



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
TOCANTINS**

**CAMPUS PORTO NACIONAL
LICENCIATURA EM COMPUTAÇÃO**

JULIANA DE SOUZA PINHEIRO

**TENDÊNCIA DO MERCADO QUANTO AO USO DOS MODELOS DE
DESENVOLVIMENTO DE SOFTWARE: Uma Experiência Tocantinense**

PORTO NACIONAL

2019

JULIANA DE SOUZA PINHEIRO

**TENDÊNCIA DO MERCADO QUANTO AO USO DOS MODELOS DE
DESENVOLVIMENTO DE SOFTWARE: Uma Experiência Tocantinense**

Monografia apresentada a
Coordenação do Curso Superior de Licenciatura
em Computação do *Campus* Porto Nacional, do
Instituto Federal do Tocantins, como exigência à
obtenção do grau de Licenciado em Computação.

Orientador: Prof. Me. Rafael Miranda Correia

Coorientador: Prof.^a Ma. Shirley Alves Viana
Vanderlei

PORTO NACIONAL

2019

JULIANA DE SOUZA PINHEIRO

**TENDÊNCIA DO MERCADO QUANTO AO USO DOS MODELOS DE
DESENVOLVIMENTO DE SOFTWARE: Uma Experiência Tocantinense**

Monografia apresentada a
Coordenação do Curso Superior de Licenciatura
em Computação do *Campus* Porto Nacional do
Instituto Federal do Tocantins, como exigência à
obtenção do grau de Licenciado em Computação.

Orientador: Prof. Me. Rafael Miranda Correia

Coorientador: Prof.^a Ma. Shirley Alves Viana
Vanderlei

Aprovado em ____/____/____.

BANCA EXAMINADORA

Professor Orientador Me. Rafael Miranda Correia
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins

Professor Coorientador – Ma. Shirley Alves Viana Vanderlei
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins

Professor Convidado – Edilson Leite de Souza
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins

Professor Convidado – Deuzelina Tavares Chagas
Instituto Federal de Educação, Ciência e Tecnologia do Tocantins

Conceito: **10,0**

Dedico este trabalho primeiramente a Deus, que me deu saúde e forças para superar todos os momentos difíceis a que eu me deparei ao longo da minha graduação, e a toda minha família e amigos por me incentivarem a ser uma pessoa melhor e não desistir dos meus sonhos.

AGRADECIMENTOS

A Deus por ter me dado saúde e força para superar as dificuldades.

A este Instituto Federal, seu corpo docente, direção e administração que oportunizaram a janela do vislumbro, um horizonte superior eivado pela acendrada confiança no mérito e ética presente.

Ao meu orientador, Prof. Me. Rafael Miranda Correia, pelo suporte no tempo que lhe coube, pelas suas correções e incentivos que culminaram na conclusão deste projeto.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“Estamos em uma era de que quem não tem conhecimento o suficiente da informática é considerado analfabeto”.

Gilberto Martini Refatti

RESUMO

Este trabalho aborda sobre a tendência do mercado quanto ao uso dos modelos de desenvolvimento de software: Uma experiência tocantinense, objetivando analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software. assim, a problemática foi evidenciada se o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do *Campus* Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados? Nesse sentido, levantou se a hipótese de que a ementa da disciplina de Engenharia de Software do Projeto Político Pedagógico (PPP). do curso de Licenciatura em Computação do *Campus* Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia – IFTO. atende parcialmente a tendência quanto ao uso de modelos e processos. Todas as etapas aqui descritas são frutos de um trabalho minucioso apresentado em formato de monografia. Para a realização deste estudo foram utilizadas pesquisas bibliográficas através de livros, artigos científicos e sites para obtenção de informações e a pesquisa de campo considerada quantitativa, utilizada para coleta de dados. O universo da pesquisa em que o estudo foi desenvolvido está definido com 61 participantes, deste total participaram diversos profissionais da área de desenvolvimento de software conforme discriminados a seguir: 15 da Tecnologia da informação, 14 de Banco de Dados, 8 de Engenharia da computação, 10 de Ciências da computação, 9 de Analise de sistema, 2 de Técnico em informática e 3 de Sistemas para internet. Os resultados obtidos mostram que 69% dos respondentes utilizam modelos tradicionais, enquanto 31% utilizam modelos ágeis. Os modelos ágeis ainda são menos usados, apesar da literatura aponta-los como tendência. A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade, mostrando com isso a relevância desse trabalho.

PALAVRAS CHAVES: Engenharia. Desenvolvimento. Modelos. Processo. Software.

ABSTRACT

This paper discusses the market trend in the use of software development models: A Tocantins' experience, aiming to analyze the current market trend in the use of software development models. thus, the problem was evidenced if the graduate of the Software Engineering discipline of the Licenciatura course in Computing of the Porto Nacional *Campus* is updated according to the current market trend regarding the software development models used? In this sense, the hypothesis was raised that the menu of the discipline of Software Engineering of the Political Project Pedagogical - PPP. of the Licentiate course in Computing of the Porto National *Campus* of the Federal Institute of Education, Science and Technology - IFTO. partially meets the trend of using models and processes. All the steps described here are fruits of a detailed work presented in monograph format. For the accomplishment of this study were used bibliographical research through books, scientific articles and sites to obtain information and the field research considered quantitative, used for data collection. The universe of the research in which the study was developed is defined with 61 participants, of which a number of professionals from the area of software development participated as follows: 15 from Information Technology, 14 from Database, 8 from Computer Engineering, 10 Computer Science, 9 System Analysis, 2 Computer Technician and 3 Internet Systems. The results show that 69% of respondents use traditional models, while 31% use agile models. Agile models are still less widely used, despite the literature pointing them as a trend. The research also showed that 10% of respondents know another model called Scrum, which according to the literature is the model that follows the trend of contemporaneity, thus showing the relevance of this work.

Keywords: Engineering. Development. Models. Process. Software.

LISTA DE FIGURAS

Figura 1 - Modelo cascata na engenharia de software	19
Figura 2 - Área de formação dos respondentes	39
Figura 3 - Modelos de desenvolvimento mais utilizados no Tocantins	40
Figura 4 - Modelos que não foram citados na pesquisa	42
Figura 5 - Modelos Tradicionais x modelos Ágeis	43
Figura 6 - Modelos descrito na ementa / plano de ensino	44

LISTA DE TABELAS

Tabela 1 – Atividades para processo de software	20
Tabela 2 - Modelos de processos prescritivo ou modelos tradicionais	21
Tabela 3 - Modelos de processos especializado	22
Tabela 4 - Modelos de processos ágeis.....	22
Tabela 5 – Vantagens uso do Scrum no desenvolvimento de Software	27
Tabela 6 - Componente Curricular: Engenharia de Software.....	30
Tabela 7 – Características que levaram a utilizarem cada modelo de software	41

LISTA DE ABREVIATURAS E SIGLAS

CH - Carga Horária

IFTO - Instituto Federal de Ciência e Tecnologia do Tocantins

LDB - Lei de Diretrizes e Bases da Educação Nacional

MD – Modelo de Software

MDA - Dirigida pelo Modelo

PPP - Projeto Político Pedagógico

PRAXIS - Processo para Aplicativos extensíveis e interativos

PS – Processo de Software

PX - Programação Extrema

RAD - Desenvolvimento Rápido de Aplicação

RUP - Processo Unificado da Rational

TO - Tocantins

SUMÁRIO

1.INTRODUÇÃO	14
1.1.Problemática da Pesquisa	15
1.2.Objetivos	16
1.2.1.Objetivo Geral.....	16
1.2.2.Objetivos Específicos	16
1.3.Hipótese.....	16
1.4.Justificativa	16
1.5.Organização do Trabalho.....	17
2.FUNDAMENTAÇÃO TEÓRICA	18
2.1.Engenharia de software	18
2.2.Os modelos de desenvolvimento de software.....	19
2.3.A atual tendência de mercado quanto ao uso dos modelos de desenvolvimento de software	24
2.3.1.Metodologia Tradicional x Metodologia Ágil – entenda as diferenças	24
2.3.2.Mudança para Métodos Ágeis de Desenvolvimento de Software	26
2.3.3.Processo de software Scrum.....	27
2.4.Ementa da disciplina de engenharia de software.....	28
2.4.1.Ementa da disciplina de Engenharia de Software	30
2.4.2.Plano de ensino da disciplina de Engenharia de Software	31
2.5.Trabalhos correlacionados.....	32
2.5.1.Caracterização de um Processo de Software para Projetos de Software Livre.....	32
2.5.2.Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software	32
2.5.3.Estação TABA: Uma infra-estrutura para Implantação do Modelo de referência para Melhoria de Processo de Software	33
2.6.Comentários Parciais.....	33
3.PROCEDIMENTOS METODOLÓGICOS.....	34

3.1.Materiais e Métodos.....	34
3.1.1.Levantamento dos modelos de processo de desenvolvimento de software. 34	
3.1.1.1.Etapas do levantamento dos modelos de processos.....	35
3.1.1.2.Identificação das características relacionadas a usabilidade dos modelos	35
3.1.2.Pesquisa de campo	35
3.1.3.Confecção do Questionário da Pesquisa.....	36
3.1.4.Submissão do Questionário da Pesquisa	37
3.1.5.Retorno e tabulação das respostas do formulário de pesquisa	38
3.2.Comentários Parciais.....	38
4.ANÁLISE E RESULTADOS DOS DADOS	39
4.1.Comentários Parciais.....	44
5.CONCLUSÕES E CONSIDERAÇÕES FINAIS	45
5.1.Trabalhos Futuros.....	46
REFERÊNCIAS.....	47
APÊNDICES	56
APÊNDICE A - Questionário usado na coleta de dados da pesquisa de avaliação do sistema	56
PARTE I - APRESENTAÇÃO DA PESQUISA.....	56
PARTE II - INFORMAÇÕES DO ENTREVISTADO	56
PARTE III – SOBRE OS MODELOS DE DESENVOLVIMENTO DE SOFTWARE	57
ANEXOS	59
ANEXO 1 - Plano de Ensino da disciplina de Engenharia de Software	59
.....	64
ANEXO 2 – Artigo Científico	65

1. INTRODUÇÃO

Na velocidade com que a tecnologia da informação se desenvolve e expande, novos serviços e produtos de Software surgem a todo o momento, fazendo com que o grau de complexidade das aplicações aumente.

Exemplos disso são as aplicações web que estão crescendo e se tornando cada vez mais complexas e importantes no dia-a-dia das pessoas. Desta forma, é necessário controlar a maneira como o software é desenvolvido. É preciso utilizar de processos que integrem entre as etapas do desenvolvimento, forneça orientações claras à equipe, direcione tarefas para indivíduos e para a equipe como um todo, especifique quais artefatos devem ser desenvolvidos e ofereça um elenco de critérios para monitoramento e medição das atividades relacionadas ao desenvolvimento do sistema (LÜDERS, 2019).

Além disso, percebe-se a existência de uma grande dificuldade em fazer com que a equipe de desenvolvimento possa trabalhar de forma equilibrada e harmônica, descentralizada, com tarefas bem definidas, respeitando o cronograma de execução, com custos reais e atendendo os requisitos estabelecidos. Diante disso, Silva, Souza e Camargo (2013) comenta que é uma questão de qualidade de vida. Seja desenvolvedor, arquiteto ou gerente de projeto, ao final do dia, todos querem sentir-se bem quanto à construção e distribuição de aplicações bem projetadas, sem cometer erros dispendiosos, trabalhar longas horas e atravessar meses de estresse.

Por isso, adotar e utilizar processos de desenvolvimento maduro e consistente constitui o ponto chave para o sucesso de projetos. Neste sentido, o presente trabalho visa explorar a sua temática: tendência do mercado quanto ao uso dos modelos de desenvolvimento de software: Uma Experiência Tocantinense.

Sendo assim, levantou-se os aspectos teóricos sobre as áreas envolvidas no presente trabalho, tais como, engenharia de software; os modelos de desenvolvimento de software; a atual tendência do mercado quanto ao uso dos modelos de desenvolvimento de software; ementas das disciplinas de engenharia de software.

Nessa lógica, partiu-se para os procedimentos metodológicos, apresentando os materiais e métodos, extração e identificação dos modelos de processo de software e suas características de usabilidade, por fim foi realizada uma pesquisa de campo.

Como instrumento de coleta dos dados, utilizou-se um questionário aplicado via *Web*. confeccionado utilizando-se da plataforma *google form*. O formulário foi submetido ao universo da pesquisa, onde houve um retorno satisfatório das respostas.

