



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS APLICADAS E EDUCAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**A utilização da metodologia *Scrum* sob a percepção da
equipe de desenvolvimento em uma empresa privada de
software: Um estudo de caso**

Jéssyca Ferreira de Oliveira

Rio Tinto - PB

2014

Jéssyca Ferreira de Oliveira

A utilização da metodologia *Scrum* sob a percepção da equipe de desenvolvimento em uma empresa privada de software: Um estudo de caso

Trabalho de Conclusão de Curso
apresentado ao Centro de Ciências
Aplicadas e Educação da Universidade
Federal da Paraíba, para a obtenção do
título de Bacharel em Sistemas de
Informação.

Orientador: Prof. José Jorge Lima Dias Júnior

Rio Tinto - PB

2014

O48u Oliveira, Jéssyca Ferreira de.

A utilização da metodologia Scrum sob a percepção da equipe de desenvolvimento em uma empresa privada de software: um estudo de caso. / Jéssyca Ferreira de Oliveria. – Rio Tinto: [s.n.], 2014.
78 f.: il. –

Orientador: Prof. José Jorge Lima Dias Júnior.

Monografia (Graduação) – UFPB/CCAE.

Jéssyca Ferreira de Oliveira

A utilização da metodologia *Scrum* sob a percepção da equipe de desenvolvimento em uma empresa privada de software: Um estudo de caso

Trabalho de Conclusão de Curso submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Assinatura do autor: _____

APROVADO POR:

Orientador: Prof. Me. José Jorge Lima Dias Júnior
Universidade Federal da Paraíba – Campus IV

Profa. Dra. Ayla Débora Dantas de Souza Rebouças
Universidade Federal da Paraíba – Campus IV

Profa. Dra. Yuska Aguiar
Universidade Federal da Paraíba – Campus IV

Profa. Me. Joyce Aline
Universidade Federal da Paraíba – Campus IV

RIO TINTO - PB

2014

*À minha família, por sempre ter me incentivado
e por ter me ensinado a nunca desistir dos meus sonhos.*

AGRADECIMENTOS

Primeiramente, agradeço a Deus, por ter guiado meus passos até aqui e por ter me dado forças para não desistir em nenhum momento, até mesmo nas situações mais difíceis.

À minha mãe, Maristela Oliveira, por ser minha referência e meu ponto de apoio em todas as horas. Obrigada por ter me incentivado, por me ensinar a lutar por nossos objetivos e por sempre acreditar no meu potencial.

Aos meus irmãos, Jenyffer Ferreira e Jorge Luiz, por sempre estarem ao meu lado e por me fazerem lembrar que como mais velha tenho que dá exemplo para eles.

Aos meus avós, Beatriz Oliveira e Manoel Gomes, que nos deixou enquanto escrevia este trabalho. Foi muito difícil continuar sem vocês e ter forças para seguir em frente, mas obrigada por terem me ensinado tudo que ensinaram enquanto estavam conosco. Por terem sido a mulher mais iluminada que conheci na minha vida, que me passou tanta coisa boa e o homem mais batalhador que conheci e que assumiu papel de pai para mim e meus irmãos, nunca deixando nos faltar nada. Sem dúvida a contribuição de vocês para mais esta conquista foi essencial.

Agradeço às minhas tias, tanto maternas quanto paternas por terem me ajudado desde sempre e a minha avó Altamira, em especial, por se fazer presente em todos os momentos e se preocupar comigo, querendo sempre o melhor para mim.

Agradeço a todos os professores que tive a honra de conhecer e de ser aluna, que compartilharam seu conhecimento e que contribuíram com minha formação durante estes anos na universidade.

Ao meu orientador, Jorge Dias, por ter acreditado em mim e por ter me chamado para ser sua orientanda. Obrigada por suas orientações e por sua paciência e contribuição que foram de grande importância.

As examinadoras da banca, Ayla, Yuska e Joyce, pela ajuda dada. Suas contribuições, sem dúvida, foram de grande valia.

Aos colegas que fiz durante o curso e todos os amigos verdadeiros que fiz ao longo da minha vida.

Aos respondentes do questionário e colegas de empresa, através dos quais adquiri muito conhecimento durante o tempo que passei trabalhando com todos.

RESUMO

A entrega de softwares com qualidade para os clientes é um fator que tem sido muito discutido nos últimos tempos. Em consequência disso, muitas organizações estão procurando incorporar ao seu processo de desenvolvimento de software as metodologias ágeis para uma entrega mais rápida e que atendam as demandas e exigências dos seus clientes. Nesse contexto, este trabalho tem o propósito de apresentar um estudo de caso sobre a utilização da metodologia ágil *Scrum* em uma empresa privada de desenvolvimento de software na cidade de João Pessoa - Paraíba. A finalidade da pesquisa é analisar as características e adesão do método ágil *Scrum* do ponto de vista da equipe através de dados qualitativos e quantitativos obtidos por meio de dois instrumentos de coleta: questionário e observação direta. Por fim, este trabalho apresenta quais as técnicas da metodologia *Scrum* foram implantadas na empresa investigada em relação ao planejamento do projeto, os artefatos utilizados, os papéis do *Scrum* e as iterações (*sprints*), mostrando que a equipe concorda que muitos pontos da metodologia são adotados na empresa, porém, algumas questões precisam ser melhor compreendidas para o uso do *Scrum* como o mesmo sugere.

Palavras-chave: Metodologias ágeis, *Scrum*, Adoção do *Scrum*.

ABSTRACT

The delivery of quality software to customers has been much discussed in recent times. As a result, many organizations are looking for incorporating agile methodologies into their software development process in order to obtain a faster delivery that meets the demands and requirements of its customers. In this context, this paper aims at presenting a case study on the use of the *Scrum* agile methodology in a private software development company in the city of João Pessoa - Paraíba. The purpose of this research is to analyze the characteristics and adhesion of the *Scrum* agile method from the team point of view through qualitative and quantitative data obtained by means of two data collection instruments : questionnaire and direct observation. Finally, this paper presents which characteristics of the *Scrum* methodology have been deployed in the investigated company in relation to project planning, artifacts used, roles and *Scrum* iterations (*sprints*), showing that the team agrees that many points of the methodology adopted in the company are, however, some issues need to be better understood using *Scrum* as the same suggests.

Keywords : Agile methodologies , *Scrum*, *Scrum* Adoption

Lista de Figuras

Figura 1. Atividades para criação de software.....	22
Figura 2. Etapas de desenvolvimento do modelo cascata (SOMMERVILLE 2007).....	24
Figura 3. Desenvolvimento iterativo em espiral (TELES 2004).....	28
Figura 4. Ciclo de vida do Scrum (WIKIPEDIA 2014)	32
Figura 5. Burndown Chart (ALL ABOUT AGILE 2011).....	34
Figura 6. Cartas do Planning Poker (IT-Zynergy ApS 2014).....	35
Figura 7. Processo do estudo de caso (YIN 2010).....	37
Figura 8. Sala da empresa.....	54
Figura 9. Gráfico comparativo entre os resultados esperados e os resultados obtidos pelo questionário	63

Lista de Tabelas

Tabela 1. Fases e objetivos do RUP.....	26
Tabela 2. Características do Scrum adotadas na empresa.....	41
Tabela 3. Resumo comparativo da avaliação do ambiente antes e após a implantação do método <i>Scrum</i> . (VASCONCELOS 2010)	42
Tabela 4. Perguntas e seus respectivos objetivos.....	51
Tabela 5. Escala de Likert e pontuação para as respostas.....	53
Tabela 6. Exemplo da avaliação das respostas (adaptado de OLIVEIRA 2005)	56
Tabela 7. Frequência das respostas e Ranking Médio do questionário	57

Lista de Quadros

Quadro 1. Fases do estudo de caso e suas respectivas descrições.....	38
--	----

Lista de Símbolos e Abreviaturas

DBA: Administrador de Banco de Dados

FDD: *Feature Driven Development*

RUP: *Rational Unified Process*

TI: Tecnologia da Informação

XP: *Extreme Programming*

Sumário

1 INTRODUÇÃO	15
1.1 Motivação	16
1.2 Objetivos.....	17
1.3 Metodologia.....	18
1.4 Estrutura da Monografia	19
2 FUNDAMENTAÇÃO TEÓRICA.....	20
2.1 Processo de desenvolvimento de software	20
2.2 Metodologias para desenvolvimento de software	23
2.2.1 Modelos e metodologias tradicionais.....	24
2.2.2 Metodologias ágeis.....	26
2.2.3 Principais modelos ágeis para desenvolvimento de software	28
2.3 <i>Scrum</i>	29
2.3.1 Papéis e responsabilidades no <i>Scrum</i>	30
2.3.2 Eventos do <i>Scrum</i>	31
2.3.3 Métricas e estimativas no <i>Scrum</i>	34
2.4 Estudo de caso	36
2.4.1 Estrutura do estudo de caso	36
3 TRABALHOS RELACIONADOS.....	41
3.1 Scrum no Brasil.....	41

3.2 Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo.....	43
3.3 Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet	45
4 ESTUDO DE CASO EM UMA EMPRESA PRIVADA DE DESENVOLVIMENTO DE SOFTWARE	47
4.1 Delineamento da pesquisa	47
4.2 Unidade de análise.....	47
4.3 Coleta de dados	49
4.4 Aplicação do piloto	55
5 ANÁLISE E DISCUSSÃO DOS DADOS.....	56
5.1 Síntese e discussão dos resultados.....	63
6 CONCLUSÃO	65
6.1 Trabalhos Futuros	65
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	67
APÊNDICE A - PROTOCOLO DE ESTUDO DE CASO	67
APÊNDICE B - ROTEIRO DE OBSERVAÇÃO	73
APÊNDICE C - QUESTIONÁRIO	74

1 INTRODUÇÃO

Considerando o papel do software nos dias atuais e sua importância em meio às organizações, as empresas de desenvolvimento de sistemas estão cada vez mais empenhadas em desenvolverem seus softwares de uma maneira mais eficaz e que satisfaça seus clientes.

O processo de desenvolvimento de software é um conjunto de atividades e resultados associados que têm por objetivo produzir software eficiente, de alta qualidade, com baixa taxa de erros e que atenda às necessidades e expectativas do usuário de forma geral (JACOBSON, 1999).

Com isso, para apoiar o processo de desenvolvimento de software, diversos modelos e metodologias como Cascata, RUP (*Rational Unified Process*), XP (*Extreme Programming*) e *Scrum* foram criados. De acordo com Sommerville (2000), essas metodologias são um conjunto de práticas recomendadas para o desenvolvimento de softwares, sendo que essas práticas, geralmente, passam por fases ou passos, que são subdivisões do processo para ordená-lo e melhor gerenciá-lo. Incumbindo assim a cada organização investigar e decidir o modelo ou metodologia mais adequada ao seu perfil e que ajude a tornar a tarefa de desenvolvimento mais eficiente/eficaz.

As metodologias ágeis têm ganhado bastante atenção da indústria pelo fato de compensarem os pontos considerados negativos dos modelos denominados tradicionais, como, por exemplo, demora na entrega de um software funcional. Propondo a ideia de entregar sistemas de forma contínua e antecipada, o desenvolvimento ágil está cada vez com mais adeptos. Além disso, pesquisas mostram que a adoção dos métodos ágeis trazem resultados mais satisfatórios tanto para as empresas desenvolvedoras quanto para os clientes e usuários, como transparência e visibilidade do status do projeto, melhoria da qualidade do produto final, melhoria na comunicação e redução dos riscos e das indesejáveis surpresas (MULLER, 2009; AZEVEDO, 2012; SOUZA, 2011).

Uma das metodologias ágeis procuradas pelas empresas é o *Scrum*, por possuir objetivos e características capazes de oferecer uma entrega rápida ao cliente, ter como uma de

suas prioridades a comunicação entre os membros da equipe e a participação do cliente ao longo do desenvolvimento.

No entanto, o desenvolvimento de software é um processo complexo (SCHWABER, 2004), e quando se trata da implantação de uma metodologia surgem muitas dificuldades, impactando assim em muitos aspectos dentro das organizações.

Muitos problemas na adoção de uma nova metodologia estão relacionados diretamente com a resistência dos membros da equipe ao novo método adotado. Segundo McMenamin (1996) e DeMarco & Lister (1999), as pessoas possuem uma tendência natural para resistirem a mudanças. A falta de treinamento adequado também é outro fator de grande influência, pois alguma característica passada por um método ou metodologia pode não ser compreendida pelo grupo e as práticas serem adotadas sem um real entendimento do que se está sendo feito e qual o objetivo.

Diante disto, é importante estudar como se tem dado a adoção e utilização das metodologias ágeis nas empresas de desenvolvimento de software e como ocorre a percepção (ação e efeito de perceber ou compreender algo) dos desenvolvedores sobre as práticas adotadas com o objetivo de aprender com os erros e aprimorar as adoções futuras. Portanto, este trabalho apresenta um estudo de caso para analisar a adoção do *Scrum* em uma empresa de software sob a percepção da equipe.

1.1 Motivação

Com as ideias sugeridas pelo Manifesto Ágil¹ e a criação de metodologias baseadas nos seus princípios, diversas organizações no mundo se propuseram a tentar implantar esses novos métodos para melhoria do desenvolvimento de software. No Brasil não foi diferente, várias empresas como C.E.S.A.R, Caelum, Globo.com, Gol, Petrobras, UOL e Vivo, colocaram-se a disposição de adotar metodologias ágeis para desenvolver sistemas.

¹ Reunião realizada no ano de 2001 que propôs quatro princípios para o desenvolvimento ágil de software.

Contudo, para adotar-se algo novo é necessário um estudo antecipado e um acompanhamento do que está sendo feito, pois ao mesmo tempo em que essa adoção pode afetar de modo positivo, pode haver comprometimento no desenvolvimento do software se a metodologia não for adotada de maneira adequada.

Várias pesquisas, como as apresentadas no Capítulo 3, foram realizadas para avaliar a implantação de métodos ágeis em diferentes organizações, tentando analisar o que de fato é benéfico e proveitoso, quais características realmente são adotadas, em que momento a adoção pode falhar e os impactos que a mesma oferece. Porém, com base no que foi pesquisado e estudado não foram encontrados na engenharia de software estudos sobre como a adoção do *Scrum* é compreendida pela equipe de desenvolvimento.

Diante desses fatos e visando entender melhor a adoção da metodologia ágil *Scrum* em uma organização, este trabalho possui sua motivação e sua contribuição no aspecto acadêmico e industrial. No que diz respeito ao aspecto acadêmico, o trabalho contribui para a evolução da maturidade científica na Engenharia de Software, uma vez que este trabalho aplicou o método de estudo de caso com rigor científico para levantar evidências acerca do objeto estudado. No âmbito da indústria, o estudo se torna relevante uma vez que se trata de um caso real. Este caso é uma empresa que se encontra em uma fase de adoção do *Scrum* há aproximadamente um ano e decidiu usar a metodologia para atender as demandas dos clientes dentro do prazo, melhorar a comunicação entre os membros da equipe e melhorar o gerenciamento da equipe.

1.2 Objetivos

1.2.1 Objetivo Geral

Para a definição do objetivo geral utilizou-se o modelo proposto por Basili (2002) para fins de organização.

