: Project Lifecycle Planning**, 2010.

GOMES, Alexandre; WILLI Renato; REHEM, Serge. O Manifesto Ágil. In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em: <<https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>>. Acesso em: 20 out. 2021.

CURSOS DE SISTEMAS DE INFORMAÇÃO EM LONDRINA – PR. Guia da Carreira, 2021. Disponível em: <<https://www.guiadacarreira.com.br/cursos-e-faculdades/parana--londrina/sistemas-de-informacao?!=graduacao&pg=3>>. Acesso em: 27/11/2021.

GUIA DO INVESTIDOR. **Prefeitura de Londrina**, 2021. Disponível em: <<http://portal.londrina.pr.gov.br/guia-do-investidor/guia-do-investidor-2021>>. 21 out. 2021

GUTIERREZ, Regina Maria Vinhais; ALEXANDRE, Patrícia Vieira Machado. **Complexo eletrônico: introdução ao software**. BNDES Setorial, Rio de Janeiro, n. 20, p. 3-76, set. 2004. Disponível em: <https://web.bndes.gov.br/bib/jspui/bitstream/1408/2191/1/BS%2020%20Complexo%20Eletr%c3%b4nico_P.pdf>. Acesso em: 11 out 2021.

MARTIN, Robert C. **Desenvolvimento Ágil Limpo: De volta as origens**. Rio de Janeiro, Alta Books, 2020.

MASCHIETTO, Luis. G .; MORAES, Diego.Martins.Polla. D .; ALVES, Nicolli.Souza. R .; AL., Et. **Desenvolvimento de Software com Metodologias Ágeis** . Porto Alegre: Grupo A, 2021. 9786556901824. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9786556901824/>. Acesso em: 20 out. 2021.

MERCADO BRASILEIRO DE SOFTWARE – PANORAMA E TENDENCIAS DE 2021. **Associação Brasileira das Empresas de Software**, 2021. Disponível em: <<https://abesssoftware.com.br/dados-do-setor/#:~:text=O%20estudo%20%E2%80%9CMercado%20Brasileiro%20de,considerados%20os%20mercados%20de%20software%2C>>. Acesso em: 11 out. 2021.

PAPO, José. OpenUP. In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em:

<<https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>>.

Acesso em: 25 out. 2021.

PARANÁ LIDERA PRODUTIVIDADE NO SETOR DE TECNOLOGIA. **Associação das Empresas Brasileiras de Tecnologia da Informação**, 2019. Disponível em: <<https://www.assespropr.org.br/parana-lidera-productividade-no-setor-de-tecnologia/>>. Acesso em 20 out. 2021.

Pereira, R. (2018). User Experience Design: Como criar produtos digitais com foco nas pessoas. Brasil: Casa do Código.

PEREIRA, Rogério. **User Experience: Como criar produtos digitais com foco nas pessoas**. 1ª Ed, Editora Casa do Código, 2018.

PRIKLADNICKI, Rafael; MAGNO, Alexandre. O Framework do Scrum. In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em:

<<https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>>.

Acesso em: 24 out. 2021.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de software: uma abordagem profissional**. 8ªed. AMGH, Porto Alegre,2016.

PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em:

<<https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>>.

Acesso em: 20 out. 2021.

PRODANOV, Cleber; FREITAS, Ernani. **Metodologia do trabalho científico**. Ed.2. Rio Grande do Sul, 2013. Disponível em:<<https://www.fee.br/Comum/midias/0163c988-1f5d-496f-b118-a6e009a7a2f9/E-book%20Metodologia%20do%20Trabalho%20Cientifico.pdf>>.

Acesso em: 26 ago. 2021.

RATAMAL, Adail Muniz. FDD – Feature-Driven Development. In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>. Acesso em: 26 out. 2021.

ROYCE, W. **Managing the development of large software systems: Concepts and techniques**. In: Proc. IEEE WESCOM. IEEE Computer Society Press, Los Alamitos. 1970.

SABBAGH, Rafael. **Scrum: Gestão ágil para projetos de sucesso**. São Paulo, Casa do Código, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ªed, Pearson Education, São Paulo, 2011.

THE CHAOS REPORT. **The Standish Group**,1994. Disponível em: https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf. Acesso em: 28 set. 2021.

VALE, Alisson. Kaban. In: PRIKLADNICKI, Rafael; WILLI, Renato; MILANI, Fabiano. **Métodos Ágeis para Desenvolvimento de Software**. Porto Alegre Grupo A, 2014. 9788582602089. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788582602089/pageid/1>. Acesso em: 8 mai. 2022.

VETORAZZO, Adriana de S. **Engenharia de Software**. Porto Alegre, Grupo A, 2018. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595026780/pagei/3>. Acesso em: 20 out. 2021.

WAZLAWICK, Raul Sidinei. **Engenharia de software: conceitos e prática**. 1ªed,Porto Alegre, Elsevier, 2013.

WAZLAWICK, Raul Sidinei. **Engenharia de software: conceitos e prática.**
2ªed,Porto Alegre, Elsevier, 2019.

**APÊNDICE A – Questionamentos para Entrevista com Responsável pela
Empresa**

1. Para dar início, eu gostaria de saber um pouco mais sobre as soluções/produtos que vocês desenvolvem. Eles se encaixam em qual categoria? São sistemas em geral, aplicativos personalizados? Como funciona?
2. Como os requisitos são definidos? O cliente geralmente já tem algo definido ou os requisitos vão surgindo à medida que o produto é desenvolvido?
3. Há algum procedimento padrão para dar início ao projeto?
4. Quais documentações do projeto vocês elaboram no início do processo?
5. Existe algum processo padrão para o desenvolvimento do produto? Há algum método em que vocês se baseiam para construir o produto?
6. O cliente é presente durante as fases do projeto?
7. As entregas do produto ocorrem de que forma? Parciais para validação do cliente ou com data para entrega final definida?
8. Como a equipe é dividida para a execução do projeto?
9. De que forma a equipe recebe possíveis alterações nos requisitos do produto?
10. Como é a comunicação entre os membros da equipe? Eles conseguem se comunicar com facilidade?
11. Como são decididas as etapas do processo de construção do projeto?
12. O senhor(a) já tinha ouvido falar sobre os métodos ágeis para desenvolvimento de software?
13. (SE SIM) O que o senhor(a) sabe/entende dos métodos ágeis?
14. Já houve alguma tentativa de incorporar algum conceito desses métodos no dia a dia do trabalho?
15. Há algum ponto de melhoria no processo que o senhor(a) consegue identificar e que poderia ser solucionado com a adoção ágil no desenvolvimento?