Em suma, o método de pesquisa utilizado empregou-se uma pesquisa no enfoque quantitativo. A pesquisa foi realizada com o universo de 70 participante, 61 responderam ao questionário, sendo que estes respondentes atuam em diversas áreas do desenvolvimento de software, onde, 25% são da área da Tecnologia da informação, 23% de bancos de dados, 16% da área de Ciência da computação, 15% de análise de sistemas, 13% da área de engenharia da computação, 5% de Sistema para internet e por fim 3% de Técnico em Informática.

Além dos seis modelos de processos de software, Ementa/Plano de Ensino da Disciplina de Engenharia de Software, que são: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral, a pesquisa apresentou mais sete modelos que são: Iterativo, Baseado em Componentes, Programação Extrema – XP, Processo Unificado da Rational – RUP, Arquitetura Dirigida pelo Modelo – MDA, Desenvolvimento Rápido de Aplicação – RAD e Em V.

A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade. Portanto conclui-se que a ementa do Curso de Licenciatura em Computação do *Campus* Porto Nacional, atende parcialmente a tendência do mercado quanto aos modelos de desenvolvimento de software utilizados.

1.1. Problemática da Pesquisa

O egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do *Campus* Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados?

1.2. Objetivos

Nesta seção estão descritos os objetivos gerais e específicos a serem alcançados através desta pesquisa.

1.2.1. Objetivo Geral

Analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software.

1.2.2. Objetivos Específicos

Para alcançarmos o objetivo anterior, especificamos em:

- a) Identificar os modelos de desenvolvimento de software;
- b) Verificar junto as especialistas tocantinenses as respectivas utilizações dos modelos de processo de desenvolvimento de software;
- c) Comparar a atual tendência do mercado sobre os modelos de desenvolvimento de *software* com a ementa da disciplina de Engenharia de Software do *Campus* Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia.

1.3. Hipótese

A hipótese levantada que solucionará a problemática constitui-se em: A ementa da disciplina de Engenharia de Software do PPP do curso de Licenciatura em Computação do *Campus* Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia atende parcialmente a tendência, contudo se ela se adequar na sua totalidade a atual tendência de mercado quanto ao uso de modelos de processo de Software, formará egressos mais qualificados.

1.4. Justificativa

A relevância deste trabalho está em apresentar uma abordagem multidisciplinar visando analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software uma vez que, se trata de um tema pouco explorado pela parte acadêmica. Nesse sentido, tende a apresentar uma melhoria junto a disciplina de Engenharia de Software do Instituto Federal de Educação, Ciência e Tecnologia *Campus* de Porto Nacional. Com isso o egresso será melhor qualificado para o mercado de trabalho.

Portanto, o presente trabalho ganha ênfase, pois dentro da literatura pesquisada não há outro igual ou que se assemelhe com esse, quanto a sua proposta, tema, problema e objetivos.

1.5. Organização do Trabalho

Para melhor entendimento, o presente trabalho está organizado em cinco capítulos:

No Capítulo 1, é apresentada a contextualização, o tema, a definição do problema, os objetivos a serem alcançados, a hipótese levantada, a justificativa para realização do presente estudo e, por fim, a presente organização do trabalho. No Capítulo 2, é apresentado o referencial teórico, em que são abordados elementos que proporcionam o suporte conceitual necessário para a elaboração da atual pesquisa. Os elementos abordados são: e trabalhos correlacionados. No Capítulo 3, está descrito de forma sistematizada os procedimentos metodológicos, onde estão apresentadas de forma minuciosa, todas as etapas do desenvolvimento deste estudo: são apresentados os materiais e métodos. No Capítulo 4, são abordadas as discussões, análises e resultados da pesquisa. No Capítulo 5, é apresentado as conclusões as quais se chegaram após o presente estudo, bem como as perspectivas e propostas para trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará os aspectos teóricos sobre as áreas envolvidas no presente trabalho. São apresentados conceitos sobre a engenharia de software; os modelos de desenvolvimento de software; a atual tendência do mercado quanto ao uso dos modelos de desenvolvimento de software; ementas das disciplinas de engenharia de software; e trabalhos correlacionados.

2.1. Engenharia de software

A engenharia de software é essencial no desenvolvimento de sistemas ao possibilitar que o analista se aproprie e utilize metodologias assertivas para projetos de software.

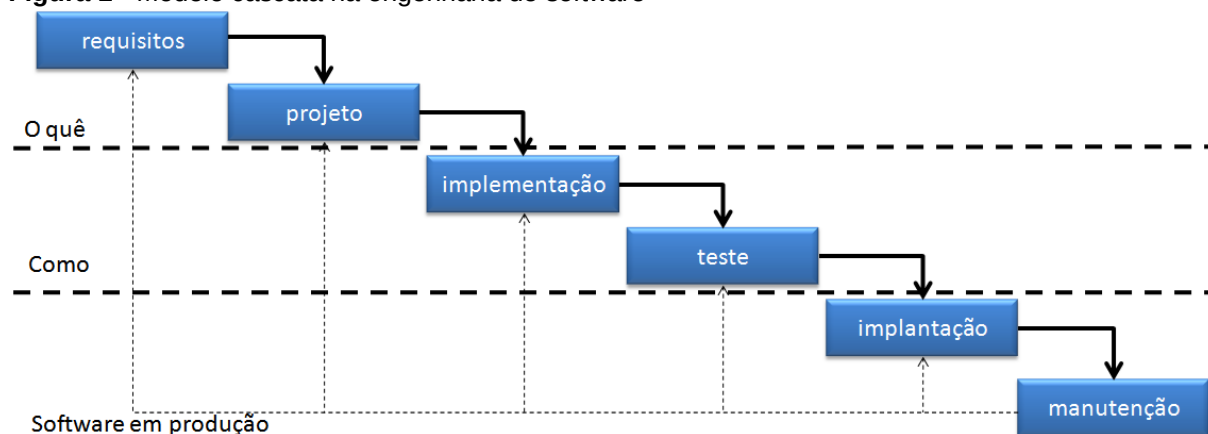
Sommerville (2003) afirma que a engenharia de software é responsável por todos os aspectos da produção de software, iniciando na especificação de sistema até a manutenção desse sistema depois que ele já está em operação. É uma disciplina que tem como meta apresentar técnicas de desenvolvimento de softwares com alta qualidade a um baixo custo.

Na visão de Rezende (2005), a engenharia de software é uma metodologia que desenvolve soluções profissionais utilizando-se dos recursos de *software*, observando os padrões requeridos de qualidade, produtividade e efetividade.

Já para Carvalho e Chiossi (2001), a Engenharia pode ser compreendida como uma disciplina que reúne metodologias, métodos e ferramentas a serem utilizados, desde a percepção do problema até o momento que o sistema desenvolvido deixa de ser operacional, visando resolver problemas inerentes ao processo de desenvolvimento e ao produto de software.

Segundo Pressman (2011), a engenharia de software é formada por processo, métodos (práticas) e ferramentas as quais permitem construir sistemas dentro do prazo e com qualidade.

Em síntese a engenharia de software é, portanto, o conjunto de processos, métodos e ferramentas que podem ser representadas na forma de camadas (Figura 1), comprometida com a qualidade.

Figura 1 - Modelo cascata na engenharia de software

FONTE: Portal /Engenharia de software, 2012.

De acordo com De Melo, Mendonça e Dias (2019), o aperfeiçoamento contínuo de processos é possível obter abordagens cada vez mais efetivas. Na engenharia de software a camada fundamental é a de processos, pois é ela que possibilita o desenvolvimento de software de forma racional e dentro do prazo, mantendo sempre o foco na qualidade

O processo de software é a base para o controle do gerenciamento de projetos, no qual estabelece o contexto onde serão aplicados os métodos técnicos que serão produzidos os produtos derivados do software, tais como: (i) modelos dos documentos; (ii) dados e etc. Também são estabelecidos marcos, a qualidade é garantida e as mudanças são geridas de forma apropriada. Além disso, a metodologia a ser desenvolvida para a entrega efetiva de tecnologia de engenharia de software também é definida na etapa do processo (Pressman, 2011; Sommerville, 2003).

2.2. Os modelos de desenvolvimento de software

Um processo de software corresponde a um conjunto de passos formados por atividades, métodos, práticas e transformações parcialmente ordenados, os quais são usados para desenvolver e manter software e produtos associados (Sommerville, 2003).

Conforme Pressman (2011), processo de software é um conjunto de atividades de trabalho, ações e tarefas realizadas quando algum artefato de software deve ser criado. O processo não é uma prescrição rígida de como desenvolver um software. Ao contrário, é uma abordagem adaptável que possibilita a equipe de

software realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas.

Existem muitos processos de *software* diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de *software* (SOMMEVILLE 2011). Conforme descrito na Tabela 1.

Tabela 1 – Atividades para processo de software

Atividades	Descrição
1. Especificação de software.	A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
2. Projeto e implementação de software.	O software deve ser produzido para atender às especificações.
3. Validação de software.	O software deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução de software.	O software deve evoluir para atender às necessidades de mudança dos clientes.

Fonte: Santos (2018), Apud SOMMERVILLE, 2011.

As atividades, ações e tarefas são alocadas dentro de um modelo ou metodologia o qual determina seu relacionamento com o processo e seu relacionamento entre elas. A metodologia de processo fornece a base para um processo de engenharia de software completo, através de um pequeno número de atividades estruturais e um conjunto de atividades de apoio que são aplicadas a todo o processo. Essas atividades são aplicáveis a todos os projetos de software, independente de complexidade ou tamanho (Pressman, 2011).

O modelo de processo de software é uma representação abstrata do processo. Ele é responsável por descrever de forma simplificada um processo de software, o qual é apresentado a partir de uma perspectiva específica. (Sommerville, 2003).

Nas Tabelas a seguir apresentam um resumo dos modelos de processo de software, referidos por vários autores, como modelo de ciclo de vida, comumente encontrados em bibliografias de autores reconhecidos em Engenharia de Software.

Tabela 2 - Modelos de processos prescritivo ou modelos tradicionais

Nº	Processos de Software	Descrição
01	Cascata	Modelo cascata possui cinco fases principais: i) definição de requisitos; ii) projeto de sistema e software; iii) implementação e teste unitário; iv)

		integração e teste de sistema; e v) operação e manutenção de software.
02	Prototipação	O protótipo é um mecanismo para identificar os requisitos do software. O protótipo pode evoluir para o sistema final.
03	Interativo e Incremental	Combina elementos dos fluxos de processos lineares e paralelos. Aplica sequências lineares de forma escalonada.
04	Espiral	Modelo evolucionário que vincula a natureza iterativa da prototipação com os aspectos sistemáticos e controlados do modelo cascata.
05	Em V	Variação do modelo cascata. Descreve a relação entre as ações de garantia da qualidade e as associadas à comunicação, modelagem e construção.
06	Evolutivo	Envolve a produção de versões iniciais - "protótipos" - de um sistema futuro com o qual pode-se realizar verificações e experimentações para se avaliar algumas de suas qualidades antes que o sistema venha realmente a ser construído.
07	Unificado	Combina uma série de práticas, nesta abordagem, o desenvolvimento é organizado em miniprojetos, de duração fixa chamada de iteração.

Fonte: Elaborado a partir de Pressman (2011), Sommerville (2011) e Paula Filho (2013).

Na tabela 3 a seguir apresentam os modelos de processo de software considerado especializados.

Tabela 3 - Modelos de processos especializado

Nº	Processos de Software	Descrição
01	Desenvolvimento baseado em componentes	O desenvolvimento da aplicação ocorre a partir de componentes de software.
02	Métodos formais	Especificação, desenvolvimento e verificação do software por meio de aplicação de notação matemática.
03	MDA (Arquitetura Dirigida a Modelos)	É uma tecnologia desenvolvida pelo Object Management Group (OMG) que consiste em promover o desenvolvimento de sistemas a níveis mais altos de abstração de tal forma que os modelos desenvolvidos fiquem independentes
04	Praxis (Processo para Aplicativos extensíveis e Interativos)	É voltado para o desenvolvimento de aplicativos gráficos interativos, baseados na tecnologia orientada a objetos.

Fonte: Elaborado a partir de Pressman (2011), Sommerville (2011), Paula Filho (2013) e Pinto, De Mattos e Capeletti (2019), Borges (2003).

Existem diversos modelos ágeis, como: *Adaptive Software Development (ASD)*, *Crystal*, *Dynamic Systems Development Method (DSDM)*, *Extreming Programming*, *Feature Driven Development (FDD)* e *Scrum* (ABRAHAMSSON et al., 2003). Contudo, eles possuem características específicas, conforme apresentadas na Tabela 4.