Analisar a utilização do *Scrum* **com relação ao** planejamento, papéis, artefatos e *sprint* **com o objetivo** de verificar a percepção da aderência ao *Scrum* **sob o ponto de vista** dos desenvolvedores **no contexto de** uma empresa privada de software.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Verificar em que nível as características da metodologia *Scrum* estão sendo adotadas na empresa sob a percepção da equipe de desenvolvimento;
- Identificar possíveis impedimentos no uso do processo;

1.2.3 Questões de Pesquisa

A fim de guiar este estudo, as seguintes questões de pesquisa foram utilizadas:

QP01. Que aspectos do *Scrum* estão sendo adotados na empresa analisada?

QP02. Qual a percepção da equipe em relação ao planejamento do projeto?

QP03. Qual a percepção da equipe em relação aos artefatos utilizados?

QP04. Qual a percepção da equipe em relação aos papéis?

QP05. Qual a percepção da equipe em relação a *Sprint*?

1.3 Metodologia

O presente trabalho é um estudo de caso, de caráter descritivo, com abordagem qualitativa e quantitativa. Para obtenção dos dados foi elaborado um questionário e realizada uma observação direta na empresa de análise, a qual era ambiente de trabalho da pesquisadora.

O estudo de caso é um método de pesquisa que utiliza, geralmente, dados qualitativos, coletados a partir de eventos reais, com o objetivo de explicar, explorar ou descrever fenômenos atuais inseridos em seu próprio contexto. Caracteriza-se por ser um estudo detalhado e exaustivo de poucos, ou mesmo de um único objeto, fornecendo conhecimentos profundos (Eisenhardt, 1989; Yin, 2009).

De acordo com Yin (1994) o método quantitativo é o uso de instrumental estatístico, de dados numéricos. Já o método qualitativo se caracteriza pela qualificação dos dados coletados, durante a análise do problema.

A pesquisa qualitativa não se preocupa com representatividade numérica, mas sim com o aprofundamento da compreensão de um grupo social, de uma organização etc.(Goldenberg, 1999).

A teoria sobre o método de estudo de caso, assim como todo seu processo, é apresentado em detalhes no Capítulo 2, seção 2.5.

Para a análise dos dados utilizou-se o modelo proposto por Oliveira (2005), o qual visa estabelecer o Ranking Médio para as respostas do questionário, em conjunto com as descrições obtidas da observação direta.

1.4 Estrutura da Monografia

Este trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, abordando temas importantes dentro do contexto da pesquisa, como, boas práticas no processo de desenvolvimento de software, modelos e metodologias para o desenvolvimento de software, principais métodos ágeis, *Scrum* e suas características, e por fim, oferece uma visão geral sobre estudo de caso. O Capítulo 3 contém alguns trabalhos relacionados à linha de pesquisa. O Capítulo 4 explana a metodologia utilizada para a pesquisa em questão, expõe o delineamento da mesma, apresenta a empresa analisada e como foi realizada a coleta de dados. No Capítulo 5 estão expostas a análise dos dados e a discussão dos mesmos. No Capítulo 6 estão apresentadas algumas considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste trabalho, alguns conceitos como processo de desenvolvimento de software, modelos tradicionais, metodologias ágeis, *Scrum* e estudo de caso fundamentam o mesmo. Em consequência disso, serão apresentados e descritos neste capítulo os principais conceitos e características relacionados aos assuntos mencionados.

2.1 Processo de desenvolvimento de software

O processo de software é um conjunto de passos parcialmente ordenados para atingir uma meta (REZENDE 2005). Ele deve proporcionar um conjunto apropriado de ações e tarefas que devem ser realizadas (PRESSMAN 1995).

Embora cada projeto tenha suas características próprias, é possível estabelecer conjuntos de atividades, artefatos e recursos comuns para toda a organização. Um processo eficaz deve, claramente, considerar as relações entre as atividades, os artefatos produzidos no desenvolvimento, as ferramentas e os procedimentos necessários e a habilidade, o treinamento e a motivação do pessoal envolvido (FALBO, 1998).

A necessidade e o exercício de se adotar um processo adequado emergiu da chamada Crise do Software. De acordo com Pressman (2004):

“os problemas que a originou tinham relacionamento direto com a forma de trabalho das equipes. Eram problemas que não se limitavam a “sistemas que não funcionam corretamente”, mas envolviam também dúvidas sobre como desenvolver e manter um volume crescente de software e ainda estar preparado para as futuras demandas. Essencialmente, eram sintomas provenientes do pouco entendimento dos requisitos por parte dos desenvolvedores, somados às técnicas e medidas pobres aplicadas sobre o processo e o produto, além dos poucos critérios de qualidade estabelecidos até então.”

Um processo é organizado em atividades, as quais são responsabilidades dos membros da equipe do projeto desempenhá-las de maneira profissional e ética, cada um exercendo seu papel.

Em um processo de desenvolvimento de software existem diversos passos que devem ser considerados para a criação de um software com qualidade. Ao se tomar como base definições relatadas por Sommerville (2007) e Pressman (2004) a Figura 1 mostra quais são as principais atividades comuns à grande parte dos processos existentes e o papel que cada uma representa. Porém, deve-se levar em consideração que para cada caso detalhes do processo de software são diferentes, existindo também outras formas de organizar essas atividades.

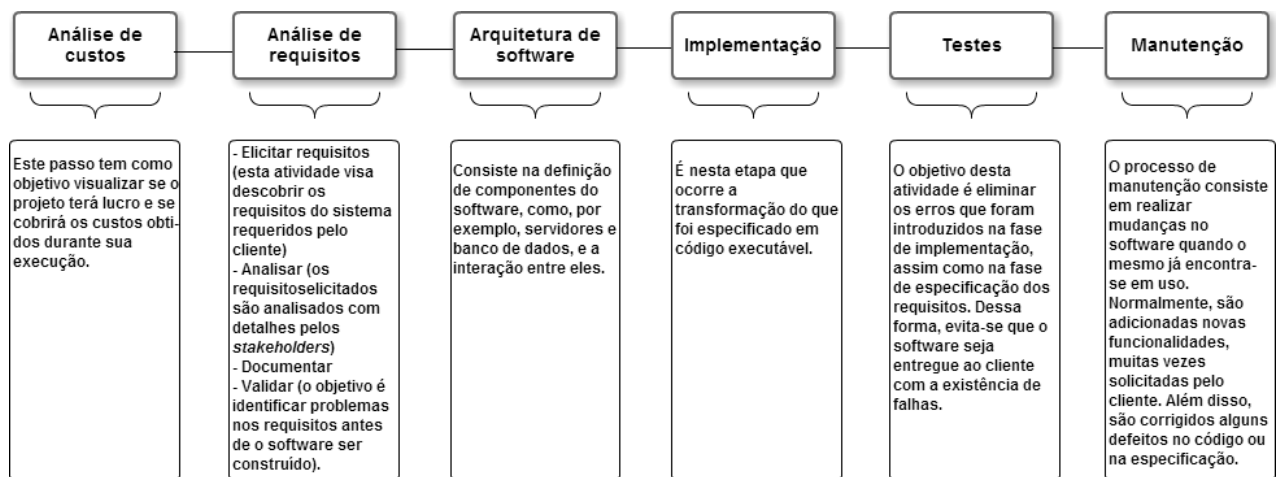


Figura 1. Atividades para criação de software

Os softwares devem atender as necessidades dos clientes e usuários, e alguns atributos durante o processo de desenvolvimento são considerados de suma importância para um bom resultado final. Esses atributos são citados por Sommerville (2007):

- Facilidade de manutenção: a manutenibilidade de um sistema pode ser afetada pela má configuração do software, indisponibilidade de uma equipe qualificada e cuidado inadequado com o projeto, codificação e teste. Por isso o software deve ser projetado e escrito de modo que evolua de acordo com as necessidades mutáveis do cliente;
- Eficiência: a eficiência abrange, entre outros fatores, o tempo de resposta do sistema, a utilização de memória e o tempo de processamento. Em um software eficiente nenhum dos fatores citados deve ser desperdiçado;
- Facilidade de uso: o sistema deve dispor de uma interface apropriada para o tipo de usuário que foi projetado, em que o mesmo deve ser utilizado sem nenhum esforço e dificuldade;
- Confiabilidade: é a possibilidade de o sistema funcionar sem a ocorrência de falhas, sem ocasionar danos econômicos e/ou físicos ao sistema.

2.2 Modelos e metodologias para desenvolvimento de software

Diante dos desafios presentes no processo de desenvolvimento de software, como, por exemplo, uma análise incorreta que pode acarretar em um resultado final diferente do desejado pelo cliente e manter as entregas dentro dos prazos, tornou-se necessário buscar modelos ou metodologias para a criação de sistemas mais eficientes e com maior qualidade.

A adoção desses modelos e metodologias para a melhoria de um processo em uma empresa vem da necessidade de deixar de lado processos geralmente improvisados, em que a equipe apenas se preocupa na solução de problemas imediatos, tendo seus cronogramas e orçamentos estourados, atividades de testes e revisão frequentemente reduzidas ou até mesmo eliminadas em decorrência dos atrasos ocorridos durante projeto, e a satisfação do cliente está aquém do desejado.

As metodologias e modelos para os processos de desenvolvimento de software passaram a existir de maneira bastante crescente e se apresentam de diversas formas, cabendo a cada empresa escolher a mais adequada ao seu perfil e seus objetivos.

2.2.1 Modelos e metodologias tradicionais

Existem alguns modelos considerados tradicionais ou convencionais, como, por exemplo, o modelo em cascata, o incremental e o evolucionário. O modelo cascata, um dos mais conhecidos, foi proposto por Royce em 1970. A versão inicial deste modelo foi aprimorada no decorrer dos anos e continua em uso atualmente. O modelo cascata consiste em que as diferentes etapas de desenvolvimento seguem uma sequência, como mostra a Figura 2.

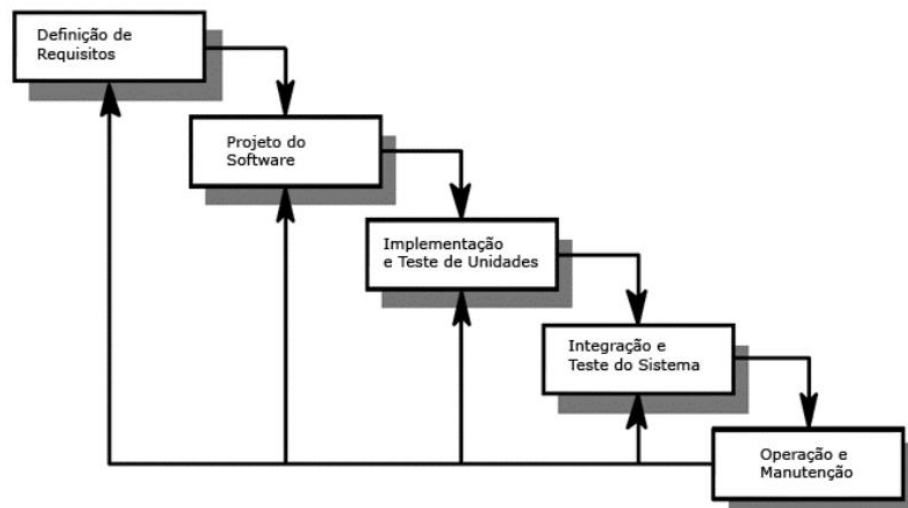


Figura 2. Etapas de desenvolvimento do modelo cascata

Fonte: Sommerville (2007)

Segundo Sommerville (2007) essas etapas podem ser definidas da seguinte forma:

(1) Definição de requisitos: As funções, as restrições e os objetivos do sistema são estabelecidos por meio da consulta aos usuários do sistema. Em seguida, são definidos em detalhes e servem como uma especificação do sistema;

(2) Projeto de sistema e software: Agrupa os requisitos em sistemas de hardware ou de software. Estabelece uma arquitetura do sistema geral;

(3) Implementação e teste de unidade: Durante esse estágio, o projeto de software é compreendido como um conjunto de programas ou de unidades de programa. O teste de unidade envolve verificar que cada unidade atenda a sua especificação;

(4) Integração e teste de sistema: As unidades de programa ou programas individuais são integrados e testados como um sistema completo a fim de garantir que os requisitos de software foram atendidos. Depois dos testes, o sistema de software é entregue ao cliente;

(5) Operação e manutenção: Normalmente, esta é a fase mais longa do ciclo de vida. O sistema é instalado e colocado em operação. A manutenção envolve corrigir erros que não foram descobertos em estágios anteriores do ciclo de vida ou aumentar as funções desse sistema à medida que novos requisitos são descobertos.

O modelo cascata pode ser utilizado principalmente quando os requisitos de um determinado problema são bem compreendidos. Uma forma de utilizar o modelo cascata é quando se precisa fazer adaptações ou aperfeiçoamentos em um sistema já existente.

Entretanto, ele é apenas um modelo, ou seja, define de forma abstrata as etapas de um processo, para que sirva como base para a elaboração de processos.

Em meio às metodologias tradicionais, uma bastante utilizada é o RUP (*Rational Unified Process*), na qual o seu ciclo de vida é uma implementação do modelo espiral. O modelo espiral foi proposto, originalmente, por Barry Boehm, em 1988, em seu artigo *A Spiral Model of Software Development and Enhancement* (Um Modelo Espiral de Desenvolvimento de Software e de Valorização) e, ao invés de representar o processo de software como uma sequência de atividades, a exemplo do Modelo Cascata, ele é representado através de uma espiral, onde cada loop representa uma fase do processo de software. (SOMMERVILLE, 2007).

Segundo Pressman (2006) e Sommerville (2007) o RUP possui quatro fases, em que a divisão do projeto nessas quatro fases tem como objetivo capturar a dimensão do tempo de um projeto. Cada fase e seu objetivo correspondente são expostos na Tabela 1

Tabela 1. Fases e objetivos do RUP

Fase	Objetivo
Fase de concepção ou iniciação	Estabelecer a visão, o escopo e a viabilidade econômica do projeto.
Fase de elaboração	Eliminar os principais riscos e desenvolve o plano de projeto.
Fase de construção	Relacionada com o projeto, programação e teste do sistema. Ao término desta fase tem-se um sistema de software em funcionamento e a documentação associada pronta para ser liberada aos usuários. A primeira versão do sistema é desenvolvida.
Fase de transição	Transferir o software do desenvolvedor para o usuário final para teste e aceitação, implantando o produto no ambiente real.

2.2.2 Metodologias ágeis

Porém, em meio às metodologias consideradas mais clássicas surgiram as metodologias ágeis em resposta as fraquezas dessas já existentes, como, por exemplo, não suportar mudanças rápidas nos requisitos, não fornecer feedback e alta complexidade para projetos de pequeno porte.

O desenvolvimento ágil é fruto de uma reunião realizada no início de 2001 com participação de dezessete profissionais conhecidos na Engenharia de software, em que houve um acordo que os projetos bem sucedidos tinham em comum um pequeno conjunto de valores. Baseado nisso criaram o Manifesto Ágil que segue os seguintes princípios:

- **Indivíduos e interação entre eles** 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.

Uma característica das metodologias ágeis consiste em tirar o foco do processo de desenvolvimento em si, dando ênfase na contribuição dos *stakeholders* (todos envolvidos no projeto) e permitindo uma participação mais efetiva do cliente no processo de construção do software. Além disso, outro aspecto importante dos métodos ágeis é a previsão antecipada e entrega contínua do software.

Segundo Highsmith (2004), agilidade em TI é a habilidade de criar e responder a mudanças, buscando a obtenção de lucro em um ambiente de negócio turbulento. As vantagens de se usar uma metodologia baseada nesse princípio são muitas, como a maximização do comprometimento por parte da equipe, melhoria na comunicação, inspeção e adaptação constantes do processo em busca da melhoria contínua e redução dos desperdícios, redução dos riscos e das surpresas indesejáveis, antecipação dos problemas e maior agilidade na tomada de decisões.

