



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JAKELINE BANDEIRA DE OLIVEIRA

**ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS EM
EMPRESAS DE *SOFTWARE* NO MUNICÍPIO DE PATOS – PB**

**PATOS – PB
2019**

JAKELINE BANDEIRA DE OLIVEIRA

**ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS EM
EMPRESAS DE *SOFTWARE* NO MUNICÍPIO DE PATOS – PB**

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Computação.

Orientador: Prof. Esp. Sergio Morais Cavalcante Filho

**PATOS – PB
2019**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

O48a Oliveira, Jakeline Bandeira de.
Análise do processo de desenvolvimento de sistemas em empresas de software no município de Patos – PB [manuscrito] / Jakeline Bandeira de Oliveira. - 2019.
49 p. : il. colorido.
Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2019.
"Orientação : Prof. Esp. Sergio Morais Cavalcante Filho, Coordenação do Curso de Computação - CCEA."
1. Software. 2. Desenvolvimento de software. 3. Empresa de software. I. Título
21. ed. CDD 005.1

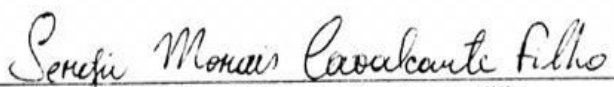
Jakeline Bandeira de Oliveira

**ANÁLISE DO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS EM EMPRESAS
DE SOFTWARE NO MUNICÍPIO DE PATOS – PB**

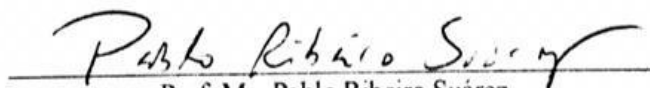
Trabalho de Conclusão de Curso apresentado ao
Curso de Bacharelado em Ciências da
Computação da Universidade Estadual da
Paraíba, em cumprimento à exigência para
obtenção do grau de Bacharel em Ciências da
Computação.

Aprovado em 13/06/2019

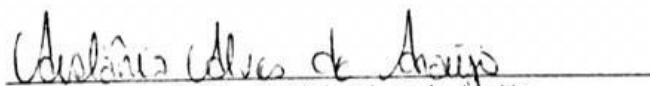
BANCA EXAMINADORA



Prof. Esp. Sergio Morais Cavalcante Filho
(Orientador)



Prof. Me. Pablo Ribeiro Suárez
(Examinador)



Prof. Me. Aislânia Alves de Araújo
(Examinador)

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida, aos meus pais por terem me apoiado e proporcionado chegar até aqui, o esforço e dedicação tiveram todos esses anos, aos meus irmãos. Não foi fácil chegar até aqui, foi uma batalha muito dura, mas cheia de lições e aprendizados. Agradeço a cada um que se fez presente direta ou indiretamente nesta caminhada.

Aos amigos que fiz durante esses anos, em especial a José Neto, obrigada por ter sido e continuar sendo meu “Google”, Michelle, Lucas e tantos outros. Thais que mesmo me odiando inicialmente, aprendi a gostar e se tornou uma grande amiga.

A todos os professores do Curso, agradeço por todos ensinamentos e lições as quais foram fundamentais na minha caminhada.

Agradeço imensamente ao meu orientador professor Sergio Cavalcante, que mesmo nunca tendo sido meu professor e sem me conhecer aceitou ser meu orientador. Obrigada! Suas contribuições foram essências para a realização desse trabalho.

RESUMO

Sabendo que atualmente, as organizações, empresas de grande e pequeno porte, independente do segmento, dependem cada vez mais da tecnologia para que possam gerir, planejar e executar seus negócios e atividades, a busca por *software* de qualidade para auxiliar nessas diversas atividades tem ganhado novas proporções, fazendo com que as empresas do seguimento de desenvolvimento de *software* possam oferecer sistemas de qualidade, confiáveis, seguros com eficácia e satisfação cada vez melhor. Diante disso este trabalho tem como objetivo verificar como ocorre a elaboração e caracterização do processo de desenvolvimento de *software* nas empresas desenvolvedoras que atuam na cidade de Patos no sertão paraibano. Utilizou-se de um *Survey*, como coleta de dados com um universo de cinco empresas para obter os resultados, onde se pôde identificar detalhadamente o processo de escolha de cada metodologia. Ao final desta pesquisa pôde-se evidenciar que 80% (quatro) empresas utilizam a metodologia Scrum, destas, apenas em uma empresa o desenvolvedor tem autonomia para escolher qual metodologia utilizará para o processo de desenvolvimento. Todas as empresas usam a entrevista como especificações dos requisitos além dos critérios que as empresas usam para suas escolhas de metodologia de desenvolvimento de *software*.

Palavras-Chave: *Software*; Desenvolvimento de *Software*; Empresas;

ABSTRACT

Knowing that currently, organizations, large and small companies, independent of the segment, are increasingly dependent on technology so that they can manage, plan and execute their businesses and activities, the search for quality software to assist in these various activities has gained new proportions, enabling software development tracking companies to deliver quality, reliable, secure systems with ever-increasing effectiveness and satisfaction. Therefore, this work has the objective of verifying how the elaboration and characterization of the software development process occurs in the companies that work in the city of Patos in the Sertão of Paraíba. A Survey was used as a data collection with a universe of five companies to obtain the results, where it was possible to identify in detail the process of choosing each methodology. At the end of this research it was possible to show that 80% (four) companies use the Scrum methodology, of these, only in one company the developer has the autonomy to choose which methodology will be used for the development process. All companies use the interview as requirements specifications beyond the criteria that companies use for their software development methodology choices.

Keywords: Software; Devolep Engineering; Companies;

LISTA DE ILUSTRAÇÕES

Figura 1 – Processo de engenharia de requisitos	21
Figura 2 – Fases genéricas da engenharia de software.....	27

LISTA DE QUADROS

Quadro 1 – Validação de <i>software</i> pelas empresas analisadas	41
---	----

LISTA DE GRÁFICOS

Gráfico 1 – Metodologias usadas pelas empresas de software	37
Gráfico 2 – Especificações de requisitos nas empresas	40
Gráfico 3 – Acompanhamento dos clientes no projeto	41

LISTA DE ABREVIATURAS E SIGLAS

APS	Análise e Projeto de Sistemas
DS	Desenvolvimento de Software
ER	Engenharia de Requisitos
ERP	Enterprise Resource Planning
ES	Engenharia de Software
PDS	Processo de Desenvolvimento de Software
SW	Software
TI	Tecnologia da Informação
XP	Extreme Programming

SUMÁRIO

1. INTRODUÇÃO.....	14
1.1 Cenário técnico-científico	14
1.2 Definição da problemática.....	15
1.3 Justificativa.....	17
1.4 Objetivos	18
1.4.1 Objetivo geral.....	18
1.4.2 Objetivos específicos	18
1.5 Estrutura do Trabalho.....	18
2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS	20
2.1 Engenharia de <i>Software</i>	20
2.2 Engenharia de Requisitos	21
2.3 Processos de Desenvolvimento de Sistemas	23
2.4 Trabalhos Semelhantes	29
2.4.1 Investigando a aplicação de metodologias ágeis em desenvolvimento de softwares por empresas localizadas no sertão paraibano	29
2.4.2 Análise da importância dos requisitos no desenvolvimento de software utilizando artefatos do scrum com o apoio da engenharia de requisitos	30
2.4.3 Avaliação e adoção de uma metodologia de gerenciamento de mudanças nos processos de manutenção de software.....	30
3. METODOLOGIA	32
3.1 Tipo de Pesquisa	32
3.2 Local da Pesquisa	33
3.3 População e Amostra	33
3.4 Critérios de Inclusão e Exclusão	33
3.5 Instrumento de Coleta de Dados.....	34
3.6 Procedimento de Coleta de Dados	34
4. RESULTADOS E DISCUSSÕES.....	36
4.1 Caracterização da amostra	36
4.2 Uso de uma metodologia no processo de desenvolvimento de <i>software</i>	36
4.3 Critérios de Utilização e método próprio	37
4.4 Especificações de requisitos	38
4.5 Validação do <i>software</i>	40
5. CONSIDERAÇÕES FINAIS.....	42
REFERÊNCIAS.....	43

APÊNDICE A – QUESTIONÁRIO PARA ANÁLISE DAS EMPRESAS DE DESENVOLVIMENTO DE SOFTWARE	46
ANEXO A – Termo de Consentimento Livre e Esclarecido.....	48
ANEXO B – Solicitação de consentimento para realização de pesquisa acadêmica	49

1. INTRODUÇÃO

Nesta seção é exposta uma breve explanação sobre o cenário dos avanços tecnológicos e a definição da problemática; o objetivo geral e específicos para o desenvolvimento do trabalho, bem como a justificativa.

1.1 Cenário técnico-científico

As tecnologias estão cada vez mais presentes no cotidiano da população mundial. Os setores de indústria, mercado, educação, entre outros, hoje, são informatizados, sendo compreendido a uma perspectiva de aplicar e utilizar os recursos computacionais de *hardware* e *software* para com os processos nos respectivos ambientes que estão inseridos.

O *hardware*, por sua vez, sozinho não é capaz de executar nenhuma tarefa, ele precisa de uma unidade lógica e aritmética capaz de traduzir e direcionar suas tarefas. Para atingir este objetivo, a unidade lógica e aritmética (*arithmetic logic unit*) – ULA tem vetores de entrada e saída de dados. Enquanto entrada de dados, há a categoria de entrada de controle que são usadas para selecionar a função específica a ser executada em um tempo determinado, cada *hardware* tem suas funções e cada uma de suas funções precisam ser controladas para que assim se faça uso do *software*.

Os avanços da área tecnológica vêm acontecendo também devido às otimizações nos novos modelos de desenvolvimento de *software* (DS) e a competição do mercado faz com que as empresas e organizações busquem sempre aprimorar seus procedimentos. O uso de um processo inadequado pode reduzir a qualidade e a eficácia do *software* a ser desenvolvido.