Tabela 4 - Modelos de processos ágeis

Nº	Processos de Software	Descrição
01	<i>Adaptive Software Development - Desenvolvimento de Software Adaptativo</i>	Foca na busca de objetivos, na participação e na colaboração da equipe, com comunicação constante, garantindo aprendizado durante o projeto (HIGHSMITH, 1996).
02	<i>Dynamic Systems Development Method – Método Dinâmico de Desenvolvimento de Sistemas</i>	Defende o processo de desenvolvimento com interações e ciclos incrementais (STAPLETON, 1997).
03	<i>Extreme Programming - Programação Extrema - PX</i>	Centra o desenvolvimento de projetos em: planejamento, entregas frequentes, descrições metafóricas, projetos simples, testes, refatoração (simplificação de funcionalidade), desenvolvimento em duplas, acessível a todos os membros da equipe, com interações contínuas, presença de cliente no processo e padronização de código (BECK; ANDRES, 2004).
04	<i>Feature-driven Development – Desenvolvimento Dirigido a Funções</i>	Construção de escopo de funcionalidades necessárias para satisfazer o cliente, ordenadas por prioridades. Um plano de desenvolvimento é realizado para cada funcionalidade, gerando uma estrutura de classes associada aos desenvolvedores responsáveis (SILVA et al., 2009).
05	Scrum	Divide o desenvolvimento em iterações, chamadas sprints. Como a equipe é pequena e claramente definida, formada por facilitador e equipe, é também a responsável pelos requisitos (SCHWABER, BEEDLE, 2002).
06	Transformação	A transformação de modelos em cada fase do desenvolvimento permite automatizar o processo de desenvolvimento, evitando erros e bugs. É pensar em 'o que deve ser feito' e não em 'como deve ser feito'. Além de permitir maior qualidade do software, permite que o desenvolvimento seja mais rápido (LAPENDA, LONIEWSKI, MADRUGA, 2008).
07	Desenvolvimento Rápido de Aplicação - RAD	permite uma prototipagem mais rápida e entrega iterativa do produto final. Trata-se de um modelo alternativo ao tradicional modelo cascata que, em geral, foca em um processo de desenvolvimento sequencial e pouco flexível (MAGRI, 2008).

Fonte: Adaptado de Beck e Andres (2004), Cockburn (2004), Highsmith (1996), Silva et al. (2009), Stapleton (1997) e Schwaber e Beedle (2002).

2.3. A atual tendência de mercado quanto ao uso dos modelos de desenvolvimento de software

Desenvolvimento de software é uma tarefa muito complexa, difícil e de alto risco. Segundo Vinicius (2016) descreveu no portal Devmedia, Inúmeros problemas são enfrentados diariamente, entre alguns desses problemas enfrentados, destaca-se: alto consumo de tempo que supera estimativas e cronograma, funcionalidades que não solucionam os problemas dos usuários, baixa qualidade dos sistemas desenvolvidos, tempo para teste e qualidade reduzidos e, em alguns casos, até o cancelamento do projeto por inviabilidade.

Tendo como grande objetivo a redução de risco associado ao desenvolvimento de software, Vinicius (2016) afirma que as metodologias definem uma maneira de trabalhar em projetos utilizando processos, sendo elas um conjunto de atividades e resultados associados que auxiliam no desenvolvimento e construção de softwares.

2.3.1. Metodologia Tradicional x Metodologia Ágil – entenda as diferenças

Na literatura especializada pode-se encontrar duas frentes de processo de software no tocante à execução. A primeira, definida por Pressman (2011) como um processo tradicional, ou também chamada de processo pesado, que se caracteriza por possuir como foco principal a previsibilidade dos requisitos do sistema o comando e o controle destes a partir de um prévio planejamento antes do início do desenvolvimento, transformando estas ações em um processo rigoroso. Como exemplo deste tipo de processo tem-se o CASCATA Schnell (2018), RUP Rocha (2018), e o Catalysis Junior, Silva et al (2019).

Já a outra frente, chamada de processo ágil ou processo leve, possui seu foco na eficiência, abordando como premissa o compromisso entre “nada de processo” e processos rigorosos Beck (2000). Esta frente propõe que a análise de requisitos seja extremamente mutável, “abraçando” mudanças como principal área de atuação da metodologia. Sendo assim, os planejamentos são constantes, não havendo uma etapa exclusiva para isso, ficando o foco principal com a codificação.

Para isso os meios para estes fins são: adaptabilidade; cada item de processo deve agregar valor; orientação a pessoas; comunicação; e aprendizado. Para exemplificar esta outra linha de processos temos o Crystal Santos (2017) e o SCRUM Machado (2018).

Assim, Silva, Souza e Camargo (2013), preconiza que as organizações enfrentam certas dificuldades diante das metodologias disponíveis para o gerenciamento de projetos de software. As empresas de grande porte e entidades governamentais têm como prática o detalhamento de vários processos por relatórios, planilhas e gráficos, sendo comumente utilizada a metodologia tradicional para projeto de software.

Por outro lado, ainda de acordo com os autores supracitados as empresas pequenas e médias encontram dificuldades em utilizar a metodologia tradicional, por ser muito caro a manutenção e controle, e o longo prazo para entrega, por essa razão utilizam se da metodologia ágil. Lutz (2018) garante que a metodologia ágil agrega baixo custo e entregas rápidas, podendo contribuir no desenvolvimento de projeto de software para qualquer tipo de empresa e ser agregada à metodologia tradicional.

Segundo o portal Gaea (2018), não há como definir de modo absoluto que uma metodologia é mais adequada para uma organização do que outra. Antes de sua adoção, devem ser analisados alguns detalhes, como o perfil do projeto a ser desenvolvido, a capacidade do time e a disponibilidade do cliente para a elaboração conjunta de metas.

Já Dos Santos Silva (2013), diz que as duas metodologias podem ser utilizadas quando a empresa entende qual a mais produtiva para os seus projetos. Para isso, são essenciais profissionais capacitados na identificação e no direcionamento dos empreendimentos, conforme a técnica adequada.

Para Machado (2018) a metodologia ágil agrega os valores mais presentes nos modos contemporâneos de trabalho, com foco na inovação, criatividade e colaboração. Embora alguns coordenadores evitem usar a tradicional, Ribeiro, Brezolin e Leonor (2018), ela pode ser muito bem aplicada em casos específicos, nos quais as normas são mais rígidas e a estrutura da empresa cliente não se adéqua à prática ágil.

Ainda em Machado (2018), os segmentos de mercado acompanham as tendências incorporadas aos novos ciclos econômicos, sendo, nesse contexto, necessário ter medidas adaptáveis, rápido aprendizado e adoção de estratégias modernas. A metodologia ágil é, portanto, um processo cada vez mais relevante no gerenciamento de projetos corporativos, portal (GAEA, 2018).

Em consonância, Dos Santos Silva (2013), afirma que a metodologia ágil se difere da tradicional principalmente por ampliar os benefícios para o cliente e trazer um planejamento flexível para os times, integrando a multidisciplinaridade e o constante desenvolvimento.

2.3.2. Mudança para Métodos Ágeis de Desenvolvimento de Software

A mudança para os Métodos Ágeis de desenvolvimento de software teve início em torno da década de 1990, e foi proposta por muitos praticantes de diversas nacionalidades e em perspectivas de projetos diferentes (LEFFINGWELL, 2007).

Em uma edição da Revista Computer, no ano 2000, Boehm (2002) afirmou que era necessário mudar o desenvolvimento de software de um processo lento para um mais rápido e dinâmico. No ano 2001, os métodos ágeis tornaram-se mais conhecidos devido à publicação do Manifesto Ágil (BECK et al., 2001). Nele, houve a união de dezessete especialistas em diversas abordagens de processos de desenvolvimento de software que buscaram estabelecer princípios comuns em práticas ágeis e compartilharam todas essas perspectivas.

O Manifesto Ágil propõe a filosofia de que no desenvolvimento do software os recursos são escassos e que os requisitos são altamente voláteis. Práticas ágeis podem auxiliar o cumprimento do desenvolvimento do software devido às práticas de gerenciamento, de processo e de desenvolvimento de software (LEE; YONG, 2010).

O portal LuizToos afirma que desde 13 de novembro de 2001, quando o Manifesto Ágil foi assinado por dezessetes especialistas da indústria de software da época, que o uso dos métodos ágeis só cresce. Em parte pelo demérito dos métodos tradicionais de desenvolvimento que priorizam muitas vezes a burocracia em detrimento do resultado.

Nessa perspectiva, Duarte (2017), afirma que o Scrum é o processo de software mais utilizado atualmente. Camargo (2018), concorda que metodologia Ágil Scrum é, sem dúvida, o método mais usado nos dias de hoje, principalmente porque

pode ser integrado a outros métodos ágeis com facilidade, aplicando-se não só ao desenvolvimento de softwares como a qualquer ambiente de trabalho.

2.3.3. Processo de software Scrum

Para os seus idealizadores Schwaber & Sutherland (2013) e Sutherland (2014) scrum é um framework estrutural dentro do qual pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível.

Terlizzi e Biancolino (2014) consideram o scrum uma metodologia ágil e que o mesmo se fundamenta no empirismo, o qual define que o conhecimento vem das experiências e tomadas de decisões baseadas no que é conhecido. O Scrum baseia-se no princípio da objetividade, papéis bem definidos e facilidade de aprendizado (SILVA E LOVATO, 2016).

Cada papel dentro do Scrum possui sua necessidade específica de interação e é de suma importância que todo o processo seja transparente o suficiente para que cada pessoa envolvida saiba exatamente o que foi, está ou será feito.

Silva e Lobato (2016) alerta que (Scrum) é um modelo aberto e de forma alguma é previsível, pois o framework não detalha o que deve ser feito. A mesma ideia é defendida por Sabbagh (2013) que conclui que o Scrum entende que as práticas necessárias para o sucesso do projeto são muito específicas para cada contexto, não podendo assim ser prescritas.

Na tabela 5 a seguir são apresentadas nove vantagens em usar o processo Scrum como modelo de desenvolvimento de Software, que segundo Duarte (2017), são os motivos principais e a tendência da atualidade em se utilizar essa metodologia.

Tabela 5 – Vantagens uso do Scrum no desenvolvimento de Software

ORDEM	VANTAGEM	DESCRIÇÃO
01	Adaptabilidade	O controle de processos empíricos e a entrega iterativa fazem com que os projetos sejam adaptáveis e abertos à incorporação de mudanças.
02	Transparência	Todas as fontes de informações, tais como, o quadro de tarefas (Kanban) e o gráfico Burndown, são compartilhadas gerando um ambiente de trabalho aberto.
03	Feedback Contínuo	O feedback contínuo é fornecido através de processos denominados como a Reunião Diária e a Sprint Review.

04	Melhoria Contínua	As entregas melhoram progressivamente, Sprint por Sprint, através do processo de refinamento do Backlog do Produto e do processo em si.
05	Entrega Contínua de Valor	Os processos iterativos permitem a entrega contínua de valor tão frequente quanto exigido pelo cliente.
06	Eficiência	O Time-boxing e a minimização de trabalho não-essencial conduzem a níveis mais altos de eficiência.
07	Motivação	Os processos de conduzir a Reunião Diária e de Retrospectiva da Sprint conduzem a níveis mais altos de motivação entre os colaboradores.
08	Alta Velocidade	Uma estrutura de colaboração permite que os times multifuncionais atinjam o seu pleno potencial e alta velocidade.
09	Ambiente Inovador	Os processos de Retrospectiva da Sprint e de Review da Sprint criam um ambiente de introspecção, aprendizagem e adaptabilidade, que levam a um ambiente de trabalho inovador e criativo.

Fonte: Duarte (2017), adaptada pela autora.

2.4. Ementa da disciplina de engenharia de software

Em primeiro lugar, antes de falar sobre a Ementa da disciplina de engenharia de software, é necessário, para melhor entendimento, falar um pouco sobre o Projeto Pedagógico do Curso de graduação de Licenciatura em Computação ofertado pelo *Campus* Porto Nacional, do Instituto Federal de Ciência e Tecnologia do Tocantins – IFTO.

Há diferentes maneiras de denominar o referido documento, tais como projeto político pedagógico, proposta pedagógica, projeto educativo, dentre outros. Essa opção se justifica quando há o entendimento de que a palavra projeto indica “[...] um documento que contém um conjunto de decisões, procedimentos, instrumentos e ações articulados na direção da superação de problemas, numa realidade específica” (FUSARI, 1995, p. 102-103).

Veiga (2000, p. 12), esclarece sobre a definição dessas terminologias, que:

A proposta pedagógica ou projeto pedagógico relaciona-se à organização do trabalho pedagógico da escola; o plano de trabalho está ligado à organização da sala de aula e a outras atividades pedagógicas e administrativas. Isso significa que o plano de trabalho é o detalhamento da proposta ou projeto (VEIGA, 2000, p. 12).

Fusari (1995) e Veiga (1998) assumem a expressão projeto político-pedagógico para definir o documento e explicam que os adjetivos político e pedagógico a ele acrescentados garantem de um lado a não neutralidade dessa

documentação e de outro o compromisso com o pedagógico, isto é, com a pedagogia como ciência da e para a educação. Em outras palavras, ciência que orienta a prática profissional dos docentes" (FUSARI, 1995, p. 102).