Para Teles (2004), no desenvolvimento ágil, assim como no RUP, os projetos adotam o modelo iterativo e em espiral (Figura 3). Neste processo, todas as fases descritas no modelo em cascata são executadas diversas vezes ao longo do projeto, produzindo ciclos curtos que se repetem ao longo de todo o desenvolvimento, sendo que, ao final de cada ciclo, sempre se tem um software funcional, testado e aprovado. Os ciclos são chamados de iterações e crescem em número de funcionalidades a cada repetição, sendo que, no último ciclo, todas as funcionalidades desejadas estarão implementadas, testadas e aprovadas.

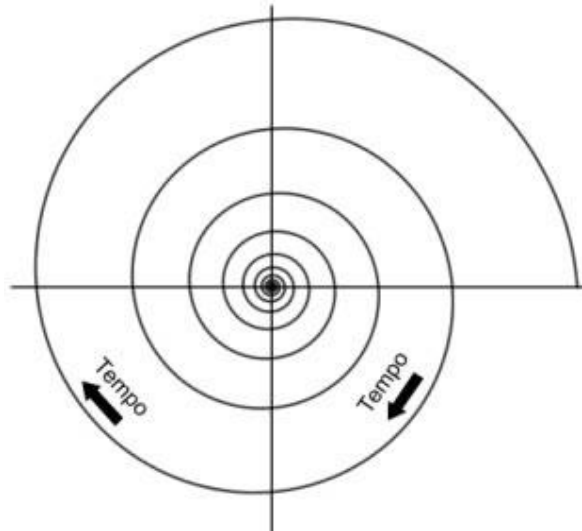


Figura 3. Desenvolvimento iterativo em espiral

Fonte: Teles (2004)

2.2.3 Principais metodologias ágeis para desenvolvimento de software

Os métodos modernos de desenvolvimento, também conhecidos como métodos ágeis se dispõem de diversas formas, apesar de manterem a mesma base, desenvolver software de forma ágil e que satisfaça o cliente. Entre os métodos existentes nos dias atuais os mais conhecidos são: *Extreme Programming (XP)*, *Scrum* e *Feature Driven Development (FDD)*.

A metodologia XP consiste numa série de práticas e regras que permitem aos programadores desenvolver software de uma forma dinâmica e muito ágil considerando aspectos que se aplicam no conceito de agilidade, custo, escopo e qualidade no desenvolvimento de sistemas (BECK, 2000).

Soares (2007) complementa ao relatar que o XP enfatiza o desenvolvimento rápido do projeto e visa garantir a satisfação do cliente, além de favorecer o cumprimento das estimativas, proporcionando um agradável ambiente de desenvolvimento de software para os seus seguidores, que são conduzidos por quatro valores: Retorno(feedback), Comunicação, Simplicidade e Coragem.

O *Scrum* está sendo usado desde 1990 na área de gerenciamento de projetos e depois de um tempo foi adaptado e passou a ser adotado para gerenciar o desenvolvimento de software. Surgiu da necessidade de se encontrar uma forma de desenvolvimento de software que suprisse a necessidade dos clientes de uma maneira diferente da abordagem tradicional.

Baseado em um jogo de Rugby, a equipe age unida para alcançar seus objetivos. Como o objeto deste estudo foca nesta metodologia, a mesma será melhor apresentada na próxima seção.

O FDD foi criado em 1997 por Jeff De Lucca e segundo Pacheco (2009) é um método que prega a visibilidade do estado do projeto de forma consistente e honesta. Através dessa metodologia se consegue saber quantas funcionalidades já foram desenvolvidas e quantas faltam ser desenvolvidas porque tudo é orientado às funcionalidades.

As metodologias ágeis são adequadas em projetos em que ocorrem muitas mudanças, em que os requisitos são passíveis de alterações, as datas de entregas do software são curtas e o desenvolvimento rápido é essencial.

2.3 Scrum

A metodologia *Scrum* será mais detalhada nesta seção por que o processo de software da empresa a qual este estudo de caso se refere é baseado no *Scrum*.

Segundo Schwaber e Sutherland (2011), fundadores da metodologia na área de desenvolvimento de software, o *Scrum* é um framework dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. Está sendo utilizado nas organizações desde o início de 1990 para gerenciar o desenvolvimento de produtos complexos, podendo empregar vários processos ou técnicas.

O *Scrum* é baseado nas teorias empíricas que afirmam que o conhecimento é fruto da experiência e da tomada de decisão com base no que é conhecido. Para Schwaber e Sutherland (2011) três pilares apoiam a implementação de controle de processo empírico, são eles:

- **Transparência:** aspectos significativos do processo devem estar visíveis aos responsáveis pelos resultados. Esta transparência requer aspectos definidos por um padrão comum para que os observadores compartilhem um mesmo entendimento do que está sendo visto.

- Inspeção: os usuários *Scrum* devem, frequentemente, inspecionar os artefatos *Scrum* e o progresso em direção ao objetivo, para detectar indesejáveis variações. Esta inspeção, não deve, no entanto, ser tão frequente que atrapalhe a própria execução das tarefas. As inspeções são mais benéficas quando realizadas de forma diligente por inspetores especializados no trabalho a se verificar.
- Adaptação: se um inspetor determina que um ou mais aspectos de um processo desviou para fora dos limites aceitáveis, e que o produto resultado será inaceitável, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios.

Segundo Schwaber e Beedle (2002), o *Scrum* tem como objetivo gerenciar e controlar processos de desenvolvimento de software, focado em pessoas e que seja indicado para ambientes com frequente mudança de requisitos.

2.3.1 Papéis e responsabilidades no Scrum

O processo de desenvolvimento de software baseado na metodologia *Scrum* precisa englobar uma série de atividades e definir alguns papéis importantes.

Para que o software seja desenvolvido seguindo as características do *Scrum* é necessário que a equipe de desenvolvimento, denominada nesta metodologia como *Time Scrum* (*Scrum Team*), tenha papéis bem definidos. Estes papéis são três: *Product Owner*, *Scrum Master* e *Team* (Equipe).

O *Product Owner*, conhecido também como dono do produto, é responsável pela definição do projeto. O PO é o único que pode gerenciar os requisitos do produto, mudando as prioridades, adicionando ou removendo novos requisitos conforme as suas necessidades.

O *Scrum Master* tem a responsabilidade de manter o processo *Scrum* ativo. Este membro é focado em remover impedimentos e também responsável pela produtividade da equipe, fazendo com que a mesma foque no que deve ser feito.

O *Team* pode ser composto de 3 a 9 pessoas, não ultrapassando deste número com intuito de a equipe ser pequena, potencializando um dos valores do *Scrum* que é a comunicação.

Segundo Schwaber e Sutherland (2011) as equipes de desenvolvimento devem possuir as seguintes características:

1. São auto-organizadas. Ninguém (nem mesmo o *Scrum Master*) diz a Equipe de Desenvolvimento como transformar o *Backlog* do Produto em incrementos de funcionalidades potencialmente utilizáveis;
2. Equipes de Desenvolvimento são multifuncionais, possuindo todas as habilidades necessárias, enquanto equipe, para criar o incremento do Produto;
3. O *Scrum* não reconhece títulos para os integrantes da Equipe de Desenvolvimento que não seja o Desenvolvedor, independentemente do trabalho que está sendo realizado pela pessoa;
4. Individualmente os integrantes da Equipe de Desenvolvimento podem ter habilidades especializadas e área de especialização, mas a responsabilidade pertence à Equipe de Desenvolvimento como um todo;
5. Equipes de Desenvolvimento não contém sub-equipes dedicadas a domínios específicos de conhecimento, tais como teste ou análise de negócios.

2.3.2 Eventos do Scrum

O ciclo de vida de desenvolvimento do *Scrum* pode ser representado de uma forma simplificada pela Figura 4.

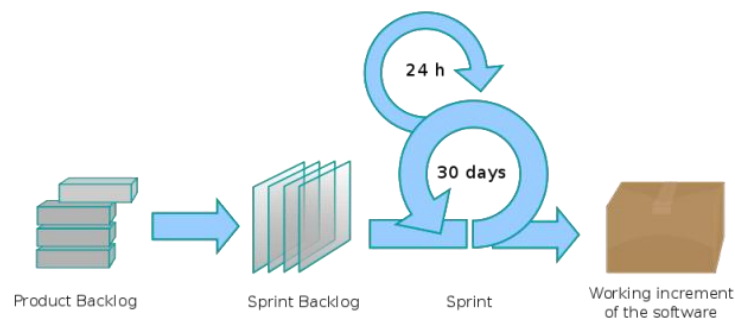


Figura 4. Ciclo de vida do Scrum

Fonte: Wikipedia (2014)

Inicialmente, durante o processo de desenvolvimento deste modelo, elabora-se uma série de requisitos do sistema a serem desenvolvidos, em que esta relação de requisitos é denominada *Product Backlog*.

O *Product Backlog* possui uma lista de itens priorizados pelo *Product Owner*, na qual está inclusa tudo o que precisa ser realizado, que possa ser associado com valor de negócio, para a finalização do projeto, abrangendo requisitos funcionais ou não.

De acordo com Schwaber e Sutherland (2011) o gerenciamento do *Backlog* do produto inclui: Expressar claramente os itens do *Backlog* do Produto; Ordenar os itens do *Backlog* do Produto para alcançar melhor as metas e missões; Garantir o valor do trabalho realizado pelo Time de Desenvolvimento; Garantir que o *Backlog* do Produto seja visível, transparente, claro para todos, e mostrar o que o Time *Scrum* vai trabalhar a seguir e; Garantir que a Equipe de Desenvolvimento entenda os itens do *Backlog* do Produto no nível necessário.

A partir do que for levantado e listado no *Product Backlog* surge o *Sprint Backlog*, que representa o que deverá ser colocado em prática durante a *Sprint* do projeto.

Segundo Nascimento (2012) a maioria dos itens que estão no *Product Backlog* serão implementados, mas para serem considerados para fazer parte do *Sprint Backlog* eles devem estar preparados, estimados e priorizados, segundo a definição de preparado estabelecida no início do projeto.

No *Scrum* o progresso do projeto baseia-se em iterações bem definidas, denominadas *Sprints*. Para cada *Sprint* recomenda-se uma duração de 2 a 4 semanas, período este chamado de *Time Box*.

Quatro reuniões são de bastante importância para uma *Sprint*:

1. *Reunião de Planejamento da Sprint*: o trabalho a ser realizado é definido nesta reunião;
2. *Reunião diária*: é um evento de aproximadamente 15 minutos, em que são criados o que a equipe irá fazer nas próximas 24 horas;
3. *Reunião de Revisão da Sprint*: nesta reunião é entregue o que foi feito ao longo da *Sprint* pela equipe ao *Product Owner*;
4. *Reunião de Retrospectiva*: executada após a reunião de revisão da *Sprint*, a equipe revê o que foi feito, quais foram os pontos positivos e negativos ocorridos, colaborando assim, para as próximas iterações.

Dentro das *Sprints* também existem o *Burndown Chart* da *Release* e da *Sprint*, que devem ser usados para o monitoramento das atividades realizadas. No gráfico exposto na Figura 5, observa-se o *Burndown Chart* da *Release*, em que no eixo X estão o número de *Sprints* e no eixo Y a quantidade de pontos do projeto.

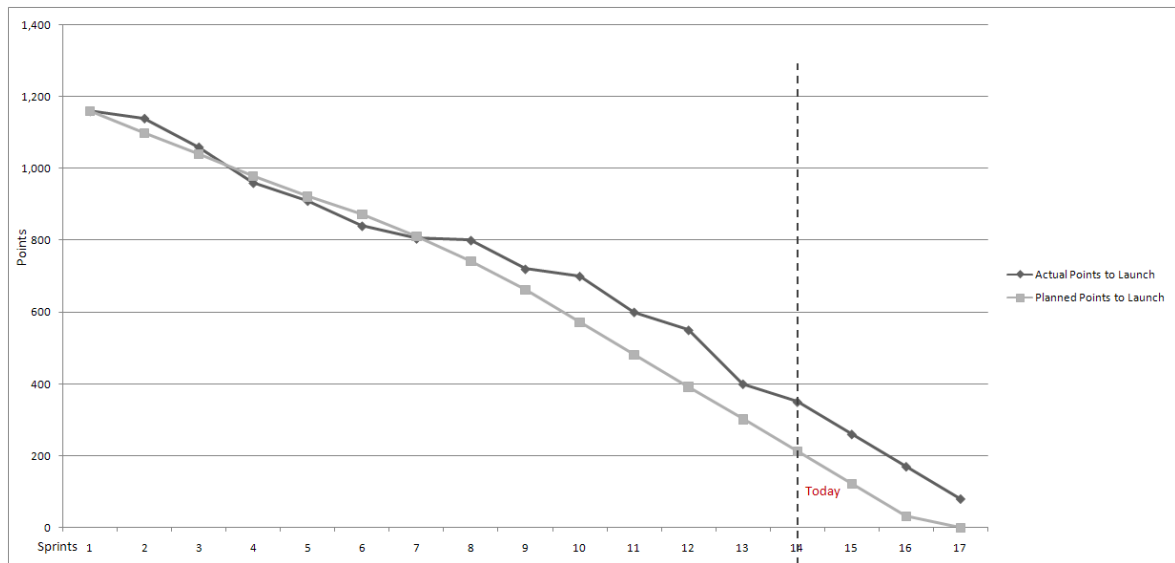


Figura 5. Burndown Chart

Fonte: All About Agile (2011)

Durante a fase de planejamento das *Sprints*, os membros da equipe *Scrum* atribuem valores (isto é, um peso) para cada iteração. Os valores planejados são apresentados no gráfico na cor cinza. Então, a medida em que se conclui uma nova *Sprint*, os membros atribuem uma nova quantidade de pontos, informando o que foi necessário de fato. Dessa forma, é possível ter uma visão analítica do andamento do projeto, além de permitir uma comparação entre o que foi planejado e o que de foi de fato executado. Esse tipo de artefato pode ser muito importante para o amadurecimento da empresa com relação ao planejamento de projetos subsequentes. Além disso, podem ser utilizados para advertir o quanto a equipe está atrasada em cada iteração.

2.3.3 Métricas e estimativas no Scrum

Segundo Pressman (2006) a mensuração é aplicada no processo de desenvolvimento de software ou atributos de um produto com o objetivo de melhorá-lo de forma contínua. Em que utilizando ao longo do projeto auxilia na estimativa, no controle de qualidade, na avaliação da produtividade e no controle do projeto.

No desenvolvimento ágil a estimativa do projeto é feita através da contagem de pontos, em que um ponto é uma maneira de comparar o esforço necessário para um requisito ser desenvolvido. Para estimar podem ser utilizadas algumas técnicas, a mais conhecida é o *Planning Poker*.

O *Planning Poker* usa o *Product Backlog* e um baralho de cartas para estimar. Essas cartas seguem uma sequência semelhante a de Fibonacci com os números 0, 1, 2, 3, 5, 8, 13, 20, 40 e 100, conforme mostra a Figura 6.