Os procedimentos adotados pelas empresas desenvolvedoras para adquirir, construir, gerenciar e otimizar a Tecnologia da Informação (TI) está mudando significativamente nos últimos anos, e conseqüentemente, continuará em constantes transformações para atender às futuras demandas.

A revista eletrônica ComputerWorld (2018) diz que o mercado de software e serviços de TI deve manter crescimento até 2021, segundo estudos. O Brasil está no décimo lugar entre os mercados de TI do mundo, representando 1,9% das vendas de hardware, software e serviços globais.

O setor de TI no Brasil está cada vez mais expansivo e centrado no DS e serviços da Tecnologia da Informação. Isso se dá pelo crescente número de *startups* nos últimos anos, que foram impulsionadas, principalmente, pela busca de inovação por grandes empresas nacionais das indústrias financeira, manufatureira e de serviços conforme aponta a revista EMIS - Insights, empresa multinacional que fornece informações estratégicas de empresas, setores e países dos mercados emergentes, e aborda como o mercado está e quais os caminhos que deve seguir até 2021 (EMIS INSIGHTS, 2018).

1.2 Definição da problemática

O processo de DS pode ser compreendido como um conjunto de etapas sistematizadas. Na literatura há diferentes ciclos de vida, ou seja, diversos processos para desenvolvimento de sistemas, além de métodos customizados a partir de etapas aplicadas pelas empresas desenvolvedoras. Entretanto, Sommerville (2011) aborda, no que se refere aos diferentes ciclos de vida de software, que em todos os modelos há quatro fases genéricas: 1) **especificação de software** que é a fase dos levantamentos de requisitos onde as metas são estipuladas por meio de perguntas aos usuários e também por entrevistas, visitas, entre outros instrumentos de coleta de dados, a partir dessas atividades que o desenvolvedor inicia a segunda etapa; 2) **projeto e implementação de software**, onde será feito tudo que está estipulado na definição, que é iniciada depois de entender as necessidade do cliente para com o software a ser implementado; 3) **validação de software**, esta etapa visa verificar se o sistema atende aos requisitos e sua finalidade; 4) **evolução do sistema** nesta atividade tem como proposta a atualização do software as novas demandas computacionais, como também às novas especificações do cliente (SOMMERVILLE, 2011).

As fases genéricas recebem esta definição, pois estão presentes nos diferentes modelos de ciclo de vida de SW¹. Alguns modelos de DS têm sido amplamente discutidos na literatura: o ciclo de vida clássico (ou cascata), a prototipação, o modelo

¹ De acordo com a NBR ISO/IEC 12207 (1998, p. 3), o ciclo de vida de SW é a “Estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema, desde a definição de seus requisitos até o término de seu uso.”

espiral e as técnicas de quarta geração (PRESSMAN, 2002). Também, pode-se dizer que outros modelos têm sido bastante usado: o modelo iterativo e incremental (MARTINS; RAMOS; LOPES, 2012).

Segundo Soares (2004), as metodologias são divididas em dois grandes grupos: tradicionais e ágeis. Na tradicional se planeja muito com antecedência todo o percurso do projeto, diferente da ágil que é feito de forma iterativa e incremental, descobrindo o percurso no caminho. Essas metodologias surgiram em um contexto de desenvolvimento de *software* muito diferente do atual, baseado apenas em um *mainframe* e terminais burros². Na época, o custo de fazer alterações era alto, uma vez que o acesso aos computadores era limitado e não existiam modernas ferramentas de apoio ao desenvolvimento do software. Por isso, o software era todo planejado e documentado antes de ser implementado.

Hoje, a documentação é a base da implementação do sistema e acompanha o processo de desenvolvimento. Para obtenção de um produto satisfatório, o sistema deve atender as funcionalidades e restrições previamente estabelecidas e com respostas a todas as necessidades do cliente final. O analista de software estuda com detalhes e trabalha todas as etapas de criação de um software junto com o cliente e/ou usuários.

Os *stakeholders*³ são recursos essenciais para conhecer necessidades do sistema, definições dos requisitos, avaliação de viabilidade, definir a metodologia do desenvolvimento e a realização de vários testes das funcionalidades. Para isto, são realizadas reuniões, entrevistas, visitas para descobrir, tornar explícito e obter o máximo de informações para o conhecimento do objeto em questão.

Para medir a qualidade externa e interna do sistema há seis atributos importantes que podem ser usados conforme a NBR ISO/IEC 9126-1 (2001) que são funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. E para medir a qualidade de uso existem quatro características segundo a NBR ISO/IEC 9126-4 (2001) que são eficácia, produtividade, segurança e satisfação.

² Um monitor de exibição que não possui recursos de processamento. Um terminal burro é simplesmente um dispositivo de saída que aceita dados da CPU, apenas mostra imagens na tela, enquanto o servidor processa e envia tudo pela rede, limitando muito o desempenho das estações (MORIMOTO, 2005).

³ *Stakeholders* é todo grupo ou indivíduo que influencia ou é influenciado pelo alcance dos objetivos da organização, são as partes interessadas (FREEMAN, 1984).

Segundo Bourque e Fairley (2014) qualidade de *software* é uma área do conhecimento da Engenharia de *Software* (ES) que pode se referir: as características desejadas de produtos de *software*; e a extensão em que um produto de *software* possui as características, anteriormente citadas, e aos processos, ferramentas e técnicas que são usadas para garantir essas características, seja uma grande empresa ou pequena, todas as fases de elaboração são indispensáveis para que o *software* seja útil.

No processo de desenvolvimento de *software* (PDS) é realizado cada uma das tarefas mencionadas anteriormente, onde o produto final está intrinsecamente atrelado a todas as etapas que o antecedem. De acordo com os aspectos elencados formula-se então a questão problema deste trabalho: Como ocorre a elaboração e caracterização do processo de desenvolvimento de *software* nas empresas desenvolvedoras que atuam na cidade de Patos no sertão paraibano?

1.3 Justificativa

Seguindo os pensamentos de Koscianski e Soares (2007) e Pressman e Maxim (2016) no qual versa sobre prerrogativas básicas de desenvolvedores que fazem a elaboração das etapas do processo de DS de forma equivocada, ou seja, que não se baseia em um modelo de desenvolvimento para auxílio organizacional. Nas condições empresariais que podem acarretar falhas em um projeto. Por exemplo, requisitos não compreendidos completamente antes do início do projeto; as que apresenta a baixa/falta de capacitação da equipe de desenvolvimento de sistemas, entre outros aspectos.

As observações e discussões sobre a importância das etapas de criação de *software* no que se refere a conhecer as necessidades do cliente, definir os requisitos, avaliar a viabilidade do projeto, documentar os procedimentos, testar as funcionalidades criadas, nas quais foram propiciadas pelos componentes curriculares de ES e Análise e Projetos de Sistemas do curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba.

Para tanto, surgiu a necessidade de um estudo nas empresas de desenvolvimento de *software* para mensurar os processos atitudinais de cada uma das etapas do desenvolvimento de sistemas, desde a idealização até a

implementação e manutenção. Dessa forma, este trabalho contribuirá como uma pesquisa atual que retrata o cenário dos processos de desenvolvimento de software.

Assim o presente estudo teve como intuito responder e analisar se as empresas desenvolvedoras da cidade de Patos, abrange de forma eficiente os critérios, atributos e modelos de ciclos de vida para implementação de *software*.

1.4 Objetivos

Os respectivos objetivos foram levantados e divididos em geral e específicos.

1.4.1 Objetivo geral

Identificar quais os modelos de ciclos de vida de software estão sendo aplicados, analisando os processos de cada etapa de elaboração e desenvolvimento de sistemas nas empresas desenvolvedoras no município de Patos – Paraíba.

1.4.2 Objetivos específicos

- Apresentar um arcabouço teórico-metodológico do processo de desenvolvimento de software;
- Conhecer como ocorre os processos de elaboração de software por empresas desenvolvedoras de sistemas;
- Aplicar um questionário com representantes das empresas desenvolvedoras do município de Patos, identificando as semelhanças e discrepâncias nos processos, bem como confrontar os dados obtidos com a literatura.
- Análisar os critérios utilizados pelas empresas para estimar o modelo de desenvolvimento a ser implantado na construção de sistemas;

1.5 Estrutura do Trabalho

Este trabalho apresenta cinco capítulos, organizado da seguinte forma: Capítulo 1, é uma explanação geral desta investigação em conformidade com a contextualização do problema, objetivos e justificativa; No capítulo 2 é apresentado o

embasamento científico sobre a proposta do trabalho e o estado da arte da literatura a partir de trabalhos relacionados; O capítulo 3 é definido a metodologia adota para a pesquisa; No capítulo 4 mostra os resultados e discussões da análise dos dados obtidas; O capítulo 5 expõe as considerações finais, suas contribuições e as sugestões para trabalhos futuros.

2. REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

Este capítulo está dividido e organizando em quatro seções: Engenharia de *Software*; Engenharia de Requisitos; Processos de Desenvolvimento de Sistemas e Trabalhos Relacionados.

2.1 Engenharia de *Software*

Entende-se por *softwares* um conjunto composto por instruções de computador, estruturas de dados e documentos (PRESSMAN, 2006). Engenharia é o conjunto de técnicas e métodos para aplicar o conhecimento técnico e científico na planificação, criação e manutenção de estruturas, máquinas e sistemas para benefício do ser humano (FERREIRA, 1999). Logo, Engenharia de *Software* é a criação e a utilização de sólidos princípios a fim de obter sistemas que sejam confiáveis e que trabalhem eficientemente em máquinas reais.