É político no sentido de compromisso com a formação do cidadão para um tipo de sociedade. [...] Pedagógico, no sentido de definir as ações educativas e as características necessárias às escolas de cumprirem seus propósitos e sua intencionalidade (VEIGA, 1998, p. 13).

Portanto, o Projeto Pedagógico do Curso de graduação de Licenciatura em Computação ofertado pelo *Campus* Porto Nacional, do Instituto Federal de Ciência e Tecnologia do Tocantins – IFTO está amparado legalmente nos seguintes marcos: Lei de Diretrizes e Bases da Educação Nacional – LDB n.º 9.394/96; Lei n.º 13.168, de 6 de outubro de 2015, que altera a redação do § 1º do art. 47 da Lei n.º 9.394/1996; Resolução CNE/CP n.º 2, de 1 de julho de 2015, que define as Diretrizes Curriculares Nacionais para a formação inicial em nível superior (cursos de licenciatura, cursos de formação pedagógica para graduados e cursos de segunda licenciatura) e para a formação continuada, entre outros (IFTO, 2018).

Então, o curso de Licenciatura em Computação ofertado pelo IFTO *Campus* Porto Nacional, descrito no PPC, vigente a partir de 2018, está estruturado em 53 componentes curriculares que somados às 200 horas de atividades complementares totalizam 3280,2 horas; entre estes componentes curriculares tem-se três estágios supervisionados obrigatórios e um componente relacionado ao TCC - Trabalho de Conclusão de Curso.

Assim, o curso possui horas-aula de 50 minutos a serem regidas no período noturno, ao longo de oito semestres, podendo ser integralizado em até 16 semestres. Em cada semestre 40 vagas serão oportunizadas e, após concluído o curso, o estudante receberá o diploma de Licenciado(a) em Computação (IFTO, 2018).

2.4.1. Ementa da disciplina de Engenharia de Software

A ementa é como um resumo ou sinopse de determinado tema ou área. Quando se fala em “ementa do curso”, significa a apresentação de um texto que evidencia as principais características deste curso (PORTAL SIGNIFICADOS, 2016).

Da mesma forma, a “ementa disciplinar” ou “ementa escolar” aponta os postos-chaves da matéria a ser apresentada ou das disciplinas que serão ministradas durante o ano letivo, continua afirmando o Portal Significados.

A ementa da disciplina de Engenharia de Software se encontra disponível no Anexo I do Projeto Pedagógico do Curso de graduação de Licenciatura em Computação ofertado pelo *Campus* Porto Nacional, do IFTO. Conforme apresentado na tabela 6 a seguir.

Tabela 6 - Componente Curricular: Engenharia de Software

Componente Curricular: Engenharia de Software	
Carga Horária (CH) total: 66,7 CH teórica: 33,3 CH prática: 33,4	
Pré-requisitos: Modelagem e Análise de Sistemas.	
Ementa	
Visão geral e princípios fundamentais da Engenharia de Software. Conhecimentos básicos sobre ciclo de vida de software e seus estágios: requisitos, projeto, implementação, gerenciamento, qualidade. Emprego de metodologias e ferramentas para análise e projeto de sistemas. Documentação de software. Engenharia Reversa. Reengenharia. Ferramentas CASE: conceitos, tipos e exemplos associando com as etapas do ciclo de vida de software.	
Competências	
• Apresentar, analisar e discutir o corpo de conhecimento que constitui a Engenharia de Software, seus princípios, métodos e ferramentas. • Identificar as técnicas da Engenharia de Software para o desenvolvimento de um produto.	
Habilidades	
• Ser capaz de descrever os métodos, ferramentas e procedimentos associados; • Conhecer e utilizar as técnicas de trabalho em grupo, especificamente para desenvolvimento de softwares; • Saber identificar os princípios da ética profissional do engenheiro de software.	
Bibliografia básica	
PRESSMAN, R. S. Engenharia de Software . 5ª ed. São Paulo: McGraw-Hill, 2007. PAULA FILHO, W. P. Engenharia de software: fundamentos, métodos e padrões . Rio de Janeiro: LTC, 2003. LTC, 2009.	
Bibliografia complementar	
PRESSMAN, R. Software Engineering: A Practitioner's Approach . EUA: G- Hill, 2005. SOMMERVILLE, I. Engenharia de Software . São Paulo: Pearson. 2007. DAVIS, A. M. Software requirements . EUA: Prentice Hall, 1993. PFLEEGER, S. L. Engenharia de Software Teoria e Prática . São Paulo: Prentice Hall, 2004. MAGELA, R. Engenharia de softwares aplicada . Rio de Janeiro: Alta Books, 2006.	

Fonte: IFTO (2018).

2.4.2. Plano de ensino da disciplina de Engenharia de Software

O Plano de Ensino é um plano de ação; é o registro do planejamento das ações pedagógicas para o componente curricular durante o período letivo. É um instrumento didático-pedagógico e administrativo de elaboração e uso obrigatórios. Pode-se pensar, primeiramente, que:

[...] planejamento é processo de busca de equilíbrio entre meios e fins, entre recursos e objetivos, visando ao melhor funcionamento de empresas, instituições, setores de trabalho, organizações grupais e outras atividades humanas. O ato de planejar é sempre processo de reflexão, de tomada de decisão sobre a ação; processo de previsão de necessidades (...) visando à concretização de objetivos, em prazos determinados e etapas definidas, a partir dos resultados das avaliações (PADILHA, 2001).

Sendo plano, trata-se de um “documento utilizado para o registro de decisões do tipo: o que se pensa fazer, como fazer, quando fazer, com que fazer, com quem fazer. Para existir plano é necessária a discussão sobre fins e objetivos, culminando com a definição dos mesmos” (PADILHA, 2001).

Dentro dessas definições, o Plano de ensino da disciplina de Engenharia de Software descreve no seu tópico 4 – Bases Tecnológicas, os modelos e processos de software utilizados/ministrados nas aulas da disciplina de Engenharia de Software IFTO (2018), conforme descrito a seguir: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral. Por sua vez, o referido Plano de Ensino da disciplina de Engenharia de Software, segue como anexo 1 deste trabalho

2.5. Trabalhos correlacionados

Esta seção contém uma síntese dos trabalhos correlatos dessa pesquisa identificados durante a revisão bibliográfica para embasamento desta pesquisa.

2.5.1. Caracterização de um Processo de Software para Projetos de Software Livre

Reis (2003), analisa projetos do ponto de vista de seu processo de software; em outras palavras, analisa as atividades que realizam para produzir, gerenciar e garantir a qualidade do seu software. Na parte inicial do trabalho é feita uma extensa revisão bibliográfica, comentando os principais trabalhos na área, e são detalhadas as características principais dos projetos de software livre. O conteúdo principal deste trabalho resulta de dois anos de participação ativa na comunidade, e

de um levantamento realizado através de questionário, detalhando mais de quinhentos projetos diferentes. São apresenta das treze hipóteses experimentais, e os resultados do questionário são discutidos no contexto destas hipóteses.

2.5.2. Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software

Salviano (2016) pesquisou sobre a melhoria de processo com a utilização da flexibilidade da arquitetura contínua da ISO/IEC 15504 para um melhor alinhamento da melhoria com o contexto e objetivos estratégicos específicos de cada organização, principalmente as micro e pequenas. São identificadas três gerações de arquiteturas de modelos de capacidade de processo (estagiada fixa, contínua fechada e contínua aberta). É proposta uma engenharia de processo, de software e de qualquer outro trabalho humano intensivo em conhecimento, dirigida por perfis de capacidade de processo.

2.5.3. Estação TABA: Uma infra-estrutura para Implantação do Modelo de referência para Melhoria de Processo de Software

Rocha (2005) escreveu um artigo que descreve o apoio para a implantação do modelo de Referência para melhoria de processo de software Brasileiro (MR mps Br) através da Estação TABA, um ambiente de desenvolvimento de software orientado a organização. Também são apresentados resultados práticos e lições aprendidas no contexto de uma experiência piloto da implantação do MR mps BR em Pequenas e Médias Empresas Brasileiras.

2.6. Comentários Parciais

Este capítulo discutiu informações sobre elementos que proporcionaram o suporte conceitual necessário para a elaboração do trabalho, os elementos abordados foram: conceitos sobre a engenharia de software; os modelos de desenvolvimento de software; a atual tendência do mercado quanto ao uso dos modelos de desenvolvimento de software; ementas das disciplinas de engenharia de software.

O estudo realizado nesse capítulo deu embasamento teórico para o desenvolvimento do trabalho apresentado metodologicamente no capítulo seguinte. No final do estudo foi realizada uma busca por trabalhos correlacionados.

3. PROCEDIMENTOS METODOLÓGICOS

Este capítulo apresenta a metodologia científica adotada neste trabalho, de forma a sistematizar a revisão bibliográfica para descrever sobre a Tendência do mercado quanto ao uso dos modelos de desenvolvimento de software, seguida como foi realizada a pesquisa, confecção e aplicação do questionário.

3.1. Materiais e Métodos

Este tópico verifica a problemática da proposta desta pesquisa, que consiste em saber se o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do *Campus* Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados?

3.1.1. Levantamento dos modelos de processo de desenvolvimento de software

Para tanto, fez-se um levantamento teórico, realizado por meio de uma pesquisa bibliográfica, uma vez que a pesquisadora busca embasamento na literatura existente. A pesquisa bibliográfica foi realizada a partir do levantamento de referências teóricas já analisadas, e publicadas por meios escritos e eletrônicos, como livros, artigos científicos, páginas de web sites.

Na opinião de Fonseca, (2002, p. 32). “Qualquer trabalho científico inicia-se com uma pesquisa bibliográfica, que permite ao pesquisador conhecer o que já se estudou sobre o assunto”. Existem, porém, pesquisas científicas que se baseiam unicamente na pesquisa bibliográfica, procurando referências teóricas publicadas com o objetivo de recolher informações ou conhecimentos prévios sobre o problema a respeito do qual se procura a resposta.

Gil (2007), afirma que os exemplos mais característicos desse tipo de pesquisa são sobre investigações sobre ideologias ou aquelas que se propõem à análise das diversas posições acerca de um problema.

Em Salomon (2004), esclarece que a pesquisa bibliográfica se fundamenta em conhecimentos proporcionados pela Biblioteconomia e Documentação, entre outras ciências e técnicas empregadas de forma metódica envolvendo a identificação, localização e obtenção da informação, fichamento e redação do trabalho científico.

Esse processo solicita uma busca planejada de informações bibliográficas para elaborar e documentar um trabalho de pesquisa científica.

3.1.1.1. Etapas do levantamento dos modelos de processos

1º Etapa: A pesquisa deu-se início a partir de um problema diagnosticado: o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do *Campus* Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados?

2º Etapa: Diante do problema diagnosticado foi feito os seguintes termos de busca: engenharia + software; os modelos + desenvolvimento + software; tendência + mercado + modelos + desenvolvimento + software; ementas + disciplinas + engenharia de software

3º Etapa: Depois de elaborar os termos de busca fez-se uma pesquisa nas bases de dados afim de extrair modelos de desenvolvimento de software em artigos, livros e revistas.

4º Etapa: Após as pesquisas realizadas nas bases de dados fez-se uma seleção dos trabalhos mais relevantes para o decorrer da pesquisa.

5º Etapa: Nesta fase realizou-se a tradução dos trabalhos que se encontravam na língua inglesa.

6º Etapa: Após a tradução dos artigos, livros e revistas foram extraídos os modelos de desenvolvimento de software, conforme descritos nas tabelas 2, 3 e 4, no capítulo 2, item 2.2 deste trabalho.

3.1.1.2. Identificação das características relacionadas a usabilidade dos modelos

Para a Identificação das características relacionadas a usabilidade dos modelos de processo de desenvolvimento de Software, procedeu-se de igual modo do item 3.1.1.1, logo após a extração dos modelos extraiu-se também as suas características de usabilidades.

3.1.2. Pesquisa de campo

Para a consecução do presente trabalho foi adotado uma classificação de pesquisa de campo, conforme Fonseca (2002), pesquisa de campo caracteriza-se

pelas investigações em que, além da pesquisa bibliográfica e/ou documental, se realiza coleta de dados junto a pessoas, com o recurso de diferentes tipos de pesquisa (pesquisa *ex-post-facto*, pesquisa-ação, pesquisa participante, etc.)

Desta forma, fez-se o uso dos métodos estatísticos, tendo em vista a necessidade de tornar quantificável algumas informações do questionário.