Figura 6. Cartas do *Planning Poker*

Fonte: IT-Zynergy ApS (2014)

No *Planning Poker*, inicialmente, as cartas ficam viradas para baixo e as funcionalidades são apresentadas uma a uma aos membros da equipe de desenvolvimento. Cada pessoa escolhe uma carta que represente o esforço que tal funcionalidade possui para ser desenvolvida e ao mesmo tempo todos da equipe viram suas cartas para cima, de modo que todos possam visualizá-las. Nesse momento, é feita a contagem dos pontos e, em seguida, esses pontos são analisados.

A ideia é discutir a variação de estimativa e colaborar com o entendimento e concepção que cada um tem sobre determinada funcionalidade, porque Fulano estimou X e Sicrano estimou Y. E no final a equipe chega a um consenso e define o peso da tarefa, partindo para a estimativa das demais, até que todas estejam estimadas (Mizerkowski, 2012).

2.4 Estudo de caso

O método científico pode ser definido como a maneira ou o conjunto de regras básicas empregadas em uma investigação científica com o intuito de obter resultados o mais confiáveis quanto for possível (FARIA, 2014). O mesmo surgiu da procura do homem em compreender sua realidade, para tanto, a racionalidade técnica do positivismo ofereceu orientação às pesquisas (DE DEUS 2010).

Dentre os métodos de pesquisa existentes encontra-se o estudo de caso, o qual segundo Fidel (1992) é um método específico de pesquisa de campo. Estudos de campo são investigações de fenômenos à medida que ocorrem, sem qualquer interferência significativa do investigador.

Yin (2010) relata que o estudo de caso é usado em muitas situações, para contribuir ao nosso conhecimento dos fenômenos individuais, grupais, organizacionais, sociais, políticos e relacionados.

2.4.1 Estrutura do estudo de caso

De acordo com Yin (2010) o estudo de caso se dá por um processo linear, mas iterativo, como mostrado na Figura 7, na qual cada fase representa uma evolução no processo do estudo de caso, porém existe a possibilidade de voltar a uma fase anterior para adequá-la a fase subsequente.

REALIZAÇÃO DA PESQUISA DE
ESTUDO DE CASO: UM PROCESSO
LINEAR, MAS ITERATIVO

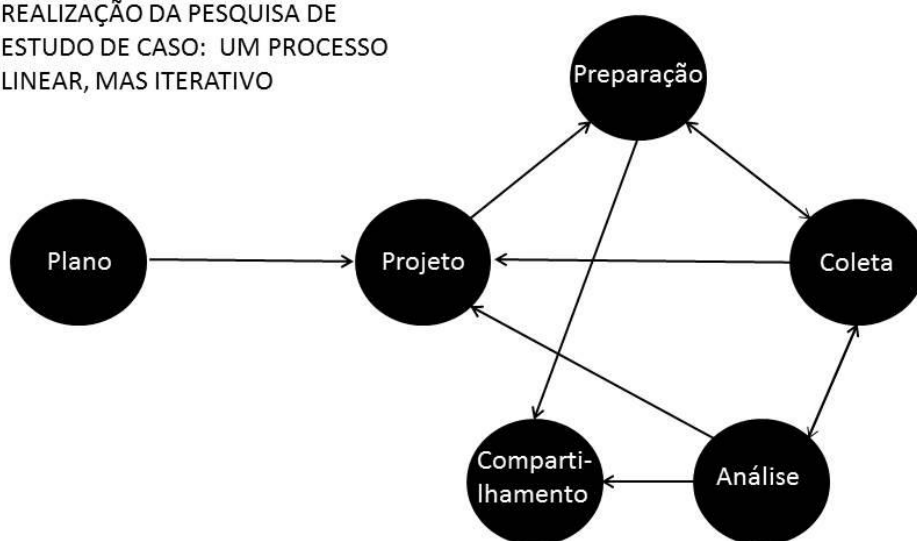


Figura 7. Processo do estudo de caso

Fonte: (YIN, 2010)

Para cada fase do processo do estudo de caso, Yin (2010) descreveu uma série de atividades que devem ser compreendidas e realizadas pelo pesquisador ao longo de sua pesquisa. As fases com suas respectivas descrições se apresentam no Quadro 1.

Quadro 1. Fases do estudo de caso e suas respectivas descrições.

Fase	Descrição
Plano	• Identificar as questões de pesquisa ou outra justificativa para a realização de um estudo de caso; • Decidir usar o método de estudo de caso, comparado com outros métodos; • Entender seus pontos fortes e suas limitações.
Projeto	• Definir a unidade de análise e os prováveis casos para estudo. Em que a unidade de análise está relacionada com o problema fundamental de definir o que é o “caso”. Um caso pode ser um indivíduo ou vários indivíduos, estes últimos poderiam ser incluídos em um estudo de casos múltiplos; • Desenvolver a teoria, as proposições e os assuntos subjacentes ao estudo antecipado; • Identificar o projeto do estudo de caso (único, múltiplo, holístico e integrado); • Definir os procedimentos para manter a qualidade do estudo de caso.
Preparação	• Ampliar as habilidades como investigador de estudo de caso; • Treinar para o estudo de caso específico; • Desenvolver o protocolo do estudo de caso; • Conduzir um caso-piloto; • Obter aprovação para a proteção dos sujeitos humanos.
Coleta	• Seguir o protocolo do estudo de caso;

	• Usar múltiplas fontes de evidência; • Criar um banco de dados do estudo de caso; • Manter um encadeamento de evidências.
Análise	• Contar com proposições teóricas e outras estratégias; • Considerar qualquer uma das cinco técnicas analíticas, usando dados quantitativos, qualitativos ou ambos; • Explorar as explicações rivais; • Apresentar os dados separados das interpretações.
Compartilhamento	• Definir o público; • Compor os materiais textuais e visuais; • Apresentar evidência suficiente para o leitor alcançar suas próprias conclusões; • Revisar e reescrever até estar bem feito.

Na fase de Projeto alguns procedimentos são importantes para manter a qualidade do trabalho, quatro testes são comuns a todos os métodos de ciência social (YIN, 2010):

- Validade de constructo: identificação das medidas operacionais corretas para os conceitos sendo estudados. Uma tática para a validade de constructo é utilizar fontes múltiplas de evidências de maneira que encoraje linhas convergentes de investigação.
- Validade interna: é principalmente uma preocupação dos estudos de caso explanatórios (quando o investigador tenta explicar como e por que o evento x levou ao evento y). Neste caso o investigador pode concluir erroneamente uma relação causal entre x e y. Para evitar colocar a validade interna em risco algumas táticas podem ser utilizadas como: combinação de padrão, construção de explicações, tratamento de explicações rivais e uso de modelos lógicos.

- Validade externa: está relacionada a definir o domínio para o qual as descobertas do estudo podem ser generalizadas. Por exemplo, o resultado de um estudo da mudança em um bairro é replicável a outro bairro? Se um estudo focasse a transição populacional na vizinhança urbana, o procedimento para a seleção da vizinhança teria começado com a identificação de uma em que as transições estivessem ocorrendo. As teorias sobre transição seriam o domínio ao qual os resultados poderiam mais tarde ser generalizados.
- Confiabilidade: demonstração de que as operações de um estudo podem ser repetidas com os mesmos resultados. Por isso se um pesquisador, posteriormente, seguir o mesmo procedimento descrito pelo primeiro, e conduzir o mesmo estudo de caso novamente, ele deverá obter os mesmos achados e conclusões. Por este motivo é importante a documentação dos procedimentos.

Na Preparação, como mostra o Quadro 1, é desenvolvido o protocolo e neste deve estar incluso:

1. Uma visão geral do projeto do estudo de caso;
2. Procedimentos de campo;
3. Questões que o investigador deve ter em mente na coleta dos dados e potenciais fontes que respondam as questões;
4. Guia para o relatório do estudo de caso (esboço, formato para os dados, informações bibliográficas).

Na fase de Coleta a evidência do estudo de caso pode vir de várias fontes: documentos, registros em arquivo, entrevista, observação direta, observação participante e artefatos físicos.

Utilizar fontes múltiplas de evidência e estabelecer o encadeamento entre elas também contribui para a validade do constructo, que corresponde ao estabelecimento de medidas operacionais corretas para os conceitos que estão sob estudo (YIN, 2010).

Durante a Análise as proposições são orientações teóricas que guiam a análise do estudo de caso ajudando a focar em determinados dados, proposições teóricas derivadas de questões “por que” e “como”, por exemplo, podem ser extremamente úteis neste sentido.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta alguns trabalhos relacionados com a pesquisa proposta, contendo visões e análises sobre a adoção do *Scrum* em empresas do Brasil. A seleção dos trabalhos foi realizada através de buscas manuais considerando algumas palavras chaves como “metodologias ágeis”, “adoção do *Scrum*”, “adoção do *Scrum* no Brasil”, “implantação da metodologia *Scrum*” e “estudo de caso da adoção do *Scrum*”, e após isso foram feitas análises subjetivas, selecionando algumas pesquisas condizentes com o assunto tratado, de forma a contribuir com o mesmo.

Os trabalhos apresentados têm relação com a implantação da metodologia *Scrum* em diferentes organizações e localidades do Brasil, assim como, perfis de equipes distintas.

3.1 Scrum no Brasil

A contribuição literária de Vargas (2011) foi constituída de uma pesquisa com procedimentos de cunho bibliográfico, por ter sido baseada em materiais como livros, artigos, e publicações periódicas. Vargas (2011) também elaborou um questionário, o qual foi entregue para 35 empresas do ramo de desenvolvimento de software na cidade de Florianópolis-SC. Seu objetivo foi identificar quais práticas do *Scrum* são aplicadas de forma efetiva nas empresas investigadas e, como a metodologia ágil *Scrum* pode contribuir para a evolução das empresas de desenvolvimento de softwares.

Todas as questões do questionário foram respondidas de acordo com o que a empresa utilizava. Apenas as questões que não tinham relação com o estilo de desenvolvimento das mesmas não foram respondidas.

Houve uma grande dificuldade em conseguir que as empresas respondessem o questionário, demonstrando não querer revelar muito de como trabalham, muitas vezes questionando em que empresa o pesquisador trabalhava, não importando para qual instituição de ensino estava sendo desenvolvida a pesquisa.

Dentre as empresas entrevistadas, 64% utilizavam a metodologia *Scrum*, enquanto 36% usavam outras metodologias. Além disso, também foram apresentadas quatorze questões sobre as características da metodologia ágil para que os entrevistados pudessem escolher quais faziam parte do seu processo de desenvolvimento atual. A Tabela 2 apresenta as características investigadas por Vargas (2011) e o resultado obtido por esse questionamento.

Tabela 2. Características do *Scrum* adotadas na empresa

Características do Scrum	Marcações
Regras Scrum	3
Burndown de Sprint	6
Burndown de Release	6
Retrospectiva da Sprint	9
Reunião da Sprint	7
Reuniões de Planejamento da Release e da Sprint	15
Sprint Backlog	11
Product Owner	8
Product Backlog	12
Time Scrum	7
Scrum Master	7
Reuniões diárias	19
Sprint	15
Outro	0

Em relação ao quesito sobre a melhoria na qualidade do produto final a partir da adoção da metodologia *Scrum*, 90% afirmaram que houve um ganho significativo, enquanto 10% atestaram que não houve melhoria alguma.

O trabalho de Vargas (2011) apresentou as características do *Scrum* presentes nas empresas, porém, não foi realizada uma pesquisa se essas características eram compreendidas pelas pessoas questionadas e se as mesmas sabiam o real objetivo de se adotar tais práticas como visa este trabalho.

3.2 Aplicação de Scrum em Ambiente de Desenvolvimento de Software Educativo

Vasconcelos (2010) realizou um estudo de caso em um ambiente de desenvolvimento de softwares educativos apresentados em forma de aulas multimídia. Foram analisadas as características do ambiente da empresa e correlacionadas com a metodologia *Scrum* e após isso foi possível definir a implantação do método ágil na empresa.

Depois da implantação do *Scrum* na empresa foi feito um comparativo entre as práticas anteriores e posteriores à adoção do *Scrum*. Um resumo desse comparativo pode ser visualizado na Tabela 3.

Tabela 3. Resumo comparativo da avaliação do ambiente antes e após a implantação do método *Scrum*. Fonte: Vasconcelos (2010)

Ambiente Pré-Implantação	Ambiente Pós-Implantação
Hierarquia da equipe composta pelas três instâncias: alta gerência, coordenador de equipe, equipe.	Hierarquia da equipe composta pelas três instâncias: Product Owner, Scrum Master e Scrum Team.
Papel da alta gerência: gerenciar os coordenadores de equipe e contatar os clientes. A alta gerência não participa do desenvolvimento em nenhuma etapa.	Papel do Product Owner: colaborar com o Scrum Master e a equipe na seleção e manutenção das prioridades do desenvolvimento de acordo com o valor de negócio da empresa. Além de contatar os clientes.
Papel do coordenador de equipe: líder. Responsável por guiar a equipe para obter resultados de acordo com suas próprias definições do produto e premissa básicas determinadas pela empresa. Responsável por remover as dúvidas frequentes da equipe quanto ao processo de desenvolvimento, visto que este é definido por sua experiência do coordenador.	Papel do Scrum Master: facilitador. É responsável por remover os impedimentos da equipe no processo de desenvolvimento, não sendo responsável por definir esse processo, mas por assegurar que a metodologia Scrum seja seguida quanto às etapas, aos artefatos e papéis.
Papel da equipe: desenvolver os objetos de aprendizagem de acordo com a	Papel do Scrum Team: É responsável por ser auto-organizada e por selecionar os itens

documentação de roteiro gerada e as instruções do coordenador e reportar todas as dúvidas e problemas ao coordenador, sempre que surgirem. Possui total dependência do coordenador para sua organização.	priorizados que irão ser executados em cada sprint, com total liberdade e comprometimento para desenvolver os objetos de aprendizagem de acordo com as etapas definidas pela metodologia Scrum; bem como responsável por reportar os impedimentos encontrados (através do Scrum Board) para que o Scrum Master os remova. É também papel da equipe participar ativamente das reuniões diárias e de revisão e retrospectiva.
Processo de desenvolvimento: iterações sem etapas definidas ou delimitadas.	Processo de desenvolvimento: sprints com etapas pré-definidas e obrigatórias.
Ciclo de desenvolvimento: produção do roteiro, seguida do desenvolvimento (com testes periódicos, mas sem padronização) e publicação do objeto de aprendizagem.	Ciclo de desenvolvimento: produção do roteiro, seguida da Sprint Planning Meeting (para validação do roteiro com equipe e Product Owner) e do desenvolvimento (com verificações diárias – Daily Scrum Meetings). O fim do desenvolvimento é seguido pela execução da Sprint Review (para validação do Product Owner) e a publicação do objeto de aprendizagem.

Na pesquisa feita por Vasconcelos (2010) foi notada a boa aceitação da metodologia *Scrum* e os papéis foram bem compreendidos e desempenhados pela equipe. Porém, mais características da metodologia como, a participação e satisfação do cliente e, a estimativa da quantidade de esforço da equipe, não foram informadas para se saber como a compreensão da equipe se deu em relação a esses aspectos. Sendo este um fator diferencial neste trabalho, já que o mesmo abrange esses pontos mencionados.

3.3 Implantando o SCRUM em um Ambiente de Desenvolvimento de Produtos para Internet

Varaschim (2009) investigou a implantação da metodologia *Scrum* em uma empresa de internet, onde o cenário da empresa antes da adoção era departamental, em que todas as áreas de desenvolvimento (Design, Arquitetura da Informação, Desenvolvimento, Teste) eram separadas e a interface entre as áreas tinha certos requisitos incompatíveis com a velocidade desejada.