A ES tem por objetivo apoiar o desenvolvimento profissional do *software*, mais do que a própria programação individual conforme elucida Sommerville (2011). A ES inclui técnicas que apoiam a especificação, projeto e evolução de programas, que normalmente não são relevantes para o desenvolvimento de *software* pessoal, mas que se aplicam em *softwares* como por exemplo um sistema bancário que deve ser bastante seguro. E requer um conjunto específico de atributos muito mais relevantes do que em um *software* mais simples de uso pessoal.

ES é uma abordagem sistemática e disciplinada para o desenvolvimento de *software* (PRESSMAN, 2006). Neste contexto se insere alguns aspectos como: custos aceitáveis, gerenciamento do processo de desenvolvimento, garantia do trabalho em equipe e desenvolvimento de *softwares* com qualidade.

Vale ressaltar, que este cenário possui diversas ferramentas e técnicas apropriadas que são escolhidas de acordo com o produto a ser desenvolvido, uma vez que cada sistema contém suas especificidades, portanto não se adequando a qualquer um dos modelos de processo de desenvolvimento.

Não existe técnicas e métodos universais adequados a todos os sistemas e todas as empresas. Um conjunto diverso de métodos e ferramentas de engenharia de *software* tem evoluído nos últimos 50 anos (Sommerville, 2011), ou seja, os

procedimentos para criação de sistemas têm se aperfeiçoado para atender as demandas.

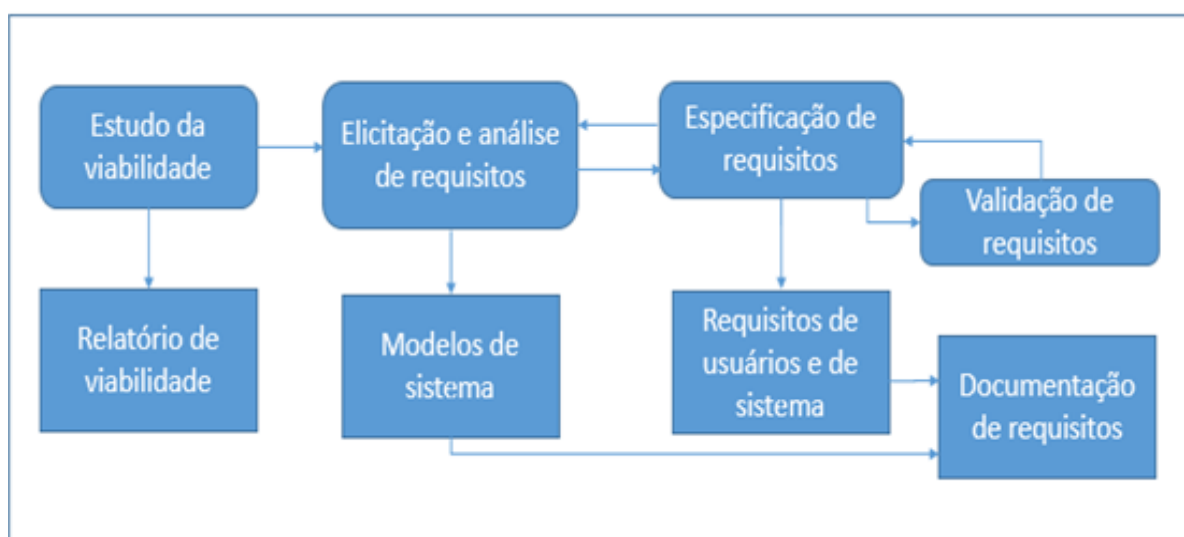
Diante disso, antes de iniciar a criação do sistema e definição das técnicas e métodos a serem adotadas para nortear o processo deve-se identificar o tipo de aplicação está a ser desenvolvida, sendo este fator mais significativo. No que se refere a aplicações existe um vasto campo de opções e as possibilidades de inovações aos que já são presentes no mercado e na literatura, tornando proeminente o processo de identificação.

2.2 Engenharia de Requisitos

A especificação de *software* ou Engenharia de Requisitos (ER) é o processo de compreensão e definição dos serviços pretendidos pelo sistema e identificação de restrições relativas à operação e ao desenvolvimento do sistema. Nessa etapa, o projeto se torna particularmente crítico, pois a ocorrência ou produção de erros, conseqüentemente gerarão problemas no projeto e na implementação do sistema (SOMMERVILLE, 2011).

Para melhor compreender e executar essa fase Sommerville apresenta um fluxograma com as etapas a serem seguidas. As especificações contidas na figura abaixo caracterizam o processo de ER que tem como objetivo produzir um documento de requisitos acordados que especifica um sistema que satisfaz os requisitos dos *stakeholders*.

Figura 1 - Processo de engenharia de requisitos



Fonte: Sommerville (2007, p.50)

A Figura 1 mostra as quatro principais atividades de ER e quatro atividades que resulta cada uma delas. Tem-se como finalidade a documentação de requisitos que é apresentada em dois níveis de detalhes. Os usuários finais e clientes precisam de requisitos em alto nível e os desenvolvedores precisam de requisitos mais detalhados do sistema.

Os requisitos de software são frequentemente classificados como requisitos funcionais e requisitos não funcionais:

Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer.

Requisitos não funcionais. São restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo (SOMMERVILLE, 2011, p. 59).

Requisitos funcionais tem a finalidade de descrever o que o sistema deve fazer, são compostos e formados de acordo com o tipo de *software* desenvolvido, são adotados pelos possíveis usuários como também pela organização ou empresa ao escrever os requisitos. São sempre escritos de forma abstrata para que possa ser entendido pelos usuários do sistema. Os requisitos funcionais podem variar de requisitos gerais, que é uma abrangência do que o sistema deve fazer, até requisitos muito específicos, que reflete os sistemas e a forma de trabalho.

Os requisitos não funcionais diferem dos funcionais pois já é mais da parte técnica, eles não estão ligados com os serviços específicos oferecidos no sistema. São direcionados a confiabilidade, tempo de resposta e ocupação de área. Surgem por meio das necessidades dos usuários, necessidade de interoperabilidade com outros sistemas de software ou hardware, ou com fatores externos.

Requisitos não funcionais são subdivididos em três grandes grupos para melhor detalhar a finalidade de cada um deles.

1. Requisitos de produto. Esses requisitos especificam ou restringem o comportamento do software. Exemplos incluem os requisitos de desempenho quanto à rapidez com que o sistema deve executar e quanta memória ele requer, os requisitos de confiabilidade que estabelecem a taxa aceitável de falhas, os requisitos de proteção e os requisitos de usabilidade.

2. Requisitos organizacionais. Esses são os requisitos gerais de sistemas derivados das políticas e procedimentos da organização do cliente e do desenvolvedor. Exemplos incluem os requisitos do processo operacional, que definem como o sistema será usado, os requisitos do processo de desenvolvimento que especificam a linguagem de programação, o ambiente de desenvolvimento ou normas de processo a serem usadas, bem como os requisitos ambientais que especificam o ambiente operacional do sistema.

3. Requisitos externos. Esse tipo abrange todos os requisitos que derivam de fatores externos ao sistema e seu processo de desenvolvimento. Podem incluir requisitos reguladores, que definem o que deve ser feito para que o sistema seja aprovado para uso, por um regulador, tal como um banco central; requisitos legais, que devem ser seguidos para garantir que o sistema opere dentro da lei; e requisitos éticos, que assegurem que o sistema será aceitável para seus usuários e o público em geral (SOMMERVILLE, 2011, p. 61).

Dentro de cada requisito ainda existem outros que detalham melhor os requisitos de produto, requisitos organizacionais e os requisitos externos. O requisito de produto é um requisito de disponibilidade que define quando o sistema deve estar disponível e o tempo diário permitido de seu não funcionamento. O requisito organizacional especifica como os usuários se autenticam para o sistema. O requisito externo deriva da necessidade de o sistema estar em conformidade com a legislação de privacidade.

É fundamental identificar os requisitos de um sistema como forma de planejamento de treinamento, utilizando-se da análise ou engenharia de requisitos para a especificação dos dados do ambiente proposto. O planejamento é um processo que consiste em preparar um conjunto de decisões, tendo em vista o agir, posteriormente para atingir determinados objetivos (KEMCZINSKI; KERN; CASTRO, 2001).

2.3 Processos de Desenvolvimento de Sistemas

Um processo de *software* (ou metodologia de desenvolvimento de *software*) é um conjunto de atividades e resultados associados que auxiliam na construção de *software*. Dentre as várias atividades associadas, o resultado do processo é um produto que reflete a forma como o processo foi conduzido (SOARES, 2004). O PDS se apresenta como um plano que auxilia nos resultados, exaltando a qualidade do SW, cumprindo o prazo estipulado e fazendo-se assim que um PDS seja de uso essencial na criação de um *software*, de tal modo que proporcione segurança,

liderança e organização para uma prática que pode, sem controle, torna-se altamente caótica.

Uma metodologia é a maneira simplificada que existe para se trabalhar de maneira mais eficiente e objetiva todo o processo de desenvolvimento, de modo que mescle a especificação do trabalho, estratégias e alternativas usadas pelos membros da equipe. Outra maneira usada trata-se de uma coleção de técnicas vistas para o desenvolvimento de SW, referindo-se que tais técnicas normalmente estão vinculadas a etapas ou estágios, de maneira a serem divididas em partes do processo, e assim, se possível mantê-lo em ordem bem como melhorar o gerenciamento.

Os processos de desenvolvimento de SW surgiram numa época de crise de software, que se fez necessário compor o trabalho de forma eficiente para solucionar problemas cada vez mais complexos. É primordial organizar as tarefas de maneira sólida para assim ser possível sanar problemas complexos. Diante disso houve a necessidade de tornar o processo de desenvolvimento estruturado e unificado, atendendo assim a todas as necessidades de forma eficaz (MAINART; SANTOS, 2010).