A escolha do universo da pesquisa em que o estudo foi desenvolvido está definido com 61 participantes, deste total participaram dessa pesquisa diversos profissionais da área de desenvolvimento de software conforme discriminados a seguir: 15 da Tecnologia da informação, 14 de Banco de Dados, 8 de Engenharia da computação, 10 de Ciências da computação, 9 de Análise de sistema, 2 de Técnico em informática e 3 de Sistemas para internet.

O instrumento de coleta de dados escolhido foi a aplicação de um questionário confeccionado para ser aplicado via web aos entrevistados, como descrito no subitem 3.1.3 a seguir.

3.1.3. Confeção do Questionário da Pesquisa.

O questionário é um instrumento ou programa de coleta de dados que segundo Carnevali e Miguel, (2001), pode ser enviado via correio, fax, Internet, etc., o questionário pode ser estruturado não disfarçado: o respondente sabe qual é o objetivo da pesquisa, e o questionário é padronizado, usando principalmente questões fechadas.

A confecção é feita pelo pesquisador; o preenchimento é realizado pelo informante. A linguagem utilizada no questionário deve ser simples e direta para que o interrogado compreenda com clareza o que está sendo perguntado (KAUARK, MANHÃES E MEDEIROS, 2010).

O questionário foi confeccionado utilizando a plataforma do google, formulário *Google Form* e utilizado como instrumento de coleta de dados, os links foram enviados via e-mail, seguido de uma nota explicativa, onde, informava a natureza da pesquisa e de cada instrumento aplicado. Pois como afirma Marconi e Lakatos (2003, p. 201):

Junto com o questionário deve-se enviar uma nota ou carta explicando a natureza da pesquisa, sua importância e a necessidade de obter respostas, tentando despertar o interesse do recebedor, no sentido de que ele preencha e devolva o questionário dentro de um prazo razoável.

Contudo, explicar os objetivos da pesquisa na nota, solicitando colaboração aumenta a influência no retorno dos questionários, pois conforme as autoras anteriormente citadas, deve se despertar o interesse do pesquisado para que ele preencha e devolva-o do mesmo modo que foi enviado.

Nesse sentido, o questionário foi definido em reunião junto ao professor orientador que norteou a confecção do mesmo, assim todas as edições do questionário foi acompanhada pelo orientador.

A edição do questionário foi realizada em alguns passos: o primeiro foi pesquisa sobre como confecciona-los, o segundo passo foi a criação dos escopos para tomada de base na edição, o terceiro passo foi as correções dos escopos, o quarto passo foi a confecção do questionário no Google Docs, ferramenta gratuita para criação de formulários, sendo fundamental para a realização da pesquisa por meio da internet.

A confecção do questionário da pesquisa se deu com base na necessidade de verificar a tendência do mercado quanto ao uso dos modelos de desenvolvimento de software no âmbito do estado do Tocantins, assim a sua elaboração se deu com base no método avaliador baseado em Carnevalli e Miguel (2001), "...o questionário pode ser estruturado não disfarçado: o respondente sabe qual é o objetivo da pesquisa," Tal método/escala garante uma maior precisão na análise e tabulação dos dados obtidos. O questionário de pesquisa construído nesta etapa está disposto no apêndice A.

3.1.4. Submissão do Questionário da Pesquisa

Após a confecção foi realizada a submissão do questionário de pesquisa, com o objetivo de coletar os dados para verificação da tendência do mercado quanto ao uso dos modelos de desenvolvimento de software no âmbito do estado do Tocantins, o questionário de pesquisa foi submetido por meio de um formulário Google Docs que permite aos participantes responderem via internet, outra vantagem deste recurso é enviar o link do formulário via e-mail, possibilitando que o mesmo seja respondido com tempo pelo participante e de qualquer lugar que disponha de internet.

3.1.5. Retorno e tabulação das respostas do formulário de pesquisa

Como explicado no item anterior 3.1.4, foi enviado um link do formulário de pesquisa via e-mail, com uma nota explicativa, isso possibilitou o retorno das respostas em tempo que o questionário era respondido pelo universo da pesquisa.

Com os dados brutos em mãos, o processamento dessas informações se deu das seguintes formas: Em primeiro momento as coletas dos dados foram feitas em planilhas eletrônicas do pacote *Google Docs*, no segundo momento os dados foram salvos em planilhas do Microsoft Excel para facilitar a análise e tabulação dos dados, em terceiro momento foi aplicado os métodos estatísticos para a tabulação dos dados. E por fim, os resultados foram impressos em gráficos e tabelas geradas pela ferramenta utilizada, para serem apresentados no capítulo 4 Análises e Resultados dos Dados.

3.2. Comentários Parciais

Neste capítulo foram apresentados os procedimentos metodológicos traçados desde a criação dos formulários até sua aplicação e os métodos utilizados para coleta e análise dos dados utilizados nesta pesquisa. O mesmo foi dividido em duas etapas, a primeira sendo a definição dos procedimentos metodológicos da pesquisa e a segunda sendo a confecção do método de coleta de dados.

Desta forma, foram apresentados na primeira etapa os embasamentos teóricos para realização desta pesquisa, foi realizada a definição do universo da pesquisa, o método a ser utilizado na pesquisa e a escolha da forma da aplicação do questionário de pesquisa.

Na segunda etapa, foi selecionado o instrumento de coleta de dados, seguindo com a sua confecção que no primeiro momento foi elaborado um escopo do questionário da pesquisa, no segundo momento foi confeccionado o questionário da pesquisa, que só então foi aplicado para a coleta dos dados.

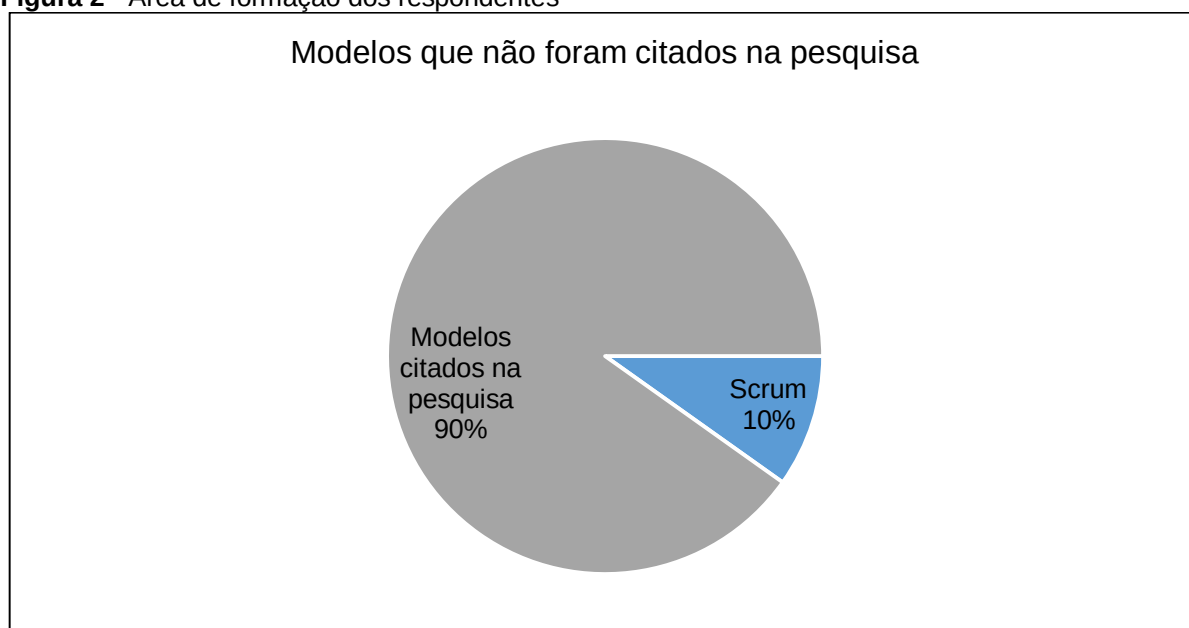
Todas as atividades realizadas neste capítulo foram importantes para alcançar os objetivos desta pesquisa que é analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software.

4. ANÁLISE E RESULTADOS DOS DADOS

Neste Capítulo são apresentados os resultados e análises obtidas com a pesquisa, Tendência do mercado quanto ao uso dos modelos de desenvolvimento de software: Uma experiência tocantinense, foram levantados os seguintes questionamentos: 1). A escolha do modelo deve levar em conta fatores como: cultura da empresa, nível de maturidade dos processos, natureza da aplicação e do projeto, entre outros. Dentre os modelos de desenvolvimento de software abaixo quais você mais utiliza? 2). Em relação aos modelos correlacionados abaixo responda quais características levam a utilizá-lo; 3). Caso existam características que não foram listadas na pergunta anterior cite-as identificando o modelo; 4). Você conhece algum modelo que não foi citado na pergunta 1? Se sim, qual (is)? 5). Por favor, informar qualquer outra consideração que julgue ser relevante. Além de informações pessoais e área de atuação dos respondentes.

Do universo da pesquisa de 70 pessoas, 61 responderam ao questionário, sendo que estes respondentes atuam em diversas áreas do desenvolvimento de software, conforme descrito na figura 2, onde 25% são da área da Tecnologia da informação, 23% de bancos de dados, 16% da área de Ciência da computação, 15% de análise de sistemas, 13% da área de engenharia da computação, 5% de Sistema para internet e por fim 3% de Técnico em Informática.

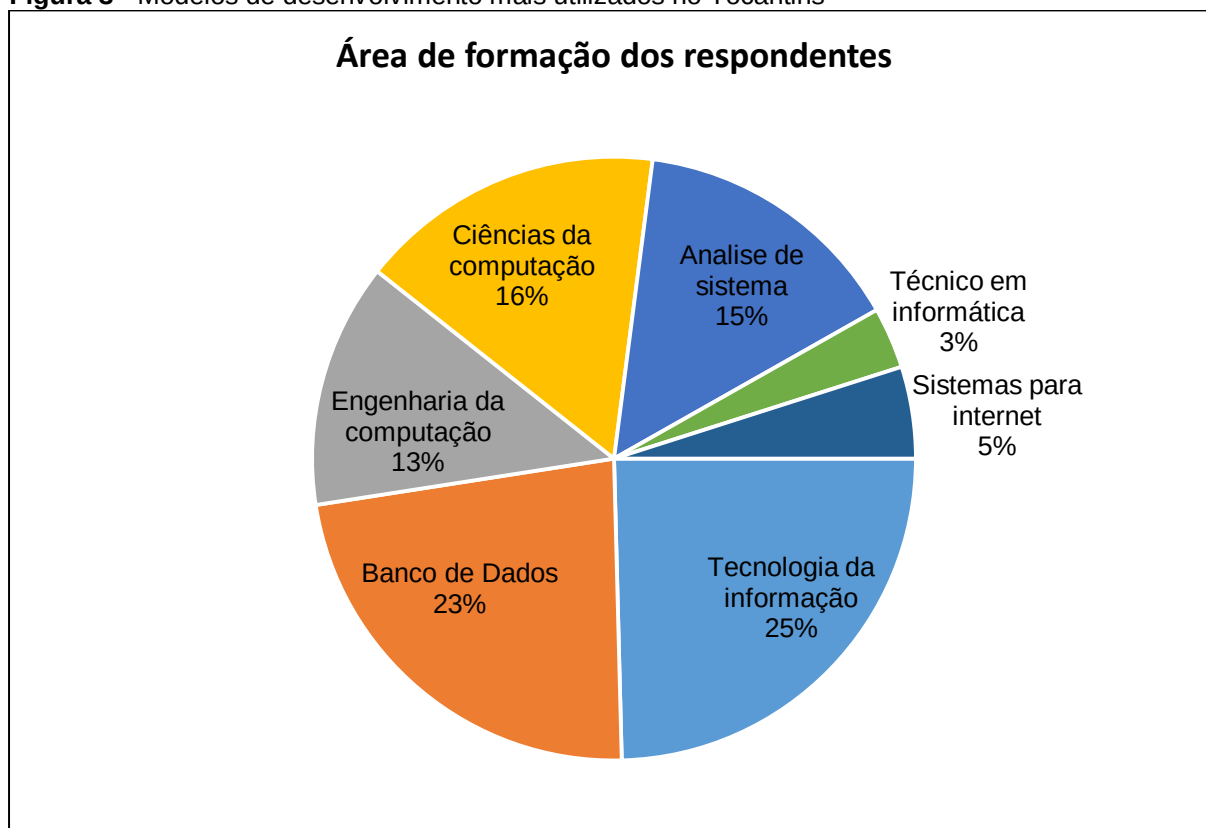
Figura 2 - Área de formação dos respondentes



Fonte: A autora (2019).