O ambiente de desenvolvimento de software para a internet possui características similares ao de um modelo de gerenciamento de projetos ágil. Esta similaridade faz com que sua adoção de métodos ágeis vá ao encontro das necessidades das empresas que estão neste mercado.

A implantação do *Scrum* na empresa investigada começou com uma equipe de quinze pessoas e logo após em toda a empresa. Essa equipe iniciou com o desenvolvimento de um projeto de ampla visibilidade com um grande potencial de acesso por parte do usuário final. Segundo Varaschim (2009), os primeiros resultados observados após a adoção do método de desenvolvimento ágil foram:

1. No segundo *Daily Scrum* a primeira história foi concluída;
2. Percepção de maior velocidade;
3. Faltou planejamento no primeiro *Sprint*;
4. Apenas metade do planejado foi desenvolvido mesmo com a percepção de maior velocidade;
5. Sobraram recursos, algumas pessoas não tinham tarefas para fazer;
6. Equipe resolveu melhorar planejamento e diminuir a equipe para nove pessoas.

Com a adoção da metodologia algumas dificuldades surgiram, dentre elas: (1) Falta de treinamento da equipe; (2) Falta de definição dos papéis do *Product Owner* e do *Scrum Master* e (3) Falta de *Product Backlog* Priorizado.

No estudo feito por Varaschim (2009) é possível observar que foram investigados alguns resultados após a adoção do *Scrum* na empresa e algumas dificuldades encontradas no

decorrer dessa adoção. Entretanto, não foi relatado em sua pesquisa, como foi feito nesta, o que os membros da equipe compreendiam sobre as características da metodologia e também não foram relatados outros aspectos relacionados no *Scrum*, como, o uso dos artefatos, como foi realizado o planejamento e mais detalhes sobre o que acontecia ao longo das *sprints*.

Dado o que foi exposto neste capítulo, o que diferencia este trabalho dos estudos apresentados é que apesar de os mesmos apresentarem as características do *Scrum* adotadas nas empresas e o que aconteceu de diferente após a adoção, não apresentam o que os desenvolvedores compreenderam do uso das práticas da metodologia, ou seja, se eles entendem o objetivo e para que servem tais práticas.

4 ESTUDO DE CASO EM UMA EMPRESA PRIVADA DE DESENVOLVIMENTO DE SOFTWARE

4.1 Delineamento da pesquisa

De maneira a alcançar os objetivos almejados, o método utilizado foi um estudo de caso de caráter descritivo em uma empresa de desenvolvimento de software na cidade de João Pessoa, Paraíba.

A finalidade da pesquisa é analisar a adesão do método ágil *Scrum* para a empresa sob a visão dos desenvolvedores, através de dados qualitativos em conjunto com dados quantitativos obtidos por meio dos instrumentos de coleta realizados com a equipe de desenvolvimento.

Segundo Zelkowitz e Wallace (1998), estudo de caso é um estudo observacional em que o pesquisador monitora projetos em profundidade e coleta dados ao longo do tempo.

Para Yin (2001), o estudo de caso permite entender a dinâmica presente em unidades organizacionais, podendo envolver um único caso como também casos múltiplos e diferentes níveis de análise.

Uma pesquisa descritiva tem por finalidade observar, registrar e analisar a ocorrência dos fatos. Possui o objetivo de descrever as características de determinadas populações ou fenômenos. Uma de suas peculiaridades está na utilização de técnicas padronizadas de coleta de dados, tais como o questionário e a observação sistemática (GIL, 2008).

4.2 Unidade de análise

Unidades de análise, segundo Siglenton (1988), são os objetos ou eventos aos quais as pesquisas sociais se referem, o que ou quem será descrito, analisado ou comparado.

Esta pesquisa foi realizada em uma empresa privada de desenvolvimento de software, que para este trabalho é uma unidade de análise. A mesma está no mercado há aproximadamente sete anos, atuando no desenvolvimento de sistemas para Juntas Comerciais,

Prefeituras, Corpo de Bombeiros, Vigilâncias Sanitárias e Secretarias da Fazenda para diversos estados do Brasil, como, Alagoas, Minas Gerais, Amazonas, Amapá, Paraíba e Rio Grande do Norte.

Anteriormente encontrava-se sediada em Brasília, Distrito Federal e atualmente está localizada na cidade de João Pessoa, estado da Paraíba.

A empresa analisada desenvolve três sistemas principais, cuja descrição sintetizada encontra-se elencada a seguir:

- Sistema A: tem o objetivo de facilitar a abertura, alteração e baixa de empresas, gerando assim uma maior segurança e agilidade para o poder público e para o empreendedor. Este sistema é uma solução *web* que faz a integração entre o Cadastro Sincronizado Nacional² e os diversos órgãos estaduais e municipais envolvidos nesses processos. A plataforma disponibilizada na internet permite ao empreendedor, de modo ágil e centralizado, dar andamento aos processos de forma otimizada, racionalizada e com burocracia reduzida;
- Sistema B: este sistema tem como principal objetivo a publicação de matérias no Diário Municipal. Essas matérias são cadastradas via *web* pelas Prefeituras e logo após analisadas por pessoas especializadas na empresa, formatando as mesmas de uma maneira padronizada;
- Sistema C: também uma solução *web*, refere-se à emissão de notas fiscais eletrônicas.

Inicialmente, houve na organização uma tentativa de utilizar como metodologia um conjunto de boas práticas do *Scrum*, XP e RUP. Todavia, existiam algumas dificuldades no exercício de se adotar essas práticas, exercendo-se apenas algumas características básicas de cada método. Esses problemas ocorreram principalmente pelo motivo de não haver tempo para treinamento dos membros da equipe e tentativa de adaptação das metodologias de maneira que a equipe achou adequada.

A proposta da adoção do *Scrum* em sua forma plena, seguindo as características indicadas pela metodologia, surgiu em março de 2013, ideia de uma desenvolvedora em

² O Cadastro Sincronizado Nacional é a integração dos procedimentos cadastrais de pessoas jurídicas e demais entidades no âmbito das Administrações Tributárias da União, dos Estados, do Distrito Federal e dos Municípios, bem como de outros órgãos e entidades que fazem parte do processo de registro e legalização de negócios no Brasil.

conjunto com o analista de sistemas, que na época já possuía *Certificação Scrum Product Owner*.

Atualmente existem quatro equipes divididas na empresa, cada uma com tarefas específicas, são elas: (1) Suporte; (2) Sustentação; (3) Arquitetura; (4) Equipe *Scrum*. Possui ainda dois DBA's, um designer, um gerente de TI e o dono da empresa, os quais participam de atividades em todas as equipes.

A equipe *Scrum*, o foco do estudo, é formada por cinco membros e desempenha o papel de desenvolver novas funcionalidades para os três sistemas existentes na empresa com o prazo de entrega de quatro semanas.

Na equipe *Scrum* o Analista de Sistemas tem o papel de *Product Owner*, o *Scrum Master* é um desenvolvedor, e a Equipe é composta por um *Tester* e dois desenvolvedores.

4.3 Coleta de dados

Para garantir a validade de constructo, validade interna e a confiabilidade do trabalho, duas fontes de evidências foram utilizadas para coletar os dados relevantes para a pesquisa e avaliá-los de acordo com o objetivo da mesma: 1) Questionário; 2) Observação direta do pesquisador ao longo do estudo.

Foram elaborados também um protocolo de estudo de caso (ver Apêndice A) apresentando detalhes sobre o projeto e um roteiro de observação (ver Apêndice B) descrevendo itens a serem observados e anotados pelo pesquisador no período da pesquisa.

O objetivo do protocolo é orientar o pesquisador nas atividades de coleta de dados, estabelecendo as regras gerais que serão seguidas em campo (Yin, 2009).

O roteiro de observação sistemática é uma das técnicas utilizadas para levantar dados e aspectos referentes à empresa e/ou do assunto pesquisado.

Com o propósito de analisar as características da metodologia ágil *Scrum* adotadas na empresa sob o ponto de vista da equipe, foi elaborado um questionário (ver Apêndice C) contendo 32 questões objetivas. O questionário se encontra dividido em quatro categorias, as

quais são elencadas abaixo, em que cada uma delas possui perguntas que pretendem responder quais os pontos do *Scrum* são atendidos.

- Planejamento: esta categoria tem como objetivo responder se há um planejamento efetivo no projeto. Refere-se a como ocorrem as estimativas, as atividades realizadas e a velocidade do time;
- Artefatos: visa saber se os artefatos estão definidos e se os mesmos estão sendo utilizados de maneira adequada;
- Papéis e responsabilidades: tem como objetivo responder se os papéis são bem definidos e conhecidos dentro da equipe;
- *Sprint*: nesta categoria o objetivo encontra-se em saber se a *Sprint* é executada de maneira adequada dentro do planejamento.

Para facilitar a definição e planejamento da pesquisa foi utilizado o método GQM (*Goal-Question-Metric*), cuja ideia básica é derivar métricas de software a partir de perguntas e objetivos (BASILI; CALDIERA, 2002). O significado do GQM é:

- *Goal*

- Quais são as metas/objetivos?

- *Question*

- Quais questões se deseja responder?

- *Metric*

- Quais métricas poderão ajudar?

Para este trabalho foram utilizados o objetivo e as questões abaixo, como foi visto no Capítulo 1, Seção 1.2. :

- Objetivo:

Analisar a utilização do *Scrum* **com relação ao** planejamento, papéis, artefatos e *sprint* **com o objetivo** de verificar a percepção da aderência ao *Scrum* **sob o ponto de vista** dos desenvolvedores **no contexto de** uma empresa privada de software.

- Questões:

Q1. Que aspectos do *Scrum* estão sendo adotados na empresa analisada?

Q2. Qual a percepção da equipe em relação ao planejamento do projeto?

Q3. Qual a percepção da equipe em relação aos artefatos utilizados?

Q4. Qual a percepção da equipe em relação aos papéis?

Q5. Qual a percepção da equipe em relação a *Sprint*?

Na Tabela 4 se apresentam as perguntas presentes no questionário e os seus respectivos objetivos.

Tabela 4. Perguntas e seus respectivos objetivos.

Planejamento	Objetivo
No início do processo de desenvolvimento é elaborada uma lista de requisitos a serem desenvolvidos.	Saber se é elaborado o <i>Product Backlog</i> .
Para as funcionalidades requeridas pelo cliente é feita uma priorização dos requisitos a serem desenvolvidos.	Saber se os requisitos são priorizados pelo <i>Product Owner</i> .
Para cada requisito é estimada uma quantidade de esforço da equipe.	Saber se existe a estimativa de esforço da equipe.
Toda a equipe participa da estimativa dos requisitos a serem desenvolvidos.	Saber se todos os membros da equipe participam da estimativa.
A estimativa é feita através da contagem de pontos e para isso utiliza-se um baralho de cartas.	Saber se é utilizado o <i>Planning Poker</i> .
O projeto é planejado para ter entregas parciais ao cliente.	Saber se as entregas são divididas em iterações/ <i>sprints</i> .
O projeto segue o modelo iterativo e incremental onde as iterações são chamadas de <i>Sprints</i> .	Saber se as iterações são conhecidas como <i>Sprints</i> .
Os membros da equipe se reúnem no início de cada iteração para planejar as tarefas a serem realizadas.	Saber se é realizada a Reunião de Planejamento da <i>Sprint</i> .
A equipe conhece sua produtividade de maneira quantitativa e utiliza esse valor para planejar o quanto entregar em cada <i>Sprint</i> .	Saber se a equipe tem conhecimento de sua produtividade.
Artefatos	
A ferramenta para visualização do trabalho em andamento está disponível em um quadro na sala.	Saber se o quadro está visível para todos da equipe em um quadro na sala.
A ferramenta para visualização do trabalho em andamento é atualizada todos os dias pelos membros da equipe.	Saber se o <i>Scrum Board</i> é atualizado todos os dias.
O quadro onde estão contempladas todas as tarefas que serão realizadas de acordo com a prioridade de cada item é chamado <i>Scrum Board</i> .	Saber se o quadro é conhecido por todos da equipe como <i>Scrum Board</i> .
Existe um gráfico em que a equipe visualize o andamento do projeto.	Saber se existe o <i>Burndown Chart</i> .
O gráfico para visualização do andamento de todo o projeto está exposto em um quadro.	Saber se todos da equipe visualizam o <i>Burndown Chart</i> .
Existe um documento onde os requisitos são registrados contendo sua descrição.	Saber se no <i>Product Backlog</i> são descritas as Estórias.

O documento de requisitos é denominado <i>Product Backlog</i> .	Saber se o artefato é conhecido por todos em relação ao <i>Scrum</i> .
A equipe consegue visualizar e acompanhar o andamento do projeto diariamente para verificar se estão dentro do estimado.	Verificar se o gráfico <i>burndown</i> é utilizado para acompanhamento do projeto.
Papéis e responsabilidades	
Existe uma pessoa na equipe responsável por resolver os impedimentos (problemas) que surgem.	Verificar se existe o papel de <i>Scrum Master</i> definido e compreendido pela equipe.
O membro da equipe responsável por remover os impedimentos é chamado de <i>Scrum Master</i> .	Saber se o papel é conhecido por todos.
O <i>Scrum Master</i> atua como gerente da equipe.	Verificar se o <i>Scrum Master</i> atua como gerente, pois no <i>Scrum</i> este papel não deve ter essa responsabilidade.
O <i>Scrum Master</i> também participa da implementação do software.	Saber se o <i>Scrum Master</i> desempenha mais de uma função.
O membro da equipe responsável pela lista de requisitos é denominado <i>Product Owner</i> .	Saber se existe o papel de <i>Product Owner</i> e se a função do mesmo é compreendida pela equipe.
A equipe é capaz de deter todo o poder na tomada de decisões no que diz respeito a implementação dos requisitos.	Verificar se a equipe possui uma das características abrangidas pelo <i>Scrum</i> que é ser auto-organizada.
O cliente é visto como membro da equipe.	Saber se o cliente tem participação junto à equipe.
Sprint	
Existem reuniões diárias entre a equipe no início ou após o expediente para conhecer as atividades que foram realizadas e as que serão realizadas por cada um.	Saber se a reunião diária está sendo realizada.
Diariamente a equipe fica sabendo sobre os impedimentos que cada um está enfrentando.	Saber se nas reuniões diárias são apontados os impedimentos existentes para a equipe executar suas funções.
Ao final de cada iteração, há uma reunião da equipe com o cliente para apresentar o que foi desenvolvido.	Verificar se são realizadas Reuniões de Revisão da <i>Sprint</i> .
Ao final de cada iteração, a equipe se reúne para avaliar os pontos positivos e negativos ocorridos.	Verificar se são realizadas Reuniões de Retrospectiva.
São realizadas reuniões complementares para solucionar impedimentos quando em outras reuniões não surge uma solução.	Saber se são realizadas reuniões se surgirem problemas que não foram solucionados em nenhuma outra reunião.
Existem entregas periódicas ao cliente em um prazo de 2 a 4 semanas.	Verificar se as entregas são feitas no prazo estipulado pelo <i>Scrum</i> .
Dentro de um projeto, o tempo de uma iteração não é alterado.	Verificar se a entrega é realizada sem haver alterações do que se foi planejado, garantindo assim que não ocorrerão atrasos.
Os requisitos desenvolvidos nas iterações representam de fato o que o cliente deseja para aquele momento específico.	Verificar se realmente as iterações estão de acordo com a necessidade do cliente.