Os modelos de desenvolvimento de software são classificados como modelo tradicional e ágil. O tradicional é caracterizado pela rigidez, resultado da definição de passos sequenciais bem definidos. Diferencia-se pela formalidade, controle e rigor, onde todo o sucesso alcançado é o resultado do que foi planejado (PRESSMAN, 2006). Já para Mainart e Santos (2010), seu foco principal é conhecer todos os requisitos antes do projeto iniciar, possibilitando um melhor planejamento, facilitando a gerência e mantendo o processo rigoroso. Os autores consideram que as metodologias consideradas tradicionais, também chamadas de “pesadas”, pois tem como característica marcante serem divididas em etapas e/ou fases. Essas fases são muito bem definidas e englobam atividades como análise, modelagem, desenvolvimento e testes.

Um exemplo de processo que marca essa proposta e é muito utilizado é o modelo em Cascata ou Modelo Clássico (SANTOS, 2004), em que a priorização é a sequência das etapas, de forma que uma só é iniciada após o término da anterior e que a cada etapa finalizada, uma documentação deve ser apresentada e aprovada para que assim possa da sequência ao projeto.

Segundo Sommerville (2011) existem muitos processos de *software* diferentes, mas todos devem incluir quatro fases básicas para a ES:

Especificação de software. As funcionalidades de um software e as restrições para seu funcionamento devem ser definidas inicialmente.

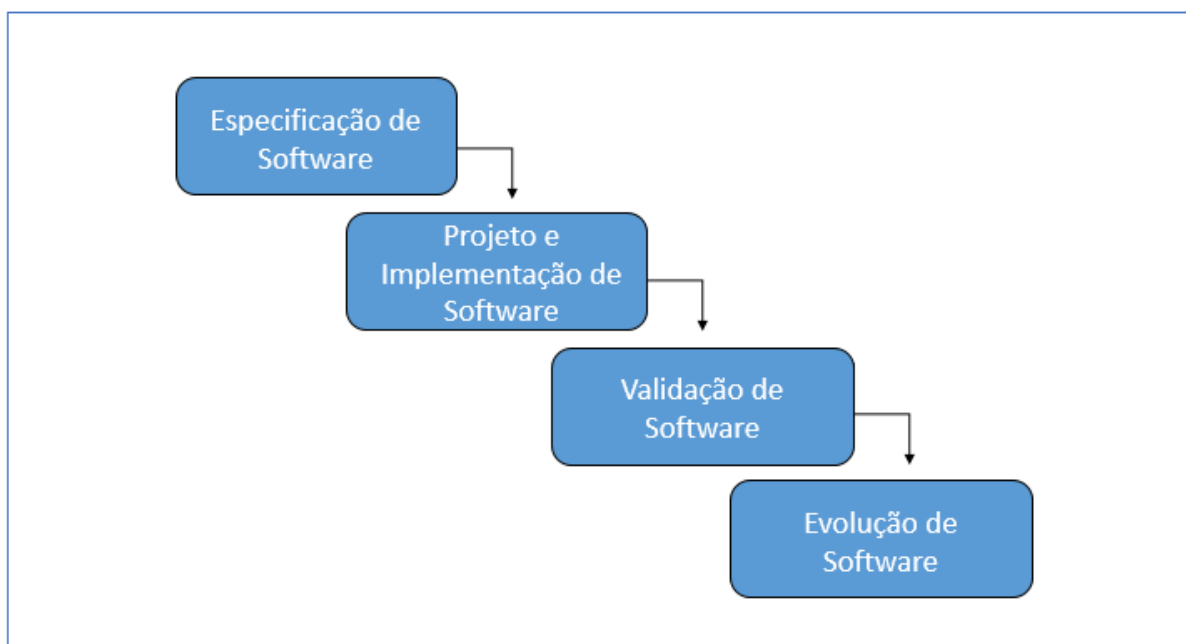
Projeto e implementação de software. O software deve ser produzido para atender às especificações desejadas pelo cliente.

Validação de software. O software deve ser validado e testado para garantir que realmente atende às demandas do cliente.

Evolução de software. O software deve ter características de evolução para atender posteriormente às necessidades de mudança dos clientes (SOMMERVILLE, 2011, p. 18).

Essas atividades compõem todos os processos de *software*. Que na prática, são atividades complexas, que incluem subatividades como validação de requisitos, projeto de arquitetura, testes unitários, dentre outros. Também existem as atividades que dão suporte ao processo, como documentação e gerenciamento de configuração de software. (SOMMERVILLE, 2011). A seguir, a figura 2 apresenta um infográfico para melhor compreensão da sequência das fases genéricas:

Figura 2 - Fases genéricas da engenharia de software



Fonte: Adaptado de Somerville (2011).

Especificação de Software: o início de todo e qualquer projeto deve ser planejado contendo as descrições de todas as atividades, partindo de suas características distintivas, servindo para decidir, determinar, e resolver algum problema. Nessa etapa os engenheiros definem o software a ser produzido e as restrições de sua operação. Todo projeto precisa ter reuniões iniciais, onde serão feitas discussões gerais sobre o que é necessário. O objetivo dessas reuniões é

entender o que o cliente precisa, o porquê ele sente a necessidade daquilo e o quão disposto ele está a investir neste projeto. Na engenharia de software se requer que as atividades para desenvolver o software sejam feitas de forma planejada, gerenciada, com pessoal capacitado, trabalhar com custos e prazos estimados e utilizando teorias, métodos, técnicas e ferramentas adequadas.

A criação de um PDS significa determinar de forma precisa e detalhada quem faz cada atividade, quando e como. Um processo pode ser visto como uma instância de um método com suas técnicas e ferramentas associadas, elaborado durante a etapa de planejamento, no qual as atividades que o compõem foram determinadas aos membros da equipe de desenvolvimento, com prazos definidos e métricas para se avaliar como elas estão sendo realizadas (LEITE, 2006).

A funcionalidade de um *software* determina as características de um sistema computacional, retratando suas funções necessárias para as soluções dos problemas em um domínio específico. Temos que diferenciar funcionalidade de funcionamento, pois se refere a como um *software* realiza suas funções, através da estrutura do *software* e seus tipos de dados e algoritmos do programa fonte determina este funcionamento.

O modelo funcional de uma aplicação determina a funcionalidade de um sistema, também chamado de modelo de funcionalidade. O mesmo é elaborado na fase de especificação funcional de *software* diante das informações colhidas durante a análise de requisitos com os usuários. Todos os conceitos básicos e técnicos são decididos na definição, etapa inicial de qualquer projeto, a questão do sistema operacional que vai ser usado, as ferramentas, o hardware, as funcionalidades, a interface gráfica, a usabilidade, os requisitos funcionais e não funcionais, entre outras que se julgue necessária.

Projeto e implementação de software: depois de todas as escolhas feitas na especificação, identificando os cenários específicos, a metodologia, sabendo o objetivo central do produto, a expectativa e necessidade do cliente e usuários em relação ao produto. Diante de tudo isso pode se iniciar o desenvolvimento, que é a codificação que resultará em um produto digital.

DS é um trabalho complexo que envolve criatividade e conhecimento acerca de um processo construtivo e principalmente, colaborativo. Para que seja desenvolvido com eficácia conta com um conjunto de profissionais interdisciplinares e

capacitados que utilizam recursos técnicos e ferramentas apropriadas para que possa ser entregue um produto final de qualidade.

No decorrer do desenvolvimento poderá acontecer algumas adversidades, em que o gerente⁴ deve sempre está preparado para resolver qualquer situação, seja alocar uma nova tarefa, novos requisitos junto com o cliente, pode também acontecer falha de comunicação, falta de conhecimento de algum membro da equipe.

Validação de Software: A etapa de validação SW consiste em assegurar que de fato ele seja adequado e que atende às necessidades, ou seja, a confirmação de que o mesmo cumpra suas especificações. A validação é uma atividade de extrema importância, a qual envolve a análise dos requisitos funcionais e não funcionais. E também é a certificação de que o sistema foi desenvolvido sobre as características desejadas e se atende as necessidade e expectativas do cliente.

Também pode se incluir dentro da validação de software, a inspeção já que é o ato de analisar as representações do sistema, analisar os documentos de requisitos, diagramas e código-fonte. A inspeção é recomendável que se utilize dela em todas as etapas do processo (SOMMERVILLE, 2007).

A etapa de validação contribui para diminuir e prevenir de possíveis erros nos sistemas. As falhas podem ser detectadas em uma das principais etapas da validação – a análise e gerenciamento de risco, lembrando que o gerenciamento de riscos deve ser periódico e constar em cronograma de revalidação dos sistemas.

Dependendo da forma estrutural de documentação da empresa, a validação poderá ser geral ou específica. O mesmo descreve o que a empresa pretende fazer, quais sistemas serão validados e como ocorrerão. Neste documento inclui, entre outros itens, a estratégia de validação, as variações de mudanças, as principais atividades e as responsabilidades.

Evolução de Software: Abrangente do termo suporte e manutenção de SW, a evolução atua auxiliando organizações e profissionais a melhor conduzir seus esforços quando necessitam manter seus produtos. A evolução pode ser uma manutenção onde podemos corrigir erros na utilização do sistema, sabendo que um

⁴ Um gerente de projetos deve ter organização, agilidade na tomada de decisões e visão para calcular riscos. Deve ser inspirador para sua equipe, saber planejar minuciosamente cada passo, ser flexível para mudar de estratégia e rápido para reorganiza-las.

software não termina após a sua implantação, ele é feito para durar muito tempo e por isso existe sempre a necessidade de se fazer manutenções.

Defeitos não são introduzidos pelo tempo nem pela sua utilização, os defeitos que possam aparecer eles já existiam, antes do sistema começar a ser produzido. E por algum detalhe, não foi detectado em fases anteriores.

Definidas por Pfleeger (2004) a manutenção não se caracteriza apenas por correções, existem três tipos principais de manutenção que é: adaptativas, corretivas e evolutivas.