Quando questionados quanto a escolha do modelo que deve levar em conta fatores como: cultura da empresa, nível de maturidade dos processos, natureza da aplicação e do projeto, entre outros. Dentre os modelos de desenvolvimento de software abaixo quais você mais utiliza? 18 deles responderam que utilizam o Modelo Incremental, 17 respondentes utilizam o modelo Iterativo, 8 utilizam o modelo Cascata, 6 utilizam o modelo Espiral 4 utilizam Prototipação, 4 utilizam Transformação, 4 Processo Unificado da Rational – RUP, 4 Arquitetura Dirigida pelo Modelo – MDA, 3 Baseado em Componentes, 3 Programação Extrema – XP, 3 Praxis, 2 Evolutivo, 2 Em V, conforme apresentado na figura 3.

Figura 3 - Modelos de desenvolvimento mais utilizados no Tocantins



Fonte: A autora (2019).

Ao analisar o gráfico da figura 3, pode-se perceber que a soma dos números das respostas contabiliza 78, enquanto que o número de participante são de apenas de 61, isso ocorre devido a questão 1 do formulário da pesquisa ser de múltiplas escolhas, ou seja o participante poderia escolher mais de uma opção, portanto a pesquisa mostrou que vários participantes utilizam mais de um modelo de processo de desenvolvimento de software.

Isso só confirma o que a literatura apresenta no item 2.3 do Referencial Teórico onde, Santos Silva (2013) diz: “...as duas metodologias podem ser utilizadas quando a empresa entende qual a mais produtiva para os seus projetos”.

Na tabela 6 a seguir, estão representadas as características que levaram aos respondentes a utilizarem cada modelo de software descrito, isso em resposta a questão número 2 do formulário da pesquisa que quis saber: Em relação aos modelos correlacionados abaixo responda quais características levam a utiliza-lo?

Tabela 7 – Características que levaram a utilizarem cada modelo de software

Total	Modelo	Características
8	Cascata	1. As fases são bem definidas; 2. Permite o planejamento das fases seguintes; 3. As etapas são divididas; 4. Permite controle departamental e gerencial; 5. As fases seguem sem sobreposições.
4	Prototipação	1. A construção de um sistema menor é sempre menos arriscada que a construção de um grande; 2. Requer pequeno investimento inicial.
2	Evolutivo	1. Flexibilidade; 2. Sem estimativa de custo e prazos.
18	Incremental	1. As fases são bem definidas; 2. As etapas são divididas.
4	Transformação	1. A construção de um sistema menor é sempre menos arriscada que a construção de um grande; 2. Cobre vasta gama de atividades; 3. Permite o planejamento das fases seguintes.
2	Espiral	1. Permite o planejamento das fases seguintes. 2. Possui interfaces explícitas.
17	Iterativo	1. Alta capacidade de reutilização; 2. Os requisitos e as táticas de alteração são acomodados.
3	Baseado em Componentes	1. Possui interfaces explícitas; 2. Permite o planejamento das fases seguintes.
3	Programação Extrema - XP	1. Cobre vasta gama de atividades; 2. Permite encontrar defeitos de forma antecipada; 3. Aplicado a projetos de alto risco; 4. Conjunto bem definido de regras.
4	Processo Unificado da Rational – RUP	1. Cobre vasta gama de atividades; 2. Permite encontrar defeitos de forma antecipada.
4	Arquitetura Dirigida pelo Modelo – MDA	1. Permite o planejamento das fases seguintes; 2. Possui interfaces explícitas.
0	Desenvolvimento Rápido de Aplicação – RAD	--
3	Praxis	1. Possui interfaces explícitas.
2	Em V	1. Permite o planejamento das fases seguintes. 2. Possui interfaces explícitas.

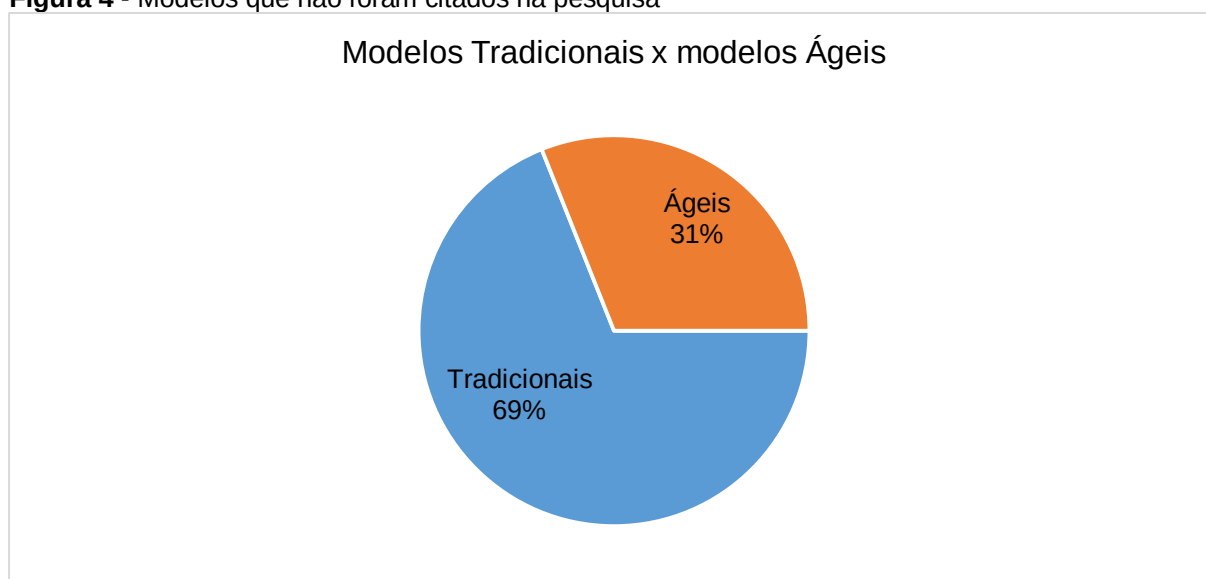
Fonte: A autora (2019).

O questionamento 3 da pesquisa foi saber dos respondentes se existiam características que não foram listadas na pergunta 2 caso existissem citá-las e

identificar o modelo, como resultado, 10% do universo da pesquisa informaram que existe o modelo denominado SCRUM e suas características de usabilidade são: Adaptabilidade, Transparência, Feedback Contínuo, Melhoria Contínua, Entrega Contínua de Valor, Eficiência, Motivação, Alta Velocidade, Ambiente Inovador

Quando indagados na pergunta 4, que se conheciam algum modelo que não foi citado na pergunta 1? Se sim, qual (is)? 10% dos respondentes informaram que conhecem o modelo Scrum conforme apresentado na figura 4.

Figura 4 - Modelos que não foram citados na pesquisa

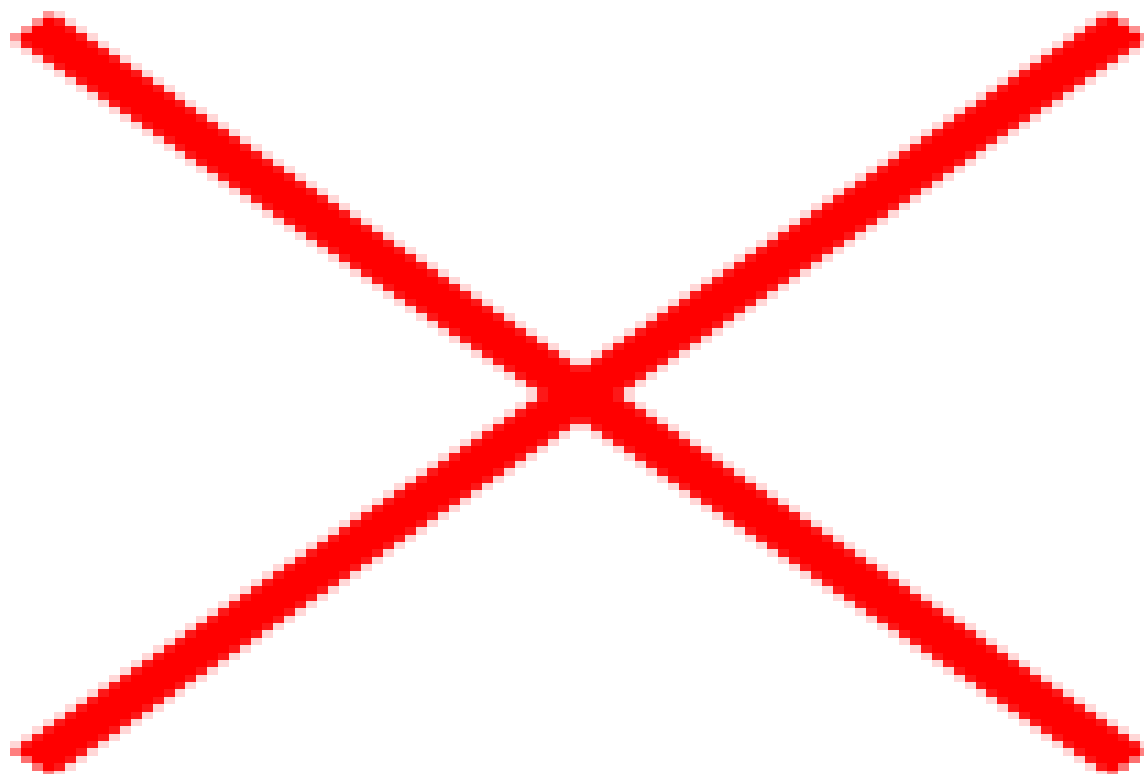


Fonte: A autora (2019).

Ainda foram questionados na pergunta 5 desta pesquisa, para informarem qualquer outra consideração que julgasse ser relevante, porém ninguém se manifestou nessa questão.

A fim de comparar a atual tendência do mercado, sobre os modelos de desenvolvimento de software com a disciplina de Engenharia de Software do *Campus* Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia, dividiu-se os modelos em dois grupos, de acordo com Pressman (2011), Ágeis e Tradicionais. Conforme apresentado no gráfico da figura 5, observa-se que 69% dos respondentes utilizam modelos tradicionais, enquanto 31% utilizam modelos ágeis.

Figura 5 - Modelos Tradicionais x modelos Ágeis

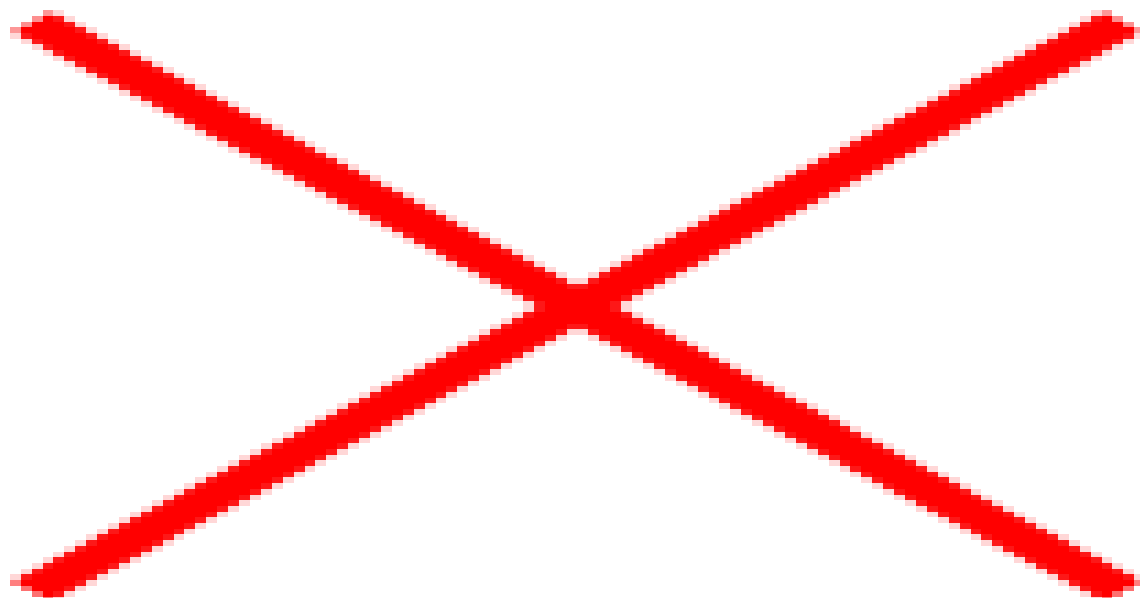


Fonte: A autora (2019).

Apesar da literatura pesquisada mostrar um crescimento das metodologias ágeis, principalmente através do método Scrum, mesmo assim, os modelos tradicionais ainda são usados pela maioria dos respondentes desta pesquisa.

Ainda, a fim de uma melhor precisão nos resultados dessa pesquisa separou-se os modelos descritos no Plano de Ensino dos modelos pesquisados e analisando o gráfico da figura 6 a seguir, comparando-o com o gráfico da figura 3, percebe-se que a ementa não contempla todos os modelos pesquisados e que apenas 1 dos modelos (Transformação), pertence aos grupos de desenvolvimentos Ágeis.