Para as respostas das questões foi utilizada a escala de *Likert* como métrica. Esta escala é bastante usada em pesquisas de opinião e ao responderem as perguntas baseadas

nesta escala, os respondentes apontam o seu nível de concordância dado uma determinada afirmação.

Nas escalas de *Likert*, ou escalas Somadas, atribuem-se valores numéricos e/ou sinais às respostas para refletir a força e a direção da reação do entrevistado à declaração. As declarações de concordância devem receber valores positivos ou altos enquanto as declarações das quais discordam devem receber valores negativos ou baixos (BAKER, 2005).

Para esta pesquisa os valores atribuídos para as respostas foi uma pontuação de 1 a 5, conforme mostra a Tabela 5.

Tabela 5. Escala de Likert e pontuação para as respostas

Escala de Likert	Pontuação
Concordo totalmente	5
Concordo parcialmente	4
Não concordo nem discordo	3
Discordo parcialmente	2
Discordo totalmente	1

O questionário foi disponibilizado na *Internet* através de um formulário entre o final do mês de março de 2014 e o início do mês de maio do mesmo ano. Os respondentes foram os cinco membros da equipe *Scrum*, os quais eram os únicos que tinham o conhecimento necessário para responder as questões, pois os mesmos eram os que participavam na prática da adoção da metodologia. É importante ressaltar que a pesquisadora não fazia parte da equipe e a mesma não respondeu as questões.

A equipe *Scrum* foi observada no período de 01 de outubro de 2013 até 01 de abril de 2014 no que se refere às práticas da metodologia *Scrum* adotadas na empresa e como ocorre o processo de comunicação entre os membros da equipe. Foram observados também alguns detalhes como, por exemplo, a localização dos membros na sala e onde ficavam anotados e expostos os artefatos. Para um melhor entendimento da disposição da sala da empresa onde se encontrava a equipe, segue uma descrição gráfica abaixo. Vale a pena ressaltar que o observador não se encontrava nessa sala, apenas entrava na mesma algumas vezes ao dia para observar.

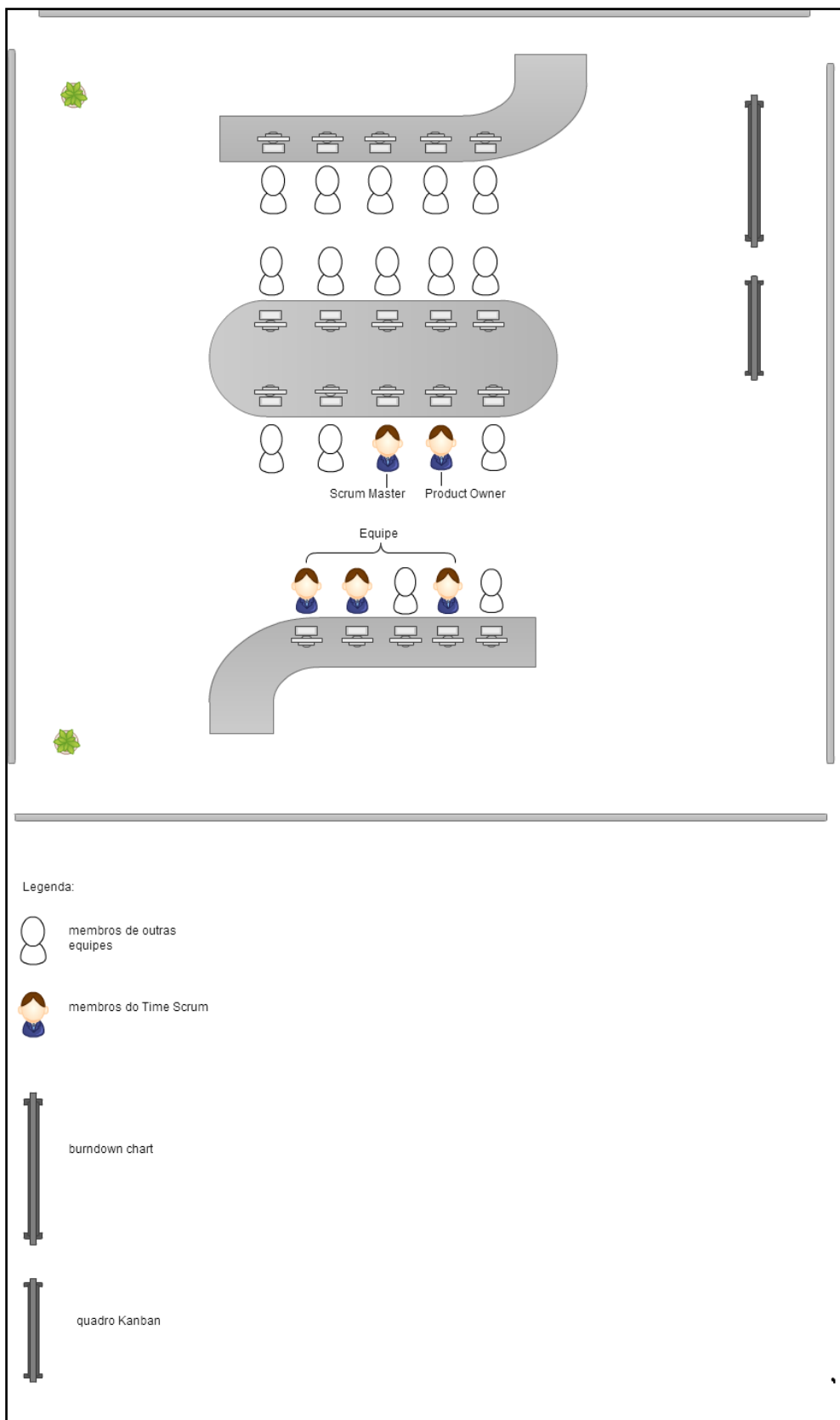


Figura 8. Sala da empresa

Para realização da observação contou-se com a utilização do diário de observação, identificado como um instrumento de bastante importância para o observador, e que consiste em um instrumento para o registro de informações que emergem do trabalho de campo e que posteriormente serão utilizadas pelo pesquisador ao fazer a análise dos dados.

4.4 Aplicação do piloto

Antes de o questionário ser aplicado na empresa à equipe *Scrum*, foi realizado um teste-piloto do questionário em outra empresa que também está adotando essa metodologia no processo de desenvolvimento de software. Através do teste-piloto é possível obter um feedback para saber se a redação e a clareza do questionário estão evidentes para todos os questionados e se as questões fazem o mesmo sentido para todos os questionados.

Segundo Iraossi (2006) as três metas básicas do teste-piloto são:

- Avaliar a competência do questionário.
- Estimar a duração do questionário ou o tempo necessário para fazê-lo.
- Determinar a qualidade do pesquisador.

O questionário foi respondido por duas pessoas da equipe de desenvolvimento de uma empresa em João Pessoa que também se encontra em fase de adoção do *Scrum* e o mesmo se deu na forma online através de um formulário do Google. O tempo para respondê-lo foi de aproximadamente 10 minutos.

Após o questionário ser respondido pelos dois participantes foram realizadas melhorias em algumas questões para uma maior compreensão dos respondentes da empresa a ser analisada.

5 Análise e discussão dos dados

Para a análise dos dados foi realizada uma abordagem qualitativa feita através das descrições obtidas do diário de observação, e uma abordagem quantitativa visando estabelecer o *Ranking* Médio (RM) proposto por Oliveira (2005) para as respostas do questionário. Para o questionário que utilizou a escala de *Likert* de 5 pontos para mensurar o grau de concordância, atribuiu-se um valor de 1 a 5 para cada resposta, e a partir deste valor é realizado o cálculo do somatório para cada item, baseando-se na frequência das respostas. Os valores menores que 3 são considerados como discordantes, os maiores que 3 como concordantes e, o valor 3 é considerado “indiferente”. Desta forma, foi obtido o RM através da seguinte estratégia:

- Somatório (S) = $\sum (f_i \cdot V_i)$
- Ranking Médio (RM) = $S / (NS)$
- f_i = frequência observada de cada resposta para cada item
- V_i = valor de cada resposta
- NS = número de sujeitos (frequência de respostas)

O exemplo do cálculo do RM (*Ranking* Médio) pode ser visto na Tabela 6:

Tabela 6. Exemplo da avaliação das respostas

Fonte: adaptado de OLIVEIRA (2005).

QUESTÕES	FREQUÊNCIA DE RESPOSTAS					RM
	1	2	3	4	5	
Para cada requisito é estimada uma quantidade de esforço da equipe.		1	1	1	2	3,2

$$\text{Somatório} = (1 \times 2) + (1 \times 3) + (1 \times 4) + (2 \times 5) = 19$$

$$\text{Logo, RM} = 19 / (1 + 1 + 1 + 2) = 3,2$$

A Tabela 7 mostra a frequência das respostas e o Ranking Médio (RM) estabelecido para cada questão.

Tabela 7. Frequência das respostas e Ranking Médio do questionário

QUESTÕES	FREQUÊNCIA DE RESPOSTAS					
	1	2	3	4	5	RM
No início do processo de desenvolvimento é elaborada uma lista de requisitos a serem desenvolvidos.					5	5
Para as funcionalidades requeridas pelo cliente é feita uma priorização dos requisitos a serem desenvolvidos.			1	3	1	4
Para cada requisito é estimada uma quantidade de esforço da equipe.		1	1	1	2	3,2
Toda a equipe participa da estimativa dos requisitos a serem desenvolvidos.	2	1			2	4,4
A estimativa é feita através da contagem de pontos e para isso utiliza-se um baralho de cartas.	2	1			2	4,4
O projeto é planejado para ter entregas parciais ao cliente.	1	1		2	1	3,2
O projeto segue o modelo iterativo e incremental onde as iterações são chamadas de <i>Sprints</i> .				2	3	4,6
Os membros da equipe se reúnem no início de cada iteração para planejar as tarefas a serem realizadas.		1		1	3	4,2
A equipe conhece sua produtividade de maneira quantitativa e utiliza esse valor para planejar o quanto entregar em cada <i>Sprint</i> .	1	3		1		2,6
A ferramenta para visualização do trabalho em andamento está disponível em um quadro na sala.		2		1	2	3,6
A ferramenta para visualização do trabalho em andamento é atualizada todos os dias pelos membros da equipe.		1		2	2	3,8
O quadro onde estão contempladas todas as tarefas que serão realizadas de acordo com a prioridade de cada item é chamado <i>Scrum Board</i> .				1	4	4,8
Existe um gráfico para que a equipe visualize o andamento do projeto.		1	1	1	2	4
O gráfico para visualização do andamento de todo o projeto está exposto em um quadro.	1	1		1	2	3,4
Existe um documento onde os requisitos são registrados contendo sua descrição.		1		1	3	4,2
O documento de requisitos é denominado <i>Product Backlog</i> .	1	1			3	3,6
A equipe consegue visualizar e acompanhar o andamento do projeto diariamente para verificar se estão dentro do estimado.		3		1	1	3
Existe uma pessoa na equipe responsável por resolver os impedimentos (problemas) que surgem.		1		2	2	4
O membro da equipe responsável por remover os impedimentos é chamado de <i>Scrum Master</i> .	1			2	2	3,8
O <i>Scrum Master</i> atua como gerente da equipe.	2	1		1	1	2,6
O <i>Scrum Master</i> também participa da implementação do software.	1			2	2	4
O membro da equipe responsável pela lista de requisitos é denominado <i>Product Owner</i> .			1	1	3	4,4
A equipe é capaz de deter todo o poder na tomada de decisões no que diz respeito a implementação dos requisitos.	1	2		1	1	2,8
O cliente é visto como membro da equipe.		2	1	1	1	3,2
Existem reuniões diárias entre a equipe no início ou		2			3	3,8

após o expediente para conhecer as atividades que foram realizadas e as que serão realizadas por cada um.					
Diariamente a equipe fica sabendo sobre os impedimentos que cada um está enfrentando.	1		2	2	4
Ao final de cada iteração, há uma reunião da equipe com o cliente para apresentar o que foi desenvolvido.	1	2	1	1	3,4
Ao final de cada iteração, a equipe se reúne para avaliar os pontos positivos e negativos ocorridos.	3		1	1	3
São realizadas reuniões complementares para solucionar impedimentos quando em outras reuniões não surge uma solução.			4	1	4,2
Existem entregas periódicas ao cliente em um prazo de 2 a 4 semanas.	3		1	1	3
Dentro de um projeto, o tempo de uma iteração não é alterado.	3	1		1	2
Os requisitos desenvolvidos nas iterações representam de fato o que o cliente deseja para aquele momento específico.		1	2	2	4,2

Para um melhor entendimento, foi decidido separar as quatro categorias divididas no questionário para discutir e analisar os dados obtidos.

- **Planejamento (questões 1 a 9)**

Através da Tabela 8 é possível observar que nas questões referentes à categoria Planejamento as respostas obtidas foram acima de 3, exceto a questão 9.

Para a primeira questão nota-se a média máxima com RM = 5, em que é possível perceber que todos os respondentes concordam que no início do projeto é elaborada uma lista contendo os requisitos que serão desenvolvidos.

A terceira e a sexta questões obtiveram valores de concordância com RM = 3,2. Porém, ainda se mantém com concordância acima de 3, significando que a equipe concorda que existe uma estimativa de esforço da equipe para cada requisito, e de que é planejada uma entrega parcial ao cliente.

Durante a fase de observação, foram anotados muitos acontecimentos relevantes no que diz respeito ao desenvolvimento dos requisitos. No parágrafo seguinte está um trecho registrado no diário de observação.

“Por volta das 14:00 horas, após todos terem retornado do almoço, começou uma reunião com os membros da equipe Scrum em que os mesmos decidiam como seria a próxima sprint. No quadro disponível na sala escreviam quais requisitos seriam implementados e a quantidade de esforço necessário para cada um dos requisitos...” (Diário de observação - 31/10/2013).

Sendo assim, a característica do *Scrum* relacionada à elaboração do *Product Backlog* foi concordada e compreendida por todos os membros da equipe, já que sabiam qual a

utilidade da lista de requisitos e também faziam o uso de maneira correta de como a mesma deve ser elaborada, priorizada e como deve ser realizada a estimativa de esforço para cada requisito.

Na questão 9 pode-se perceber que o valor do RM possui uma discordância, pois o Ranking Médio foi de 2,6. Dessa forma, a equipe não conhece sua produtividade de maneira quantitativa.

- **Artefatos (questões 10 a 17)**

Na categoria de Artefatos pode ser visto que o valor do RM mais próximo de 5 corresponde a questão 10 com o RM = 4,8. Percebendo-se assim, que os membros da equipe conhecem o quadro em que são visualizadas as atividades realizadas como *Scrum Board*.

De acordo com o diário de observação, esse dado pode ser confirmado:

“Ao andar pela sala de desenvolvimento percebi que os membros da equipe visualizavam um quadro on-line denominado Scrum Board em seus computadores, o qual continha as tarefas que deveriam ser realizadas. Este quadro eu também já tinha visto disponível em um quadro disposto na sala...”
(Diário de observação - 06/01/2014).

A questão que tem o segundo maior Ranking Médio é a questão 13 (RM = 4,2), referente à existência de um documento onde os requisitos são registrados contendo sua descrição. Podendo dizer assim, que a equipe concorda que é elaborado um documento que possui as histórias que serão desenvolvidas durante a *sprint* e entende para que serve esse documento.