Adaptativas: são mudanças que visam adaptar o SW a uma nova tecnologia, novo ambiente externo. Exemplo: mudanças de leis ou regras, definidas pelo governo ou órgão reguladores. Manutenção desse tipo referem-se a adequar o software ao seu ambiente externo.

Corretivas: semelhante ao que o nome já diz, servem para eliminar as falhas encontradas em produção. É comum encontrar, principalmente quando o processo de desenvolvimento não se preocupou de maneira adequada com a qualidade do SW. Manutenções assim visam corrigir defeitos de funcionalidade, o que inclui acertos emergenciais de programa.

Evolutivas: são alterações que visam agregar novas funcionalidades e melhorias para usuários que a solicitaram (PFLEEGER, 2004, p.535).

As correções são úteis na manutenção do *software*, sendo ações auxiliares a equipe, na prevenção de falhas, erros ou mudanças que possam surgir, adequações ao ambiente externo.

Evolução do software é de suma importância, uma vez que as organizações investem grandes valores em seus softwares e são totalmente dependentes desses sistemas. Seus sistemas são ativos críticos de negócios, e as organizações devem investir nas mudanças de sistemas para manter o valor desses ativos (Sommerville, 2011). Corrobora com essa perspectiva Erlikh apud Sommerville (2011, p.164) “sugere que 85% a 90% dos custos organizacionais de software são custos de evolução, os custos para mudança de software requerem grande parte do orçamento de TI para todas as empresas.

Portanto, a evolução de um sistema raramente pode ser considerada de forma isolada. Alterações no ambiente levam a mudanças nos sistemas que podem, então, provocar mais mudanças ambientais.

Durante a evolução, o software é usado com sucesso, e existe um fluxo constante de propostas de alterações de requisitos. Mas, no entanto, como o software é alterado, sua estrutura tende a degradar e as mudanças ficam mais e mais caras. Isso ocorre com frequência depois de alguns anos de uso, quando outras mudanças

ambientais, como o hardware e sistemas operacionais, também são necessárias. Em algum estágio do ciclo de vida, o software chega a um ponto de transição em que mudanças significativas e implementação de novos requisitos se tornam menos rentáveis (Sommerville, 2011).

Essa atividade requer algumas observações de etapas consideradas adequadas para obter um resultado positivo. Primeiramente avaliar a documentação de código, a arquitetura, estrutura de dos e interface. Para que assim possa identificar as modificações necessárias e avaliar seus impactos, em seguida, realizar as modificações e testá-las.

2.4 Trabalhos Relacionados

Nesta seção são apresentados o estado da arte a partir de trabalhos que se relacionam aos temas abordados nesta pesquisa, porém, as investigações de Souza (2018), Pereira (2018) e Vargas (2017) divergem da proposta e objeto de pesquisa.

2.4.1 Investigando a aplicação de metodologias ágeis em desenvolvimento de softwares por empresas localizadas no sertão paraibano

Esta pesquisa foi realizada por Raket Pereira de Souza em 2018, apresentada na Universidade Estadual da Paraíba – UEPB para obtenção do grau de Bacharel em Ciência da Computação, a pesquisa teve como objetivo identificar o cenário atual das empresas localizadas no sertão da Paraíba quanto a aplicação de metodologias ágeis, apresentando o uso de métodos ágeis como processo importante para oferecer sistemas com maior qualidade, redução de riscos e a entrega em um curto prazo. Para a coleta de dados foi aplicado um *survey* descritivo com uma amostra de cinco empresas de desenvolvimento de *software* no sertão paraibano. Os resultados apontaram que dentre as cinco empresas duas não aplicam métodos ágeis, duas aplicam parcialmente e apenas uma utiliza os princípios ágeis em seu processo de desenvolvimento (SOUZA, 2018).

2.4.2 Análise da importância dos requisitos no desenvolvimento de software utilizando artefatos do scrum com o apoio da engenharia de requisitos

Trabalho de conclusão de curso em Ciência da Computação de Alcinael Fernandes Pereira, apresentado em 2018 na Universidade Estadual da Paraíba, este trabalho teve como objetivo a utilização de artefatos do Scrum voltados para requisitos como referências nos processos que se destacam na engenharia de software: engenharia de requisitos e o Scrum. Para chegar no resultado desejado são utilizados processos e técnicas da engenharia de requisitos inseridas no escopo do *framework* Scrum que se baseia nos princípios ágeis mostrando que é possível trabalharem juntas, sem que demande mais tempo para o cliente e a equipe, inibindo a possibilidade da perda de prazo. Em conclusão nesta pesquisa foi dado de forma gradativa, pois diversos conceitos e meios a priori desconhecidos foram sendo conhecidos e caracterizados para melhor compreensão, além do que através desse estudo foi possível complementar o conhecimento obtido no meio acadêmico (PEREIRA, 2018).

2.4.3 Avaliação e adoção de uma metodologia de gerenciamento de mudanças nos processos de manutenção de software

Este trabalho foi apresentado por Diana Rodrigues de Vargas como requisito para a obtenção do Grau de Bacharel em Sistemas de Informação da Universidade de Caxias do Sul e tem como objetivo organizar o processo de manutenção de software de uma equipe de desenvolvimento através da criação de um plano de gerenciamento de mudanças baseados nas melhores práticas da *Information Technology Infrastructure Librar* – ITIL. Plano este aplicado e avaliado através de vários indicadores com o intuito de identificar se o mesmo melhorou o cumprimento dos prazos acordados com os clientes. A coleta de dados foi feita por meio da observação no dia-a-dia de uma equipe de desenvolvimento, já que esse trabalho é um estudo exploratório (do tipo estudo de caso) e essa equipe foi utilizada para validar o plano proposto. O objetivo foi alcançado, pois foi possível perceber diversas melhorias nos processos de atendimento das solicitações. Dessa forma, o plano de

gerenciamento de mudanças aplicado foi considerado bem-sucedido porque trouxe vários benefícios para a organização (Vargas, 2017).

3. METODOLOGIA

O presente capítulo está organizado em sete seções, nas quais contemplam o caminho metodológico e a conceituação dos métodos adotados para a presente investigação. Assim, as seções apresentadas, respectivamente, tipo de pesquisa, local da pesquisa, população e amostra, critérios de inclusão e exclusão, instrumentos de coleta de dados e procedimentos de coleta de dados.

3.1 Tipo de Pesquisa

Este trabalho iniciou com uma pesquisa bibliográfica que segundo Marconi e Lakatos (2006, p.71) "a pesquisa bibliográfica não é mera repetição do que já foi dito ou escrito sobre certo assunto, mas propicia o exame de um tema sob novo enfoque ou abordagem, chegando a conclusões inovadoras".

A pesquisa bibliográfica proporciona ao pesquisador e leitor entender os conceitos relacionado ao tema abordado, buscando bases e informações novas para inserir aspectos a serem analisados, bem como delimitar o objetivo pesquisado no trabalho para o alcance da fundamentação teórica acerca do tema.

Após o referencial, a investigação foi composta por um estudo de caso de natureza descritiva a partir de uma abordagem mista (quantitativa e qualitativa). De modo geral quantitativa é passível de ser medida em escala numérica e qualitativa não (ROSENTAL; FRÉMONTIER-MURPHY, 2001).

Este método (quantitativo) caracteriza-se pelo emprego da quantificação, tanto nas modalidades de coleta de informações, quanto no tratamento dessas através de técnicas estatísticas, desde as mais simples até as mais complexas. Já o enfoque qualitativo tem características que se fazem necessários explicitar: o pesquisador é o instrumento-chave; o ambiente é a fonte direta dos dados; não necessita do uso de técnicas e métodos estatísticos; possui caráter descritivo; o resultado não é o foco da abordagem, mas sim todo o processo e seu significado, ou seja, o principal objetivo é a interpretação do fenômeno objeto de estudo (GODOY, 1995B, SILVA; MENEZES, 2005).

Adotou-se ainda o objetivo metodológico descritivo, uma vez que, conforme Vergara, esta pesquisa

[...] expõe características de determinada população ou de determinado fenômeno. Pode também estabelecer correlações entre variáveis e definir sua natureza. Não tem compromisso em explicar os fenômenos que descreve, embora sirva de base para tal explicação (VERGARA, 2004, p. 47).

A pesquisa descritiva se faz levando em consideração os aspectos formulados nas perguntas que norteiam e estabelecem a pesquisa em relação as variáveis definidas no objetivo do trabalho. Cabe ao pesquisador fazer a descrição, estudo, análise e interpretação dos dados sem que faça interferência dos mesmos.

3.2 Local da Pesquisa

A pesquisa foi realizada, em empresas na cidade de Patos – PB, município brasileiro no estado da Paraíba, localizado na microrregião de Patos, na mesorregião do Sertão Paraibano, distante 307 km da capital João Pessoa.

3.3 População e Amostra

Como base, adotamos o último relatório do Sebrae-PB, entre 2015 e 2017, na qual especifica que a cidade de Patos, possui nove (9) empresas que trabalham com desenvolvimento de *software*, dentre estas cinco (5) empresas consentiram à participação e aplicação da pesquisa. Sendo realizada com algum funcionário da empresa habilitado a responder o questionário de pesquisa, que fosse um gerente, desenvolvedor, analista ou testador.

3.4 Critérios de Inclusão e Exclusão

Dentre as empresas da cidade de Patos, cinco (5) atenderam ao critério de inclusão que era possuir inscrição estadual, e assinatura do Termo de Consentimento Livre e Esclarecido – TCLE. Os representantes que voluntariamente desejarem e assinarem o TCLE estarão aptos (inclusos) para participação/colaboração da investigação. A não assinatura do termo e estando em situação irregular ou não inscrito no sistema governamental da Secretaria Estadual da Paraíba de Receita, já caracteriza a exclusão dos possíveis participantes.