Figura 6 - Modelos descrito na ementa / plano de ensino



Fonte: A autora (2019).

Observa-se, portanto que a ementa do curso de Licenciatura em computação do IFTO de Porto Nacional – TO. atende parcialmente a tendência quanto ao uso dos modelos de processos de desenvolvimento de Software.

4.1. Comentários Parciais

Neste capítulo está sendo apresentados os resultados obtidos da pesquisa que visou observar a tendência do mercado quanto ao uso dos modelos de desenvolvimento de software nos limites do estado do Tocantins.

Os dados obtidos com a pesquisa permitiram avaliar que a ementa da disciplina de Engenharia de Software do PPP do curso de Licenciatura em Computação do *Campus* de Porto Nacional do IFTO. atende parcialmente a tendência quanto ao uso dos modelos de processos de desenvolvimento de Software.

5. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Este trabalho propôs verificar se o egresso da disciplina de Engenharia de Software do curso de Licenciatura em Computação do *Campus* Porto Nacional do IFTO Porto Nacional - TO. está atualizado, conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados.

Portanto, a problemática da pesquisa foi respondida, a hipótese levantada foi validada, pois, a pesquisa mostrou que a ementa da disciplina de Engenharia de Software do PPP do curso de Licenciatura em Computação do *Campus* Porto Nacional do IFTO atende parcialmente a tendência quanto ao uso dos modelos de processos de desenvolvimento de Software.

Ainda, todos os objetivos deste trabalho foram alcançados pois identificou-se os modelos de desenvolvimento de software; verificou junto as especialistas tocantinenses as utilizações dos modelos de processo de desenvolvimento de software e ainda Comparou a atual tendência do mercado sobre os modelos de desenvolvimento de *software* com a disciplina de Engenharia de Software do *Campus* Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia.

Nesse sentido, a contribuição desta pesquisa fica evidente, pois, a atual ementa do PPP vigente do curso de Licenciatura em Computação do *Campus* Porto Nacional do IFTO, apresenta de forma geral a utilização dos modelos de desenvolvimento de software, descrito no subtópico 2.4.2 do Capítulo – 2 e no anexo – 1, Plano de Ensino da Disciplina de Engenharia de Software, que são: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral.

Além desses descrito acima, a pesquisa apresentou mais sete modelos que são: Iterativo, Baseado em Componentes, Programação Extrema – XP, Processo Unificado da Rational – RUP, Arquitetura Dirigida pelo Modelo – MDA, Desenvolvimento Rápido de Aplicação – RAD e Em V. A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade.

Portanto, a contribuição dessa pesquisa é sugerir a atualização da ementa e plano de ensino, acrescentando os modelos que não são contemplados na disciplina, principalmente o modelo Scrum.

Todavia a temática desse estudo se torna relevante pois, apresentou uma abordagem multidisciplinar ao analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software, uma vez que, trata-se de um tema pouco explorado pelo universo acadêmico e apresentou uma melhoria junto a disciplina de Engenharia de Software do *Campus* porto Nacional do IFTO. Com isso, o egresso será melhor qualificado para o mercado de trabalho.

Assim, o presente trabalho ganha ênfase, pois dentro da literatura pesquisada não há outro igual ou que se assemelhe com esse, quanto a sua proposta, tema, problema e objetivos.

Por fim, este trabalho como um todo propiciou grande experiência a autora, a principal razão é o envolvimento e foco em todas as etapas de desenvolvimento e escrita do projeto. Com o foco no desenvolvimento, foi possível colocar em prática todos os conceitos levantados durante a pesquisa bibliográfica bem como as habilidades adquiridas nas disciplinas no decorrer do curso.

5.1. Trabalhos Futuros

Como trabalhos futuros a autora pretende desenvolver um artigo sobre o uso do modelo de processo de software Scrum, modelo este, apresentado pelos respondentes desta pesquisa e apontado pela literatura como sendo a tendência da atualidade para o futuro.

Pretende se também acompanhar essa tendência e observar até quando é viável utilizar-se dos modelos de processos de software tradicionais e modelos de processos de software ágeis, em cada projetos.

A autora pretende ainda aprofundar seus estudos na área de engenharia de software, realizando uma pós-graduação/mestrado nesse segmento, pois pretende se desenvolver softwares e apps, gerenciar projetos ligados aos softwares, arquitetar o design estrutural dos programas, realizar testes nos sistemas e gerenciar bancos de dados.

1. REFERÊNCIAS

- ANICETO, Jefferson. **Aplicações Web. Apostila ASP.net**. Escola Técnica da Univale (ETEIT). 2009.
- Beck, K., (2000) “**Extreme Programming Explained**”, Addison-Wesley.
- BECK, K.; BEEDLE, M.; VAN BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. **Manifesto for Agile Software Development**. p. 2–3, 2001. Disponível em: <<https://www.researchgate.net/file.PostFileLoader.html?id=57d055b593553b11467dd59&assetKey=AS%3A403742915612673%401473271220194>>.
- BECK, Kent. **Manifesto for agile development**. 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 01 jun. 2019.
- BEIGHLEY, Lynn. **Use a cabeça SQL**. São Paulo: Alta Books Editora, 2010.
- BELTRAME, M. M.; MAÇADA, A. C. G. **Validação de um Instrumento para medir o valor da tecnologia da informação (TI) para as organizações**. Disponível em: <<http://www.spell.org.br/documentos/download/5930>>. Acesso em: 13 de abr. 2019.
- BERNERS-LEE, T. **The World Wide Web — past, present and future**. Journal of Digital Information, [S.R.], v.1, n.1, Aug. 1996. ISSN: 1368-7506. Disponível em: <<https://journals.tdl.org/jodi/index.php/jodi/article/view/3/3>>. Acesso em: 27 mar. 2019.
- BERTOLLO, G. e FALBO, R. A. (2003) “**Apoio Automatizado à Definição de Processos de Software em Níveis**”. In: II Simpósio Brasileiro de Qualidade de Software (SBQS), Fortaleza, Brasil.
- BOEHM, B. W. **Get ready for agile methods, with care**. IEEE Computer, v. 35, p. 64–69, 2002.
- BORGES, Eduardo Pereira. **Um modelo de medição para processos de desenvolvimento de software**. Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Belo Horizonte, 2003.
- BRASIL, Lei Federal n.º 9394/96 – **Diretrizes e Bases da Educação**, Brasília, 1996.
- BRZEZINSKI, Iria (org.). **LDB interpretada: diversos olhares se entrecruzam**. São Paulo. Cortez. 2005.

BRASIL, Ministério da Educação. **Lei Federal n.º 9131/95**. Conselho Nacional de Educação, Brasília 1995.

BROWN, Alan W. **Large-scale component-based development**. Upper Saddle River, NJ: Prentice Hall, 2000. 286p.Campus.

CAMARGO, Robson. **Metodologia Agile: quais as mais utilizadas?** 2018. Disponível em: <https://robsoncamargo.com.br/blog/Metodologia-Agile-quais-as-mais-utilizadas>. Acesso em: 19 de jun. de 2019

CARVALHO, A.; CHIOSSI, T. **Introdução à Engenharia de Software**. São Paulo: Unicamp, 2001.

CARVALHO, A.; CHIOSSI, T. **Introdução à Engenharia de Software**. São Paulo: Unicamp, 2001.

CHIAVENATO, Idalberto. **Introdução à teoria geral da administração**. 6 ed. Rio de Janeiro: Campus, 2000. Coleção formação a distância & e-Learning, Inofor, 2002.

Cockburn, A. (2002) “Crystal Clear – **A human-powered methodology for small teams including the seven properties of effective software projects**”, Humans and Technology.

CONCEIÇÃO, Frederico do Nascimento. **Crêterios fundamentais que determinam a construção de uma liderança cristã em uma entidade religiosa do Distrito Federal**. UniCEUB – Centro universitário de Brasília. 2013.

CONSMASER, **Empresas de todos os tamanhos e segmentos podem desfrutar das inovações da plataforma SAP**. Disponível em: <<http://consmaster.com.br/2017/01/>>. Acesso em 29 out. 2018.

COPLIEN, J. H.; HARRISON, N. B. **Organizational Patterns of Agile Software Development**. Englewood Cliffs, NJ: Prentice Hall, 2004.

COUNCILL, W. T.; HEINEMAN, G. T. **Definition of a Software Component and its Elements**. In: Heineman, G. T.; Councill, W. T. (Orgs). Component-based Software Engineering. Boston: Addison-Wesley, 2001, p. 5-20.

CYBIS, W. A; BETIOL, A. H.; FAUST, R. **Ergonomia e Usabilidade: conhecimentos, métodos e aplicações**. Novatec, 2007.

D'souza, D., Wills, A. (1999) “**Objects, Components and Frameworks with UML – The Catalysis Approach**”, Addison-Wesley Publishing Company.

DAMKE, Elói. J; WALTER, S. Anita; SILVA, E. Damião. **A Administração é uma Ciência? Reflexões Epistemológicas acerca de sua Cientificidade**. Revista de Ciências da Administração, v. 12, n. 28, p. 125-144, set./dez.2010.

DE MELO, José Damião; MENDONÇA, Mikaele Costa; DIAS, Wanderson Roger Azevedo. **Estudo de caso no contexto da Engenharia de Software: SGCOPLEX/Case study in the context of Software Engineering: SGCOPLEX**. **Brazilian Journal of Development**, v. 5, n. 2, p. 1836-1851, 2019.

DOCUMENTAÇÃO DO PHP. **Introdução**. Disponível em: <<http://www.php.net/manual/en/intro.pdo.php>>. Acesso em: 22 out. 2018.

DOS SANTOS SILVA, Daisy Eliana; DE SOUZA, Ingredy Thaís; CAMARGO, Talita. **Metodologias Ágeis Para O Desenvolvimento De Software: Aplicação E O Uso Da Metodologia Scrum Em Contraste Ao Modelo Tradicional De Gerenciamento De Projetos**. Revista Computação Aplicada-UNG-Ser, v. 2, n. 1, p. 39-46, 2013.

DUARTE, Luiz. **9 vantagens de usar Scrum em seus projetos**, 2017. Disponível em: <https://www.luiztools.com.br/post/9-vantagens-de-usar-scrum-em-seus-projetos/>. Acesso em: 20 de jun. de 2019.

EHLERS, U.D. **Linking strategic directions for European eLearning beyond 2010**. In: eLearningLisboa07 Conference Proceedings. Coordenação: Isabel Vilhena. Fundação para a Divulgação das Tecnologias de Informação, Tipografia Guerra, Viseu, Lisboa: 2008, p. 17.

FOWLER, M.; SCOTT, K. **UML essencial um breve guia para a linguagem – padrão de modelagem de objetos**. 2. ed. Bookman, Porto Alegre, 2000.

FRIGERIO, A. (2008). **O paradigma da escolha racional: mercado regulado e pluralismo religioso**. **Tempo Social**, Revista de Sociologia da USP, 20(2), 17-39. Disponível em: <<http://www.revistas.usp.br/ts/article/view/12577>. <http://dx.doi.org/10.1590/S0103-20702008000200002>>. Acesso em 6 maio 2019.

FUGGETTA, A. (2000) **“Software Process: A Roadmap”**. In: 22nd International Conference on on Software Engineering (ICSE), Proceedings of the Conference on The Future of Software Engineering, pp. 25-34., New York: ACM Press.

FUSARI, José Cerchi; RIOS, Terezinha Azêredo. **Formação continuada dos profissionais do ensino**. Cadernos Cedes, v. 36, p. 37-45, 1995.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, Mass.: Addison-Wesley, 1995.

GREER, D. e CONRADI, R. (2009) “**Software project initiation and planning – an empirical study**”, IET Software, 3(5), p. 356–368

HIGHSMITH, Jim A. **Adaptive software development**. Dorset House Publishing, 1996.

JUNIOR, Silva et al. **Gestão de valor em projetos de TI: um estudo sobre organizações no Brasil**. Gestão & Produção, v. 26, n. 2, 2019.

KELLNER, M. I. (1996) "**Connecting Reusable Software Process Elements and Components**". In: 10th International Software Process Workshop, pp. 8-11, Dijon, France.

Kruchten, P. (2003) “**Introdução ao RUP – Rational Unified Process**”, Rio de Janeiro:

LAPENDA, Rodrigo; LONIEWSKI, G.; MADRUGA, P. **Utilizando transformação de modelos para o desenvolvimento de softwares de qualidade**. Revista Científica Tecnologus, v. 3, p. 1-6, 2008.

LEE, S.; YONG, H.-S. **Distributed agile: project management in a global environment**. *Empirical Software Engineering*, v. 15, n. 2, p. 204–217, 2010. Disponível em: <<http://link.springer.com/10.1007/s10664-009-9119-7>>.