A questão que obteve um menor valor no RM foi a questão 17, com o valor igual a 3, considerado como “indiferente” ou “ponto neutro”. O RM não chegou a ser considerado como discordante já que para isso é necessário serem estimados valores inferiores a 3.

Sobre a questão 17 pode-se dizer que a equipe nem concorda e nem discorda que é possível visualizar e acompanhar o andamento do projeto diariamente para verificar se estão dentro do estimado. Pela pontuação percebe-se que o grau de concordância não obteve uma numeração de concordância. Para melhor compreender o entendimento da equipe sobre esta característica do *Scrum* que envolve o *burndown chart* foram retirados no diário de observação alguns trechos:

“Assim que entrei na sala de desenvolvimento visualizei dois quadros dispostos do lado esquerdo da sala, o quadro maior era referente ao burndown chart...” (Diário de observação - 01/10/2013).

“Ao entrar na sala de desenvolvimento e olhar para o burndown chart percebi que o mesmo ainda não tinha sido atualizado desde o primeiro dia que iniciei a observação...” (Diário de observação - 02/11/2013).

“No final da tarde o Scrum Master estava atualizando o burndown chart enquanto o time dava sua opinião...” (Diário de observação - 05/12/2013).

Em relação a este aspecto, pode-se dizer que a equipe compreende para que serve o *burndown chart*, porém, o mesmo não é atualizado com a devida frequência que deveria, dificultando assim a visualização do andamento do projeto pela equipe.

- **Papéis e responsabilidades (questões 18 a 24)**

Para a categoria de papéis e responsabilidades, visualizando a Tabela 8, nota-se o maior RM na questão 22 com $RM = 4,4$. Existindo assim, uma concordância da equipe no que diz respeito ao *Product Owner*, em que a mesma concorda que o PO é o responsável pela lista de requisitos.

Na metodologia *Scrum* o *Scrum Master* não é visto como gerente, ele tem o papel de manter o processo da metodologia ativo e não de gerir a equipe. Para avaliar esta questão utilizou-se a forma negativa, ou seja, foi perguntado se o *Scrum Master* era visto como gerente de projetos (questão 20). O resultado foi um $RM = 2,8$, o que significa que houve discordância, mas que neste caso é visto como um resultado positivo em relação a aderência com o *Scrum*.

Se tratando ainda do *Scrum Master*, a questão 21 possui $RM = 4$, em que esse valor corresponde a um grau concordante em relação à implementação desse papel como uma das atividades desse membro na equipe, mas o *Scrum Master* também não deveria implementar. Logo essa questão também se torna negativa como a questão 20.

Porém, foi escrito no diário de observação um ponto negativo em relação ao *Scrum Master* e também ao *Product Owner*.

“No meio da tarde o Scrum Master entrou na sala em que fico e passou algumas horas ajudando um colega de outra equipe que estava com um pouco de dificuldade em achar uma melhor solução para um problema que havia sido relatado por um cliente...” (Diário de observação - 03/01/14).

“Hoje, assim como outros dias, o Product Owner foi interrompido diversas vezes para tirar dúvidas de vários colegas de outras equipes...” (Diário de observação - 24/03/14).

Baseado no que foi descrito nos dois parágrafos anteriores, é possível perceber que o trabalho tanto do *Scrum Master* quanto do *Product Owner* era interrompido para tirar dúvidas e realizar outras atividades junto a outras equipes. Esse era um caso que se encontrava em bastante discussão na empresa, pois tinham dias que esse fato ocorria por diversas vezes. Logo, este é um aspecto considerado como um dos impedimentos existentes para o uso da metodologia, pois a responsabilidade de cada um desses papéis abrangia mais do que é

sugerido pelo *Scrum*, atrapalhando assim, na realização das tarefas que deveriam ser desempenhadas pelos mesmos na equipe *Scrum*.

Na questão 24 questionou-se sobre a participação do cliente junto à equipe, para esta pergunta o RM foi igual a 3,2. No *Scrum* a participação do cliente é bastante importante, pois o mesmo deve estar disponível para sanar as dúvidas referentes aos requisitos, evitando atrasos e também uma construção errada do que deverá ser feito.

A presença do cliente junto à equipe foi anotada no diário de observação, como mostra no parágrafo a seguir.

“Hoje ao chegar na empresa por volta das 12:00 horas todos estavam comentando que um dos clientes iria passar lá para tirar algumas dúvidas referentes a alguns requisitos e dizer novos requisitos que deverão ser implementados...Aproximadamente 02:00 horas o cliente chegou e se juntou ao Product Owner para tirar as dúvidas e ficou na empresa até o fim do expediente...” (Diário de observação - 06/11/2013).

“Em mais um dia trabalho e semana de treinamento assim que cheguei na empresa fiquei sabendo que os dois analistas foram visitar o cliente principal de João Pessoa e só retornariam no final da tarde. A visita tinha o objetivo de verificar junto ao cliente os principais problemas que estavam ocorrendo no sistema e o que deveria ser alterado...” (Diário de observação - 17/01/2014).

Com base nesses fatos, o cliente tem sua participação no período de desenvolvimento e nas melhorias que deverão ser feitas para a satisfação do mesmo.

- ***Sprint (questões 25 a 32)***

Nesta categoria, em relação às reuniões diárias que acontecem no *Scrum*, é possível perceber pelas respostas da questão 25 um RM = 3,8, o qual se caracteriza como concordante. Logo, a equipe em geral, concorda que são realizadas reuniões diárias, em que as mesmas têm o objetivo de disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho a ser realizado no dia que se inicia.

A respeito das reuniões que ocorrem no final de cada iteração, a de Revisão da *Sprint* e a de Retrospectiva, os valores do Ranking Médio foram, respectivamente, 3,4 (questão 27) e 3 (questão 28).

Por meio desses dados, é possível entender que para a questão 28 não há concordância e nem discordância. Neste caso, é possível notar que a equipe tem dúvidas em relação a esse aspecto, e com isso, pode-se dizer que as possíveis soluções para melhorias que deveriam ocorrer nessa reunião não são assistidas.

Já para a questão 27 a equipe concorda que ao final de cada iteração, há uma reunião da equipe com o cliente para apresentar o que foi desenvolvido, ver se o que foi feito atende o que foi pedido, se é realmente o caminho que deve ser seguido e se algo deve ser mudado.

Na questão que diz respeito à existência de entregas periódicas ao cliente em um prazo de 2 a 4 semanas (questão 30) o RM foi igual a 3, significando que não há concordância e nem discordância sobre esse aspecto do *Scrum*.

De acordo com um trecho do diário de observação foi percebido um pouco de insatisfação do cliente em relação às entregas.

“No final do expediente encontrei um dos analistas do time Scrum e foi me passado que o cliente não estava muito satisfeito, pois haviam alterações que deveriam ser entregues que o mesmo pediu fazia bastante tempo e nada foi feito até o momento e isso não aconteceu apenas essa vez...” (Diário de observação - 17/01/2014).

Para a questão 30 pode-se dizer que o que foi descrito no diário de observação foi também percebido pela equipe através de suas respostas, pois algumas entregas não estavam sendo feitas dentro do prazo de no máximo 4 semanas e este fator estava gerando uma insatisfação em alguns clientes.

Na questão 31 é possível notar uma discordância da equipe com um RM = 2. Neste caso, a característica do *Scrum* que envolve a alteração do tempo do projeto não foi satisfeita, pois na metodologia o tempo não deve ser alterado dentro do projeto para não atrapalhar no estabelecimento da velocidade do time. Velocidade do time é uma métrica que define a produtividade de um time *Scrum*. Geralmente ela é calculada utilizando histórias de usuário por *Sprint*. Esta métrica ajuda a planejar com mais eficiência as *sprints* futuras.

Para compreender melhor o motivo dos fatos ocorridos nas questões 30 e 31, foram retirados do diário de observação os seguintes trechos:

“Hoje o dono se encontrava na empresa, um fato um pouco raro já que o mesmo está sempre viajando para visitar os clientes, e ele se reuniu com o Product Owner pra verificar a lista de requisitos durante a sprint e pediu que fossem colocados novos requisitos e que outros fossem retirados...” (Diário de observação - 02/12/2013).

“Por volta das 15:00 horas o dono da empresa enviou um email para a equipe Scrum com uma lista do que deveria ser desenvolvido e que teria que ser desenvolvido o mais rápido possível. Teria que dar prioridade à essa lista e deixar o que se estava sendo feito de lado para que isso fosse atendido...” (Diário de observação - 17/01/2014).

Com esses dados pode ser visto que o dono da empresa tinha total liberdade para alterar o *Product Backlog* mesmo durante o desenvolvimento dos requisitos. Logo, o prazo

para os requisitos que tinham sido priorizados na Reunião de Planejamento era estendido porque outras priorizações durante a *sprint* eram feitas. Outro fator perceptível é que o dono também assumia o papel de *Product Owner*, sendo este também outro impedimento para o uso da metodologia da forma como a mesma sugere.

5.1 Síntese e discussão dos resultados

Em uma visão mais ampla, na Figura 9 é possível visualizar uma comparação entre os valores que se consideram esperados para cada categoria, ou seja, valor igual a 5, e os valores obtidos através do questionário (dispostos na cor roxa). Chegou-se a esses valores somando os resultados dos RMs das perguntas de cada categoria e dividindo pela quantidade de questões.

Para as questões 20 e 21 que foram perguntadas na sua forma negativa, foi utilizado seu complemento, ou seja, como o RM obtido para questão 21 foi igual a 2,6, ele foi alterado para 3,4 ($5,0 - 2,6$) e a questão 21 para 2. Logo, as frequências das respostas ficaram: Concordo totalmente = 2; concordo parcialmente = 1; não concordo nem discordo = 0; discordo parcialmente = 1; discordo totalmente = 1

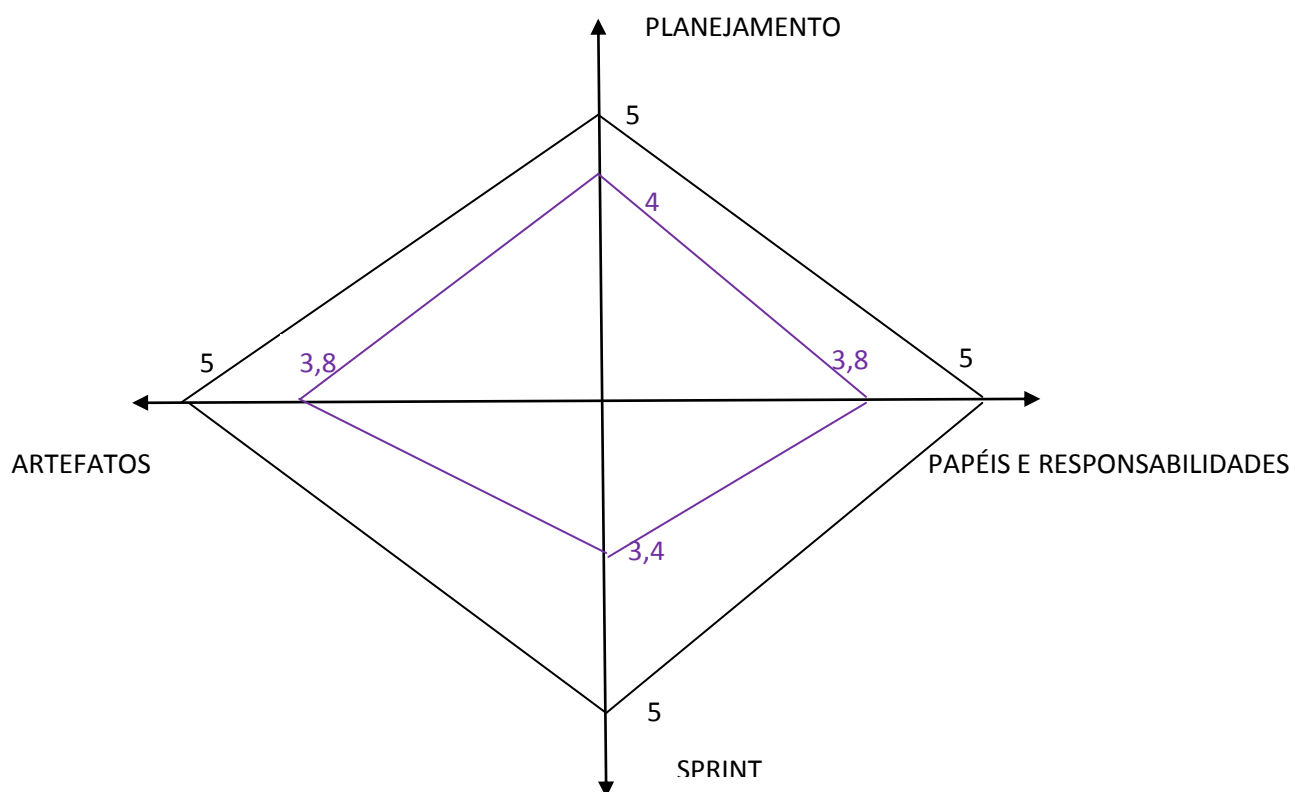


Figura 9. Gráfico comparativo entre os resultados esperados e os resultados obtidos pelo questionário.

Durante a análise, foi possível identificar que em relação à categoria referente ao Planejamento a equipe não teve muitas dificuldades em entender os aspectos referentes ao *Scrum* no que diz respeito ao *Product Backlog*, a estimativa dos requisitos, o *Planning Poker* e a Reunião de Planejamento. Porém, os membros da equipe precisam ter um maior conhecimento de sua produtividade.

Na categoria Artefatos notou-se que a equipe conhece os artefatos utilizados no *Scrum* e quais são suas utilidades. Entretanto, devem aprimorar mais alguns pontos, como a atualização dos quadros, pois assim a equipe terá uma melhor visualização do que está se passando ao longo da *sprint*.

Em relação aos Papéis e Responsabilidades existentes na metodologia, a equipe da empresa analisada conhece quais são os papéis do *Scrum*, mas as responsabilidades desses papéis não são bem desempenhadas, já que o *Scrum Master* e o *Product Owner*, não executam apenas o que é sugerido. Sendo este ponto considerado como um impedimento no uso do processo.

Na *Sprint*, por meio dos dados obtidos, pode-se dizer que a equipe *Scrum* utiliza as práticas da metodologia, como a Reunião diária e Reuniões de Revisão da *Sprint*, porém, a equipe discorda que é realizada a Reunião de Retrospectiva. Nesta categoria também foi percebido que o dono da empresa também assumia o papel de PO e que isso influenciava nos prazos de entregas que deveriam ocorrer.

6 CONCLUSÃO

O presente trabalho teve como objetivo analisar a utilização do *Scrum* com relação ao planejamento, papéis, artefatos e *sprint* com a intenção de verificar a percepção dos desenvolvedores sobre a aderência da metodologia na empresa. O mesmo buscou compreender também possíveis impedimentos no entendimento e execução da equipe no que diz respeito às características sugeridas pelo *Scrum*.

Percebe-se com esta pesquisa que para todas as categorias (Planejamento, Artefatos, Papéis e responsabilidades e *Sprint*) apesar de várias questões já serem bem compreendidas pela equipe, alguns pontos ainda precisam ser melhor entendidos pela mesma e que os impedimentos devem ser removidos para que o *Scrum* seja adotado da forma como de fato é sugerido, para que ocorram melhorias no prazo de entrega do produto final ao cliente e na satisfação do mesmo, no conhecimento do grupo sobre sua produtividade e o que está sendo realizado durante cada iteração, e no desenvolvimento das atividades de cada papel.