Vale ressaltar, que as quatro (4) empresas que não participaram da pesquisa não atenderam aos critérios de inclusão. Outro critério de exclusão evidenciado no contato com os representantes foi que mesmo constando no relatório do Sebrae-PB, havia uma empresa que não trabalha diretamente com desenvolvimento, apenas com suporte e manutenção.

3.5 Instrumento de Coleta de Dados

A coleta de dados foi realizada por meio de questionário aplicado nas empresas de desenvolvimento de *software* do município de Patos - PB, com a finalidade de saber como ocorre o processo de DS. Segundo Gil (1999) o questionário pode ser definido como:

[...] técnica de investigação composta por um número mais ou menos elevado de questões apresentadas por escrito às pessoas, tendo por objetivo o conhecimento de opiniões, crenças, sentimentos, interesses, expectativas, situações vivenciadas etc (GIL, 1999, p.128).

O questionário é composto por dez perguntas objetivas e duas subjetivas. O questionamento subjetivo elucida às opiniões e percepções dos investigados frente aos objetivos deste trabalho. Assim, foi realizada uma pesquisa de campo, a qual contribuiu para entender um pouco mais do desenvolvimento de sistemas, entender quais os pontos relevantes são utilizados, como também sua utilidade e aplicação. Finalmente a pesquisa detectou as metodologias e fases, percebidas por quem mais as utiliza no dia a dia, visando assim o objetivo principal do trabalho.

Segundo Gerhardt e Silveira (2009) às questões mistas (fechadas e abertas) são aquelas em que dentro de uma lista predeterminada, há um item aberto, por exemplo, "outros".

3.6 Procedimento de Coleta de Dados

O questionário foi aplicado com uma amostra de cinco (5) empresas de desenvolvimento de Software, situadas no município de Patos-PB, escolhidas de acordo com a função de relevância para o assunto aqui tratado. A coleta das

informações foi realizada no mês de maio de 2019, seguindo o roteiro proposto no cronograma do projeto.

Após a coleta dos dados, foi feita a análise dos resultados, com o objetivo de analisar os critérios adotados pelas empresas baseando-se pelas fases genéricas de criação de *software*.

As empresas analisadas assinaram um Termo de Consentimento Livre e Esclarecido para a participação e divulgação dos resultados, mas sendo mantido em sigilo os nomes das empresas, as mesmas responderam um questionário (Apêndice A) para a formulação dos objetivos deste trabalho.

4. RESULTADOS E DISCUSSÕES

Serão discutidos e explanados neste capítulo os resultados obtidos com a análise da aplicação do questionário de pesquisa junto as empresas desenvolvedoras de sistemas comparando-as com literatura apresentada neste trabalho.

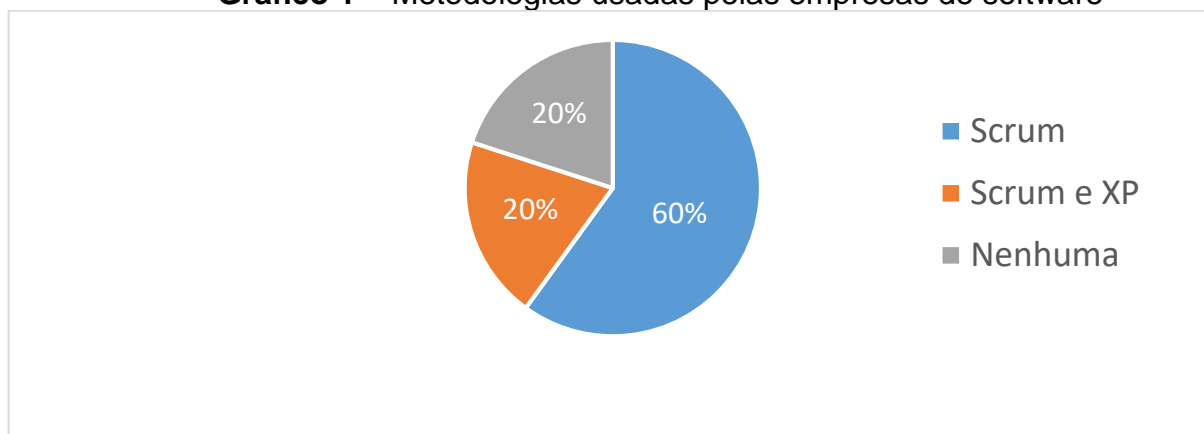
4.1 Caracterização da amostra

Os funcionários que responderam a esta pesquisa estão caracterizados da seguinte forma: Um (1) gerente, dois (2) desenvolvedores, um (1) analista e um (1) testador, são profissionais que já trabalham nas empresas analisadas entre oito (8) meses a quatro (4) anos. Três (3) funcionários relataram que já estiveram alocados em outras atividades diferentes dentro da empresa.

4.2 Uso de uma metodologia no processo de desenvolvimento de *software*

Conforme Soares (2004) metodologia de desenvolvimento de software é um conjunto de etapas que auxilia no processo de criação de um sistema. Dentre os dois grandes grupos de que autor fala estão os modelos tradicionais e ágeis. O gráfico 1 mostra qual a metodologia usada pelas organizações, evidenciando o método Scrum.

Gráfico 1 – Metodologias usadas pelas empresas de software



Fonte: Dados da Pesquisa (2019).

As três empresas (**A**, **B**, **C**⁵) utilizam apenas o Scrum como metodologia para o desenvolvimento de software; a organização **D** utiliza o Scrum e Extreme Programming, e a empresa **E** não utiliza nenhuma metodologia, ambas correspondem a 20% cada. A empresa **E**, relatou que não utiliza nenhuma metodologia pois trabalha com Enterprise Resource Planning – ERP, que é o planejamento de recursos da empresa, apenas o implementam com algumas solicitações feitas pelos clientes, sem seguir nenhuma metodologia.

O Scrum e XP são bastante parecidos, no entanto, o XP demonstra ser mais radical com base em valores de simplicidade, comunicação, *feedbacks* que permitiu que todos os membros possam acompanhar o desenvolvimento do projeto. E já o Scrum é um método bastante popular, é um processo de gestão e controle que exclui a complexidade e foca na construção de *software* que atenda às necessidades do negócio em questão.

As duas metodologias mencionadas, possuem características semelhantes e todas duas seguem o padrão que rege as fases genéricas, segundo Sommerville (2011). Inicia-se com a especificação de *software*, onde os desenvolvedores vão se reunir com o cliente, traçar as metas e requisitos; Projeto e implementação de *software*, segunda etapa onde o desenvolvedor vai iniciar a codificação sobre o que foi elucidado na fase anterior; Validação de *software*, após o término da codificação acontecem os testes e validação do mesmo para a confirmação de que foi construído sobre conformidade com os requisitos da especificação; *Evolução de software*, por último a evolução abrange alguma manutenção, correção que se julgue necessário fazer após a entrega do sistema para o cliente.

4.3 Critérios de Utilização e método próprio

Perguntados sobre quais os critérios levam a escolher tal metodologia e se acrescentavam algum método próprio da empresa os funcionários formularam respostas breves e objetivas, as empresas A, B e C que utilizam o Scrum relataram

⁵ Como forma de resguardar a identidade das organizações serão utilizadas letras para identificação das empresas.

respectivamente da seguinte forma: “*Por ser a melhor entre as que conheço*”; “*Pela necessidade do cliente*”; “*Por ser de fácil utilização, pelas etapas detalhadas, fazendo com que o software seja produzido de acordo com planejado*”. As três (3) não acrescentam método próprio, somente segue os passos da metodologia citada.

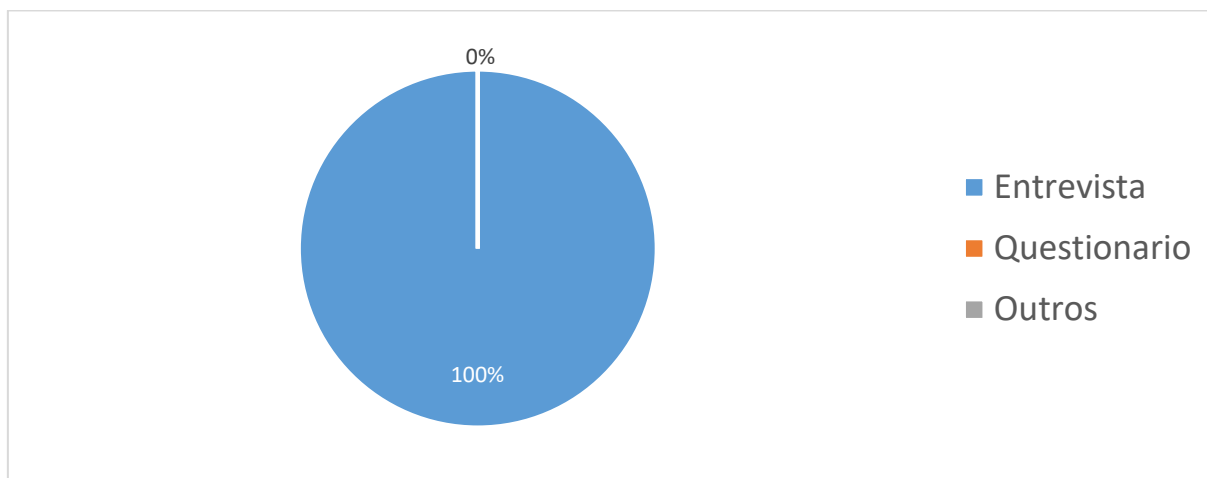
A empresa D que faz uso das duas metodologias – Scrum e XP – relatou que “*Dependendo do sistema e também do desenvolvedor ele fica livre para escolha, já que dentro da empresa não se faz exigência quanto ao uso padrão de apenas uma*”. E a empresa também não acrescenta nada de métodos próprios.

A organização E, trabalha com *software* de gestão, não utilizam metodologias de desenvolvimento de sistemas, apenas adequada o software com as características do cliente. Usam apenas como métodos próprios a elucidação de requisitos através de entrevistas com os clientes.

Percebe-se que as organizações A, B e C, fazem suas escolhas divergindo-se do que elucida Sommerville (2011), onde ele diz que a metodologia deve ser escolhida de acordo com o tipo de sistema a ser desenvolvido e não por ser a mais fácil, nem a que o desenvolvedor conhece, porque o método deve ser de acordo com a necessidade do sistema. Na empresa D, isso é mais perceptível tendo em vista que o desenvolvedor escolhe qual o método ele vai utilizar porque não existe um padrão na empresa, o que deixa mais evidente que ele deve escolher de acordo com o sistema a ser desenvolvido.

4.4 Especificações de requisitos

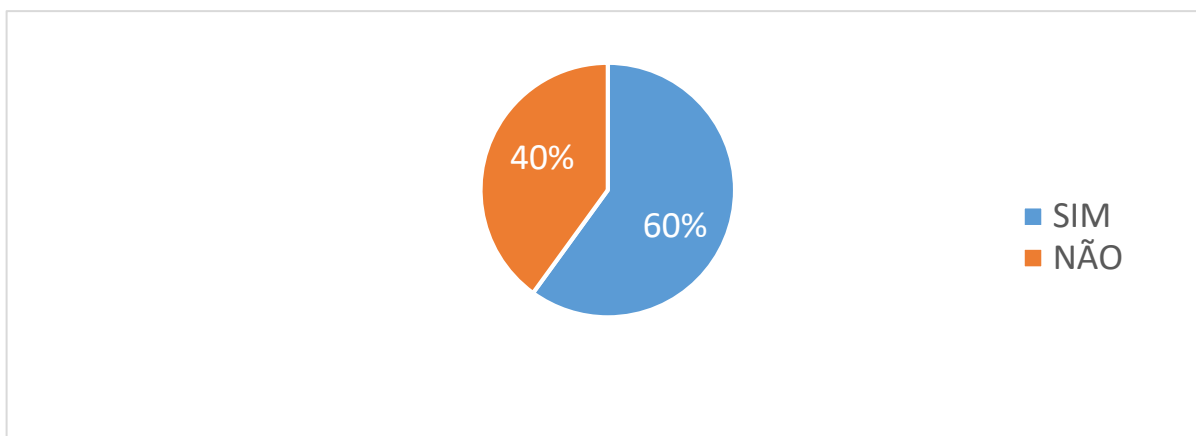
Sommerville (2011) deixa claro a importância da especificação de requisitos, pois nessa etapa o projeto se torna particularmente crítico e a ocorrência ou produção de erros, consequentemente afetará no projeto e na implementação do sistema. Diante disso a 8ª pergunta do questionário versava sobre qual método era utilizado para especificação dos requisitos como pode ser visto no gráfico a seguir.

Gráfico 2 – Especificações de requisitos nas empresas

Fonte: Dados da Pesquisa (2019).

Como observado no gráfico 2, 100% da amostra, ou seja, as cinco (5) organizações pesquisadas utilizam a entrevista como forma de especificar os requisitos junto aos clientes. Corroborando com os pensamentos de Sommerville (2011) em que a entrevista é um método de coleta de dados, dentro das especificações de requisitos que é a primeira etapa das fases genéricas. E por isso as organizações consideram que a entrevista é uma forma mais detalhada de colher as informações necessárias para caracterizar os requisitos. Porém as alternativas da pergunta também tinha como sugestão o questionário, não sendo citado por nenhuma empresa.

Em questionamento na pergunta 9ª sobre como formulam esses requisitos, todas as empresas disseram fazer visitas aos clientes no ambiente em que o software será usado, uma forma melhor de entenderem em outros aspectos mais técnicos como o cliente deseja ter em seu sistema.

Gráfico 3 – Acompanhamento dos clientes no projeto

Fonte: Dados da Pesquisa (2019).

A pergunta 10^a, a qual investiga se o cliente acompanha o projeto durante todas as etapas ou só na finalização, o gráfico evidencia que 60% das organizações tem a prática de sempre estarem mostrando o desenvolvimento do sistema para o cliente. Assim dando sequência as próximas etapas com uma garantia maior de que os requisitos até o momento foram compreendidos e estão sendo executados corretamente.

4.5 Validação do *software*

Validação de *software* é a terceira fase genérica e de suma importância, pois assegura que o sistema foi desenvolvido adequadamente, atendendo todas às necessidades exigidas pelo cliente nas especificações dos requisitos seguindo os posicionamentos de Sommerville (2011).

Na 11^a pergunta do questionário de pesquisa a qual se referia como ocorre a validação de *software* na organização, as respostas detalhadas no quadro abaixo.

Quadro 1 – Validação de *software* pelas empresas analisadas

Empresa A	Através de testes.
Empresa B	Juntamente com cliente, se atende os requisitos que foram levantados
Empresa C	Na parte final o cliente vem até a empresa e acompanha juntamente com o testador o seu sistema.

Empresa D	Realizamos testes junto com o cliente, caso necessite alguma mudança ou que tenha apresentado algum erro, já seria corrigido de imediato.
Empresa E	A validação do software finaliza quando se seguiu todas as etapas do processo como: requisitos, pedidos de implementação e solicitações dos clientes. Já que trabalhamos com ERP o gerenciador se adapta a necessidade de cada cliente.

Fonte: Dados da Pesquisa (2019).

A empresa A valida seus produtos com testes, não relatando se ocorrem com a presença do cliente ou se é apenas por um testador da equipe. O que difere da resposta da empresa B, que deixa claro que ao realizarem a validação a mesma é feita com a presença do cliente, em consonância com Sommerville (2007) que evidencia o processo de validação com a inspeção dos requisitos e com o cliente. As organizações C e D também realizam testes com um testador juntamente com o cliente, para que possam identificar algum erro, mudança ou necessidade que o cliente queira acrescentar. A organização E, consiste sua validação em conformidade com os requisitos e pedidos do cliente inicialmente, assim sendo o cliente não participa ativamente dos testes.

A 12ª pergunta do questionário, onde versava sobre modificações ou atualizações que as empresas continuavam prestando aos seus clientes. A mesma, tinha como intuito elucidar a última etapa das fases genéricas que é a evolução de *software*.

Depois do *software* entregue ao cliente, as empresas A, B, C, D e E, continuam prestando serviços de evolução e manutenção ao cliente que desejar fazer alguma alteração ou que julgue ter algum erro. Todos os respondentes das empresas analisadas relataram prestar esse serviço por tempo determinado em contrato ou caso sejam contratadas novamente. Conforme Sommerville, (2011) espera-se desse processo uma evolução do software adequando as mudanças de hardware e atualizações de mercado.

5. CONSIDERAÇÕES FINAIS

Diante do que foi apresentado neste trabalho, constatou-se a preocupação de diversos autores e pesquisadores com a forma que é desenvolvido *softwares*, como é seguido os métodos dos processos de desenvolvimento de *software*, os requisitos e os planejamentos, respondendo ao primeiro objetivo específico desta pesquisa.

Para atingir os objetivos específicos – segundo e terceiro – deste trabalho no qual foi obtido através do questionário aplicado com um representante da empresa, para que assim pose-se confrontar com a literatura.

O último objetivo específico da pesquisa, onde foram alcançados através do questionário aplicado, percebeu-se que num universo de cinco empresas os critérios que cada uma utiliza para estimar o modelo de desenvolvimento são semelhantes.

Diante da análise dos dados da pesquisa, pôde-se perceber que apenas uma empresa deixa livre para o desenvolvedor a utilização de diferentes metodologias, sendo em contexto com a definição de Sommerville (2011) que a metodologia deve ser de acordo com o sistema a ser produzido.

Percebeu-se diante das respostas do universo investigado a presença das etapas elucidadas como fases genéricas, uma vez que os processos ocorrem não necessariamente de forma sequencial, depois de analisarem todos as especificações e requisitos, iniciam a implementação seguindo o cronograma do projeto e as sprints. Executam testes como etapa de validação a fim de melhorias e de encontrar possíveis erros, todas as empresas prestam serviços de manutenção aos clientes após a entrega do produto, acompanhando sua evolução e possíveis mudanças.

Com todos os aspectos observados, constata-se que as empresas desenvolvem seus produtos utilizando-se métodos de metodologias, seguindo cronologicamente as etapas das mesmas. Os resultados apresentam os ciclos de vida que cada empresa usa, os processos desde a inicialização até os testes.

Em contribuição, está pesquisa tem valor significativo para enaltecer a importância de desenvolver sistemas sobre a regência de alguma metodologia, que sejam construídos de forma correta e ao final possa ser satisfatória aos usuários.

Em sugestão para trabalhos futuros, pode ser feito um estudo de caso mais aprofundado, investigando e acompanhando a produção de *software* dentro de uma das empresas e medir a qualidade e satisfação dos clientes.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 12207, **Engenharia de sistemas e software - Processos de ciclo de vida de software**. Rio de Janeiro: 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 9126-1, **Engenharia de software - Qualidade de produto: Parte 1: Modelo de qualidade**. Rio de Janeiro: 2001.

BOURQUE, P.; FAIRLEY, D. **SWEBOK 3.0 Guide to the Software Engineering Body of Knowledge**. [S.l.]: IEEE Computer. 2014 Disponível em: https://pt.wikipedia.org/wiki/Qualidade_de_software#cite_note-6-1. Acesso em: 26 de set. 2018.

COMPUTERWORLD, **Mercado de Software e Serviços de TI deve Manter Crescimento até 2021, prevê estudo**. 2018. Disponível em: <https://computerworld.com.br/2018/04/23/mercado-de-software-e-servicos-de-ti-deve-manter-crescimento-ate-2021-preve-estudo/>. Acesso em: 19 mar. 2019.