Por fim, notamos que os objetivos da pesquisa foram atendidos. Ela proporcionou a vivência de um caso real que utiliza uma metodologia ágil, permitindo a aprendizagem na prática sobre a metodologia *Scrum*.

6.1 Trabalhos futuros

O trabalho apresentado pode ser melhorado fazendo uma análise da eficiência do uso do *Scrum* na empresa e também através da aplicação do estudo em mais times/empresas. Além disso, algumas limitações deste trabalho podem ser melhor analisadas para permitir uma evolução da pesquisa, principalmente em relação ao questionário, que depois de aplicado, nos permitiu a identificação de melhorias, por exemplo, em alguns níveis de detalhes sobre elementos do *Scrum* como *timebox*, velocidade do time, reuniões diárias, entre outros.

7 REFERÊNCIAS

AZEVEDO, C. *Metodologias ágeis e gestão de pessoas*. 2012.

BAKER, M. J. (Org.). *Administração de marketing*. Rio de Janeiro: Elsevier, 2005.

BASIL, V.; CALCIERA, G. *The Goal Question Metric Approach*, 2002.

BECK, Kent. *Extreme Programming Explained*., New York: 2000, Addison Wesley.

EISENHARDT, K.M. *Building theories from case study research*. *Academy of Management Review*. New York, New York, v. 14 n. 4, 1989.

FUGGETTA, A. *Software process: a roadmap*. *Conference on The Future of Software Engineering*. Limerick, Irlanda. Junho, 2000. 25-34 p.

GIL, Antonio Carlos. *Como elaborar projetos de pesquisa*. 4. ed. São Paulo: Atlas, 2008.

GOLDENBERG, M. (1999) *A arte de pesquisar: como fazer pesquisa qualitativa em Ciências Sociais*. Rio de Janeiro: Record.

IRAOSSI, G. (2006). *The Power of Survey Design: A User's Guide for Managing Surveys, Interpreting Results, and Influencing Respondents*. Washington, D.C.: The World Bank.

JACOBSON, Robert E. *Information design*. Cambridge: MIT Press, 1999.

MIZERKOWSKI, D. *Estimativa utilizando Planning Poker*. Disponível em <http://tipratica.wordpress.com/2012/01/04/planning-poker/> acessado em 02 de Março de 2014.

MULLER, Gert. *Métodos Tradicionais versus ágeis: Um estudo comparativo através do TRAININGCAD*. Caruaru (PE), 2009.

NASCIMENTO, P. *O que é o Sprint Backlog*. Blog Scrum Half, 2012.

SCHWABER, K., BEEDLE, M. *Agile Software Development with Scrum*. Prentice Hall, 2002.

SCHWABER, K. *Agile Project Management with Scrum*. Microsoft Press, 2004.

SINGLENTON, Jr. Royce et alli. *Aproches to social research*. New York: Oxford University Press, 1970.

SOARES, Michel dos S. disponível em <http://www.dcc.ufla.br/infocomp/artigos/v3.2/art02.pdf> acessado em 20 de Janeiro de 2014.

SOMMERVILLE, I.,(2000) “*Software Engineering*”, 6th edition, Addison-Wesley.

SOMMERVILLE, Ian. *Engenharia de Software* - 8a edição, 2007.

SUTHERLAND, J.; SCHWABER, K. *Guia do Scrum*, 2011.

OLIVEIRA, L. H.. *Exemplo de cálculo de Ranking Médio para Likert*. Notas de Aula. Metodologia Científica e Técnicas de Pesquisa em Administração. Mestrado em Adm. e Desenvolvimento Organizacional. PPGA CNEC/FACECA: Varginha, 2005.

PACHECO, Diego. *FDD: um método ágil e eficiente*. Disponível em <http://imasters.com.br/artigo/13370/agile/fdd-um-metodo-agil-e-eficiente/> acessado em 03 de Fevereiro de 2014.

POWER, N. e MOYNIHAN,T. ;“*A Theory of Requirements Documentation Situated in Practice*” ; ACM ; San Francisco, California, USA. 2003.

PRESSMAN, Roger. *Software Engineering – A Practitioner’s Approach*, 6ª Edição, McGraw-Hill Professi, 2004.

PRESMMAN, Roger S. *Engenharia de Software*. 6. ed São Paulo: McGraw-Hill, 2006. 720 p.

WOHLIN, C. *Experimentation in Software EGINEERING: An introduction*. Kluwer Academic Publishers, USA, 2000.

YIN, R. K. *Estudo de caso: planejamento e métodos*. 2 ed. Porto Alegre: Bookman, 2001.

YIN, R. K. *Case study research, design and methods (applied social research methods)*. Thousand Oaks, 2009. California: Sage Publications.

YIN, R. K. *Estudo de caso: planejamento e métodos*. 4 ed. Porto Alegre: Bookman, 2010.

Zelkowitz, M. V. e Wallace, D. (1998) “*Experimental models for validating Technology*”. IEEE Computer, 31(5): 23-31.

APÊNDICE A - PROTOCOLO DE ESTUDO DE CASO

PROTOCOLO

“A utilização da metodologia *Scrum* sob a percepção da equipe de desenvolvimento em uma empresa privada de software: Um estudo de caso”

Jéssyca Ferreira de Oliveira

José Jorge L. Dias Jr.

Universidade Federal da Paraíba – Campus IV

Centro de Ciências Aplicadas e Educação

Rio Tinto, PB - Brasil

2014

SOBRE ESTE DOCUMENTO

Este documento descreve o protocolo de estudo de caso sobre o uso e adesão do método ágil *Scrum* em uma empresa privada de software. O mesmo apresenta detalhes sobre o projeto e instrumentos de pesquisa para coleta de dados.

Segundo Yin (2005), o protocolo de desenvolvimento do estudo de caso é um documento que serve como um roteiro facilitador para a etapa de coleta de dados. Contempla o instrumento de coleta de dados e toda a conduta a ser seguida pelo pesquisador durante a verificação. Possuindo assim, a função de aumentar a confiabilidade da pesquisa ao servir como guia ao investigador ao longo das atividades do estudo.

PERFIL DOS PESQUISADORES

Jéssyca Ferreira de Oliveira se encontra no último período do curso de Sistemas de Informação na UFPB (Universidade Federal da Paraíba) – Campus IV, está sendo orientada pelo professor José Jorge Lima Dias Júnior, do Departamento de Ciências Exatas, coordenador da equipe de desenvolvimento da UFPB Virtual, mestrado em Ciência da Computação pela Universidade de Pernambuco e doutorando em Ciência da Computação também pela UFPE. O professor tem experiência na indústria e em pesquisa qualitativa na área de Engenharia de Software.

EQUIPE DE PESQUISA

Tabela 1. Equipe de Pesquisa

Equipe	Cargo	Unidade	Papel
Jéssyca Ferreira	Estagiária	Sistemas de Informação	Pesquisadora
José Jorge Dias	Professor	DCE, UFPB Virtual	Orientador do projeto

DESIGN

- **Objetivo Geral**

Analisar a utilização do *Scrum* **com relação ao** planejamento, papéis, artefatos e sprint **com o objetivo** de verificar a percepção da aderência ao *Scrum* **sob o ponto de vista** dos desenvolvedores **no contexto de** uma empresa privada de software.

QUESTÕES DE PESQUISA

QP01. Que aspectos do *Scrum* estão sendo adotados na empresa analisada?

QP02. Qual a percepção da equipe em relação ao planejamento do projeto?

QP03. Qual a percepção da equipe em relação aos artefatos utilizados?

QP04. Qual a percepção da equipe em relação aos papéis?

QP05. Qual a percepção da equipe em relação a *Sprint*?

FERRAMENTAS DE COLETA E ANÁLISE DE DADOS

Tabela 2. Ferramentas utilizadas no processo de coleta e análise dos dados.

Atividade	Ferramenta
Coleta de dados	Microsoft Word
Análise de dados	Microsoft Word

PARTICIPANTES DA PESQUISA – INFORMAÇÕES DEMOGRÁFICAS

Tabela 3. Código dos participantes colaboradores da equipe de software

ID	Papel
P01	Analista de Sistemas – PO
P02	Desenvolvedor - Scrum Master
P03	Tester – Equipe
P04	Desenvolvedor – Equipe
P05	Desenvolvedor – Equipe

MÉTODO

Com base nas definições apresentadas em Runeson (2008), este estudo pode ser classificado como mostrado na tabela abaixo:

Tabela 4. Metodologia

Natureza da pesquisa	
Método	Estudo de caso
Propósito	Descritiva <i>“Descritiva: Descrever o fenômeno da auto-gestão no desenvolvimento de software, estabelecendo relações entre as variáveis” Runeson, 2008, pg. 135). A finalidade é observar, registrar e analisar os fenômenos ou sistemas técnicos, sem, contudo, entrar no mérito dos conteúdos.</i>
Perspectiva	Interpretativa <i>“Tentativas de compreender os fenômenos através da interpretação de seu contexto dos participantes” (Runeson, 2008, p.135).</i>
Natureza do dado	Qualitativo e quantitativo <i>“Dados qualitativos são representados como palavras e imagens, e não números(...). Os resultados são mais ricos e informativos.(...)” (Seaman, 1999, pg 557).</i>
Design	Flexível <i>“Em um processo de concepção flexível principais parâmetros do estudo pode ser alterado durante o decorrer do estudo” (Runeson, 2008, pg 136).</i>
Triangulação	Observação e questionário
Técnica de coleta	
Método de procedimento	Estudo de caso único e significativo suportado por etnografia <i>“Estudos de casos investigam um fenômeno contemporâneo em seu contexto natural”(Yin, 2003) e da etnografia é uma forma de investigação centrada sobre a sociologia do significado através de observação de campo. O objetivo é estudar uma comunidade de pessoas a entender como os membros da comunidade que fazer o sentido de suas interações sociais (Robinson, et al, 2007).</i>

REFERÊNCIAS

YIN, R. K. **Estudo de caso: planejamento e métodos**. 3. ed. Porto Alegre: Bookman, 2005.

RUNESON, P.; HOST, M. **Guideline for Conducing and Reporting Case Study Research in Software Engineering**, *Empirical Software Engineering*, v. 14, n. 2, p. 131- 164, 2008. DOI 10.1007/s10664-008-9102-8.

APÊNDICE B - ROTEIRO DE OBSERVAÇÃO

ROTEIRO DE OBSERVAÇÃO

Avaliador (a): Jéssyca Ferreira de Oliveira

Período: 01/10/2013 a 01/04/2014

Observar e anotar como ocorrem as seguintes questões:

1. Práticas do *Scrum* utilizadas pelos profissionais da empresa:
 - a. Discussões diárias em que cada membro da equipe responde às seguintes perguntas:
 - O que fiz desde ontem?
 - O que estou planejando fazer até amanhã?
 - Existe algo me impedindo de atingir a minha meta?
 - b. Reunião de planejamento da *Sprint*;
 - c. Reunião de revisão da *Sprint*;
 - d. Transparência no planejamento e desenvolvimento;
 - e. Planos frequentes de mitigação de riscos desenvolvidos pela equipe;
 - f. Reuniões frequentes com os stakeholders para monitorar o progresso;
 - g. Papéis bem definidos (*Product Owner*, Equipe de desenvolvimento e *Scrum Master*)
 - h. Entregas frequentes e intermediárias de funcionalidades;
 - i. Participação dos clientes como parte da equipe de desenvolvimento.
2. Como ocorre o processo de comunicação?
 - a. Realmente as pessoas fazem reuniões diárias?
 - b. O quadro *burndown* é usado?
 - c. As pessoas conversam apenas nas reuniões ou durante o trabalho também ocorrem debates sobre os artefatos?
3. Possíveis impedimentos na implantação da metodologia do *Scrum* na empresa.
 - a. Como as pessoas tratam os conflitos? Quais as medidas adotadas? Tudo é dito apenas *face-to-face* ou existem processo de documentação?

APÊNDICE C - QUESTIONÁRIO

Planejamento	
Q1. No início do processo de desenvolvimento é elaborada uma lista de requisitos que serão desenvolvidos.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q2. Para as funcionalidades requeridas pelo cliente é feita uma priorização dos requisitos a serem desenvolvidos.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q3. Para cada requisito é estimada uma quantidade de esforço da equipe.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q4. Toda a equipe participa da estimativa dos requisitos a serem desenvolvidos.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q5. A estimativa é feita através da contagem de pontos e para isso utiliza-se um baralho de cartas.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q6. O projeto é planejado para ter entregas parciais ao cliente.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q7. O projeto segue o modelo iterativo e incremental onde as iterações são chamadas de Sprints.	
a)	Concordo totalmente.

b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q8. Os membros da equipe se reúnem no início de cada iteração para planejar o prazo de entrega e as tarefas a serem realizadas.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q9. A equipe conhece sua produtividade de maneira quantitativa e utiliza esse valor para planejar o quanto entregar em cada <i>Sprint</i>.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Artefatos	
Q10. A ferramenta para visualização do trabalho em andamento está disponível em um quadro na sala.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q11. A ferramenta para visualização do trabalho em andamento é atualizada todos os dias pelos membros da equipe.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q12. O quadro onde estão contempladas todas as tarefas que serão realizadas de acordo com a prioridade de cada item é chamado <i>Scrum Board</i>.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q13. Existe um gráfico em que a equipe visualize o andamento do projeto.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q14. O gráfico para visualização do andamento de todo o projeto está exposto em um quadro.	
a)	Concordo totalmente.
b)	Concordo parcialmente.

c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q15. Existe um documento onde os requisitos são registrados contendo sua descrição.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q16. O documento de requisitos é denominado <i>Product Backlog</i>.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q17. A equipe consegue visualizar e acompanhar o andamento do projeto diariamente para verificar se estão dentro do estimado.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Papéis e responsabilidades	
Q18. Existe uma pessoa na equipe responsável por resolver os impedimentos (problemas) que surgem.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q19. O membro da equipe responsável por remover os impedimentos é chamado de <i>Scrum Master</i>.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q20. O <i>Scrum Master</i> atua como gerente da equipe.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q21. O <i>Scrum Master</i> também participa da implementação do software.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q22. O membro da equipe responsável pela lista de requisitos é denominado <i>Product Owner</i>.	

a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q23. A equipe é capaz de deter todo o poder na tomada de decisões no que diz respeito a implementação dos requisitos.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q24. O cliente é visto como membro da equipe.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Sprint	
Q25. Existem reuniões diárias entre a equipe no início ou após o expediente para conhecer as atividades que foram realizadas e as que serão realizadas por cada um.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q26. Diariamente a equipe fica sabendo sobre os impedimentos que cada um está enfrentando.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q27. Ao final de cada iteração, há uma reunião da equipe com o cliente para apresentar o que foi desenvolvido.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q28. Ao final de cada iteração, a equipe se reúne para avaliar os pontos positivos e negativos ocorridos.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q29. São realizadas reuniões complementares para solucionar impedimentos quando em outras reuniões não são surge uma solução.	
a)	Concordo totalmente.
b)	Concordo parcialmente.

c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q30. Existem entregas periódicas ao cliente em um prazo de 2 a 4 semanas.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q31. Dentro de um projeto, o tempo de uma iteração não é alterado.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.
Q32. Os requisitos desenvolvidos nas iterações representam de fato o que o cliente deseja para aquele momento específico.	
a)	Concordo totalmente.
b)	Concordo parcialmente.
c)	Não concordo nem discordo.
d)	Discordo parcialmente.
e)	Discordo totalmente.