EMIS INSIGHTS. **Mercado de software e serviços de TI deve manter crescimento até 2021, prevê estudo**. 2018. Disponível em: <https://computerworld.com.br/2018/04/23/mercado-de-software-e-servicos-de-ti-deve-manter-crescimento-ate-2021-preve-estudo/>. Acesso em: 04 abr. 2019.

FERREIRA, A. B. H. **Aurélio século XXI: O dicionário da Língua Portuguesa**. 3. ed. rev. e ampl. Rio de Janeiro: Nova Fronteira, 1999.

FREEMAN, E. Stakeholder capitalism and value chain. **E.M.Journal**, 1984.

GERHARDT, T. E.; SILVEIRA, D. T. **Metodologia de Pesquisa**. Rio Grande do Sul. Rio Grande do Sul: Ufrgs, 2009.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 5.ed. São Paulo: Atlas, 1999.

GODOY, A. S. Introdução a pesquisa qualitativa e suas possibilidades. **Revista de Administração de Empresas**. São Paulo, v. 35, n. 2, p. 57-63, Mar./Abr. 1995.

KEMCZINSKI, A.; KERN, V.; CASTRO, J. E. A engenharia de requisitos no suporte ao planejamento de treinamento. **Revista Produção Online**, Santa Catarina, v. 1, n. 1, p.1-10, out. 2001. Disponível em: www.producaoonline.inf.br. Acesso em: 04 maio 2019.

KOSCIANSKI, A.; SOARES, M. S. **Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. Novatec Editora, 2007.

LEITE, J. C. **Planejamento e Gerenciamento de Projetos de Software**. Rio Grande do Norte. 2006.

MAINART, D. A.; SANTOS, C. M. **Desenvolvimento de Software: Processos Ágeis ou Tradicionais? Uma visão crítica**. 2010. Disponível em: http://www.enacomp.com.br/2010/cd/artigos/completos/enacomp2010_4.pdf. Acesso em: 16 out. 2019.

MARCONI, M. A.; LAKATOS, E. M. **Técnicas de Pesquisa**: planejamento e execução de pesquisas, amostragens e técnicas de pesquisa, elaboração, análise e interpretação de dados. São Paulo: Atlas, 2006.

MARTINS, D. A.; RAMOS, A. S. M.; LOPES, F. D. Caracterização do processo de desenvolvimento de software: um estudo de múltiplos casos da região metropolitana de recife. In: XXXI Encontro Nacional De Engenharia De Produção. 2011, Belo Horizonte. **Anais do Encontro Nacional de Engenharia de Produção - Enegep**. Belo Horizonte: Abepro.

MORIMOTO, C. E. **Terminal Burro**. 2005. Disponível em: <https://www.hardware.com.br/termos/terminal-burro>>. Acesso em: 02 maio 2019.

PEREIRA, Alcinael Fernandes. **Análise da importância dos requisitos no desenvolvimento de software utilizando artefatos do scrum com o apoio da engenharia de requisitos**. 2018. 62 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Estadual da Paraíba, Campina Grande, 2018.

PFLEEGER, S. L. **Engenharia De Software Teoria E Prática**. São Paulo: Pearson, 2004. p.535.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. Rio de Janeiro: McGraw Hill, 2006.

PRESSMAN, R. S. **Engenharia de Software**. Rio de Janeiro, RJ: Makron Books, 2002.

PRESSMAN, R. S.; MAXIM, B. **Engenharia de Software**. McGraw Hill Brasil, 2016.

ROSENTAL, C.; FRÉMONTIER-MURPHY, C. **Introdução aos métodos quantitativos em ciências humanas e sociais**. Porto Alegre: Instituto Piaget, 2001.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. 4 ed. Florianópolis: UFSC, 2005.

SOARES, M. S. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. **INFOCOMP**, [S.l.], v. 3, n. 2, p. 8-13, nov. 2004. ISSN 1982-3363. Disponível em: <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68> . Acessado em: 17 mar 2019.

SOMMERVILLE, I. **Engenharia de Software**. 8. ed. São Paulo: Pearson Education, 2007. 568 p.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Education, 2011. 529 p.

SOUZA, Raket Pereira de. **INVESTIGANDO A APLICAÇÃO DE METODOLOGIAS ÁGEIS EM DESENVOLVIMENTO DE SOFTWARES POR EMPRESAS LOCALIZADAS NO SERTÃO PARAIBANO**. 2018. 68 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Estadual da Paraíba, Patos, 2018.

VARGAS, Daiana Rodrigues de. **Avaliação e adoção de uma metodologia de gerenciamento de mudanças nos processos de manutenção de software**. 2017. 135 f. TCC (Graduação) - Curso de Sistema de Informação, Universidade de Caxias do Sul, Caxias do Sul, 2017.

VERGARA, S. C. **Projetos e relatórios de pesquisa em administração**. 5. ed. São Paulo: Atlas, 2004.

**APÊNDICE A – QUESTIONÁRIO PARA ANÁLISE DAS EMPRESAS DE
DESENVOLVIMENTO DE SOFTWARE**



UEPB

**UNIVERSIDADE ESTADUAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS - CCEA - Campus VII
Rua Alfredo Lustosa Cabral, S/N Bairro: Salgadinho, Patos - PB, 58706-560**

SURVEY: GESTORES E EQUIPE DE DESENVOLVIMENTO DE SOFTWARE

Prezado Colaborador (a),

A presente pesquisa intitulada “Análise do processo de desenvolvimento de Sistemas em empresas de Software no município de Patos-PB ” da aluna Jakeline Bandeira de Oliveira, coordenada pelo professor Sergio Morais Cavalcante Filho, tem como objetivo analisar quais as metodologias são utilizadas no processo de desenvolvimento de software na cidade de Patos-PB, visando identificar quais e como utilizam essas metodologias de desenvolvimento de produtos de software, suas características e práticas. Gostaríamos de convidá-lo(a) para participar e responder algumas informações sobre a empresa em que colabora. Sua participação se dará de forma voluntária e anônima com finalidade exclusiva em colaborar para o sucesso da pesquisa.

Agradecemos imensamente a sua contribuição.

1- QUAL SUA FUNÇÃO NA EQUIPE/EMPRESA?

- () Gerente
- () Desenvolvedor
- () Analista
- () Testador

2- A QUANTOS ANOS ESTÁ TRABALHANDO NESSA EMPRESA?

- () Menos de 1 ano
- () Entre 2 a 4 anos
- () Mais de 4 anos

3- DURANTE ESSE TEMPO, JÁ MUDOU DE FUNÇÃO/CARGO?

- () Sim
- () Não

4-AO SEREM CONTRATADOS PARA DESENVOLVER UM SOFTWARE, UTILIZAM-SE DE ALGUMA METODOLOGIA. SE SIM, QUAL?

- () Sim _____
- () Não

5- TRABALHA EXCLUSIVAMENTE COM ESSA METODOLOGIA ?

- Sim
- Não

6- QUAIS CRITÉRIOS LEVAM A UTILIZAR ESSA METODOLOGIA/PROCESSO?

7- É ACRESCENTADO ALGUM MÉTODO PRÓPRIO A ESSA METODOLOGIA?

- Sim
- Não

8- COMO É FEITA A ESPECIFICAÇÃO DOS REQUISITOS COM O CLIENTE?

- Questionários
 - Entrevistas
 - Outros. Quais?
-

9- REALIZAM VISITAS AOS CLIENTES ANTES DE INICIALIZAR O DESENVOLVIMENTO?

- Sim
- Não

10- AO INICIAREM A IMPLEMENTAÇÃO, O CLIENTE ACOMPANHA AS ETAPAS DESENVOLVIDAS OU SÓ A FINALIZAÇÃO?

- Sim
- Não

11- QUANTO A VALIDAÇÃO DO SISTEMA, COMO ELA OCORRE NA EMPRESA?

12- APÓS A ENTREGA FINAL DO SISTEMA, A EMPRESA ATENDE ALGUMA MODIFICAÇÃO OU ATUALIZAÇÃO DO CLIENTE?

- Sim
- Não

ANEXO A – Termo de Consentimento Livre e Esclarecido

Declaro ser do meu conhecimento e livre vontade à participação na pesquisa presente sobre Análise do processo de desenvolvimento de Sistemas em empresas de Software no município de Patos-PB. A investigação é vinculada ao trabalho de conclusão de curso de Bacharelado em Ciência da Computação da Universidade Estadual da Paraíba – UEPB – *Campus VII*.

Estou ciente de que o questionário é o instrumento de coleta de dados do processo investigativo e que as informações por mim fornecidas serão utilizadas na escrita do trabalho mantendo em sigilo a minha identidade pessoal, bem como a instituição a qual estou vinculado.

(Assinatura do Participante)

(Pesquisador responsável)

ANEXO B – Solicitação de consentimento para realização de pesquisa acadêmica

Declaro que Jakeline Bandeira de Oliveira está matriculada na Disciplina de Trabalho de Conclusão de Curso (TCC) 2, do Curso de Bacharelado em Ciência da Computação da UEPB, *campus* VII, Patos. Desenvolve, sob minha orientação, um estudo monográfico sobre Análise do processo de desenvolvimento de Sistemas em empresas de Software no município de Patos-PB.

O instrumento de coleta de dados do estudo é um questionário que tem por finalidade analisar as empresas de desenvolvimento de *software* no município de Patos-PB.

Nesta perspectiva, solicito que esta Empresa, gentilmente, permita a realização do presente estudo junto aos funcionários. Ressalto ainda que será mantida em sigilo a identidade pessoal dos funcionários e da empresa.

Antecipadamente agradeço pela vossa valorosa colaboração.



SERGIO MORAIS CAVALCANTE FILHO
Professor Orientador – Mat. 729276-5