LEFFINGWELL, D. **Scaling software agility: best practices for large enterprises**. [s.l.] Pearson Education, 2007.

LÜDERS, Edson et al. **Proposta metodológica simplificada para avaliação e operação de aplicações WEB nas Cidades Inteligentes em um ambiente seguro**. 2019.

LUTZ, Douglas Fernando. **Implantação de novo processo de trabalho em uma fábrica de software baseado nos modelos ágeis de desenvolvimento**. 2018. Trabalho de Conclusão de Curso.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: uma visão prática**. 16 ed. São Paulo: Érica, 2009.

MACHADO, Rafaela Sano. **Desenvolvimento ágil de software: uma revisão sistemática da literatura**. 2018.

MACIEL, Ira Maria. Educação a distância. **Ambiente virtual: construindo significados**. Boletim Técnico do SENAC, v. 28, n. 3, p. 38-45, 2018.

MAGRI, João Alexandre. **Arquitetura Dirigida a Modelos (MDA): Utilizando Modelos no Desenvolvimento de Sistemas**. Augusto Guzzo Revista Acadêmica, São Paulo, n. 8, p. 29-43, nov. 2008. ISSN 2316-3852.

MANUAL DO MPDF. **Documentação**. Disponível em: <<https://mpdf.github.io/>>. Acesso em 22 out. 2018.

MORER, Albert Sangrà. **E-learning value**. In: eLearningLisboa07 Conference Proceedings. Coordenação: Isabel Vilhena. Fundação para a Divulgação das Tecnologias de Informação, Tipografia Guerra, Viseu, Lisboa: 2008, p. 136.

NIEDERAUER, Juliano. **PHP para quem conhece PHP: Recursos avançados para criação de websites dinâmicos**. São Paulo: Novatec Editora, 2008.

NIELSEN, Jakob. **Usability Engineering**. San Francisco, CA, USA: Morgan Kaufmann

OSTERWEIL, L. (1987) **“Software Process Are Software Too”**. In: 9th International Conference on Software Engineering (ICSE), Monterey, Estados Unidos, p. 2-13.

PADILHA, R. P. **Planejamento dialógico: como construir o projeto político pedagógico da escola**. São Paulo: Cortez; Instituto Paulo Freire, 2001.

PAULA FILHO, Wilson de Pádua. **Engenharia de software: fundamentos, métodos e padrões**. 3 ed. Rio de Janeiro: LTC. 2013.

Portal /Engenharia de software, 2012. Disponível em: <<https://engenhariasoftware.wordpress.com/2013/01/24/a-aplicabilidade-do-modelo-cascata-na-engenharia-de-software/>>. Acesso em 10 de jun. de 2019.

PORTAL COMPOSER. **Introdução**. Disponível em: <<https://getcomposer.org/doc/00-intro.md>>. Acesso em 22 out. 2018.

PORTAL DEWILLIAM. **Biblioteca mpdf gerar relatórios em php**. Disponível em: <<http://www.devwilliam.com.br/php/biblioteca-mpdf-gerar-relatorios-em-php>>. Acesso em 22 out. 2018.

Portal GAEA. **Metodologia Ágil x Metodologia Tradicional – entenda as diferenças**. maio/2018. Disponível em: <https://gaea.com.br/metodologia-agil-x-metodologia-tradicional-entenda-as-diferencas/>. Acesso em: 20 de jun. 2019.

Portal Infomoney. **Acelere o crescimento da sua empresa em até 35% com uma plataforma ERP.** Disponível em: <<http://www.infomoney.com.br/conteudo-patrocinado/noticia/5206438/accelere-crescimento-sua-empresa-ate-com-uma-plataforma-erp>>. Acesso em: 06 de maio 2018.

PORTAL MACORATTI.NET, **Padrões de Projeto: O modelo MVC - Model View Controller.** Disponível em: <http://www.macoratti.net/vbn_mvc.htm>. Acesso em 22 out. 2018.

PORTAL MYSQL.COM. **Documentação.** Disponível em: <<https://www.mysql.com/products/workbench/>>. Acesso em 22 out. 2018.

PORTAL SIGNIFICADOS. **Significado de Ementa,** 2016. Disponível em:<<https://www.significados.com.br/ementa/>>. Acesso em 23 de jun de 2016.

PORTAL TECHTUDO. **Modelos softwares com Astah.** Disponível em: <<https://www.techtudo.com.br/tudo-sobre/astah-community.html>>. Acesso em out. 2018.

PRESSMAN, R. **Engenharia de Software: Uma Abordagem Profissional.** 7ª Ed., São Paulo: McGraw-Hill, 2011.

PRESSMAN, R. S. (2006) **“Software Engineering: A Practitioner's Approach”**, 6th Edition, New York: McGraw-Hill, 2006. Publishers Inc, 1993.

Pressman, Roger S. **Engenharia de Software - Uma Abordagem Profissional.** – Porto Alegre, Brasil: Editora AMGH, 2011, 773p.

REBRATES, Rede Brasileira do Terceiro Setor. **O Terceiro Setor.** Disponível em: <http://www.terceirosetor.org.br/quemsomos/index.cfm?page=terceiro>. Acesso em 25 de Maio 2018.

REIS, Christian Robottom; DE MATTOS FORTES, R. P. **Caracterização de um Processo de Software para Projetos de Software Livre.** Dissertação de Mestrado, Universidade de São Paulo, São Carlos-SP, 2003.

REIS, T. S; ALMEIDA, B. M. B. M. CORREIA, R. M. **Softwares de auxílio à docência no processo de Ensino-aprendizagem de Língua Portuguesa.** In: FUCHS, C; SCHWENGBER, I. L; SCHUTZ, J. A. Educação, Escola e Contemporaneidade: Questões e Diálogos. São Carlos. Pedro & João Editores. 2018. 421.

REZENDE, D. **Engenharia de Software e Sistemas de Informação.** 3ª Ed., Rio de Janeiro: Brasport, 2005.

REZENDE, D. **Engenharia de Software e Sistemas de Informação.** 3ª Ed., Rio de Janeiro: Brasport, 2005.

RIBEIRO, André Philipe Santos Pinto; BREZOLIN, Lúgia Maria T. de F.; LEONOR, Bruno Bustamante Ferreira. **ESTUDO DA APLICAÇÃO DAS METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE TRADICIONAIS E ÁGEIS UNIFICADAS AO**

DESENVOLVIMENTO LEAN DE SOFTWARE. Revista H-TEC Humanidades e Tecnologia, v. 2, n. 1, p. 96-112, 2018.

RIBEIRO, Leandro. Portal Devmidia. **O que é UML** Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em 22 out. 2018.

ROCHA, Ana Regina et al. **Estação TABA: Uma infra-estrutura para Implantação do Modelo de referência para Melhoria de Processo de Software.** Simpósio Brasileiro de Qualidade de Software, p. 49-60, 2005.

ROCHA, José Gladistone. **Arquitetura em Camadas com uso do Paradigma MVC e Processo Unificado na Programação de Software Orientado a Objetos.** TECNOLOGIAS EM PROJEÇÃO, v. 9, n. 1, p. 31-49, 2018.

SABBAGH, R. **SCRUM: Gestão Para Projetos De Sucesso.** Casa do Código. São Paulo. 2013.

SALVIANO, Clenio Figueiredo et al. **Uma proposta orientada a perfis de capacidade de processo para evolução da melhoria de processo de software.** 2006.

SAMETINGER, Johannes. **Software Engineering with Reusable Components.** New York: Springer, 1997. 271p.

SANTOS, Crystal de Menezes. Crystal: **uma linguagem de modelagem simplificada para apoiar a especificação de programas procedurais em banco de dados relacionais.** 2017. Dissertação de Mestrado. Universidade Federal de Pernambuco.

SANTOS, Nilson Alves. **DESENVOLVIMENTO E VALIDAÇÃO DE SOFTWARE DE AUXILIO A GESTÃO DE INSTITUIÇÕES EVANGELICAS:** Uma pratica a luz da igreja Salvação em Cristo de Porto Nacional – TO. – Instituto Federal de Educação, Ciência e Tecnologia, 2018. 140 f.

SCHNELL, Alexandre. **Uso do modelo cascata no desenvolvimento de um sistema para serviços de oficina mecânica automotiva.** 2018. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SCHWABER, Ken. **Agile project management with Scrum.** Estados Unidos da América: Microsoft Press, 2004.

SEIBEL, Tiago. **O Que São Aplicações WEB?** - 2008. Disponível em: <<http://www.tiagoseibel.com.br/2008/02/o-que-so-aplicaes-web.html>>. Acesso em 26 maio/2018.

Sellers, B. H. et. Al (1997) “**The OPEN Process (Tasks, Techniques and Management)**”, Chapter of Handbook of Object Technology, Centre for Object Technology Applications and Research – School of Information Technology – Swinburne University of Technology, CRC Press.

SETZER, Valdemar W; SILVA, Flávio Soares Corrêa da. **Banco de Dados: Aprenda o que são melhora seu conhecimento construa os seus**.1 ed. São Paulo: Edgard Blucher, 2005.

SHNEIDERMAN, Ben; PLAISANT, Catherine. **Designing the user interface: Strategies for effective Human-Computer Interaction**. 4th edition. Berkeley, California: Addison Wesley Longman, Inc., 2004.

SILVA Fernanda G.; HOENTSCH, Sandra C. P.; SILVA, Leila M. de A. **Uma análise das Metodologias Ágeis FDD e Scrum sob a perspectiva do modelo de qualidade MPS.BR**. Scientia Plena, v. 5, n. 12, 2009

SILVA, E. LOVATO, L. **Framework Scrum: Eficiência Em Projetos De Software**. Revista de Gestão e Projetos – GeP Vol. 7, N. 2. Maio/Agosto. 2016.

SILVA, M. S. (2015). **Bootstrap 3.3.5**. Novatec Editora, Sao Paulo, SP.

SNYDER, Howard A. e RUNYON, Daniel. **Decoding the Church**, 2002 p. 61

Sommerville, Ian. **Engenharia de Software**. – São Paulo, Brasil: Editora PEARSON, 2003, 353p.

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves ; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

STAPLETON, Jennifer. **DSDM: a framework for business centered development**. Addison Wesley Professional, 1997.

SUTHERLAND, J. SCHWABER, K. (2013). **Um Guia Definitivo para o Scrum – As Regras do Jogo**. Julho de 2013. Disponível em: <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-PortugueseBrazilian.pdf>

SUTHERLAND, J. **Scrum A Arte De Fazer O Dobro Do Trabalho Na Metade Do Tempo**. LeYa. Pacaembu — São Paulo, SP, 2014.

SZYPERSKI, C. **Component Software: Beyond Object-oriented Programming**, 2a ed. Harlow, Reino Unido: AddisonWesley, 2002.

SZYPERSKI, Clemens. **Component Software: beyond object-oriented programming**. Harlow: Addison-Wesley, 1999. 411p.

TERLIZZI, M. BIANCOLINO, C. **Projeto De Software No Setor Bancário: Scrum Ou Modelo V**. TAC, Rio de Janeiro, v. 4, n. 1, art. 4, pp. 46-58, Jan./Jun. 2014

VEIGA, Ilma Passos Alencastro; DE RESENDE RESENDE, Lúcia Maria Gonçalves. **Escola: espaço do projeto político-pedagógico**. Papirus Editora, 1998.

VINICIUS. **Processo de teste ágil x tradicional**. Portal DevMedia, 2016. Disponível em: <<https://www.devmedia.com.br/processo-de-teste-agil-x-tradicional/36854>>. Acesso em: 19 de jun. de 2019.

2. APÊNDICES

2.1. APÊNDICE A - Questionário usado na coleta de dados da pesquisa de avaliação do sistema

2.1.1. PARTE I - APRESENTAÇÃO DA PESQUISA

Este questionário é parte integrante de uma pesquisa de elaboração da monografia para obtenção do título de licenciado em computação junto ao Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, sob orientação do Professor M.e Rafael Miranda Correia.

A pesquisa objetiva analisar a tendência do mercado tocantinense atual quanto ao uso dos modelos de desenvolvimento de software, bem como verificar junto as especialistas suas respectivas utilizações.

Desta forma, as informações a serem inseridas serão utilizadas exclusivamente para fins desta pesquisa, não existindo a total obrigatoriedade do preenchimento da parte II deste questionário onde são encontradas as perguntas sobre a identificação dos respondentes.

2.1.2. PARTE II - INFORMAÇÕES DO ENTREVISTADO

Nome:

E-mail:

Telefone:

Cidade:

Nome da Empresa ou Instituição na qual trabalha:

Qual seu Vínculo com a Empresa ou Instituição:

☐ Professor(a)

☐ Analista de Sistema

☐ Desenvolvedor de Sistema

☐ Outros

Área de Formação:

2.1.3. PARTE III – SOBRE OS MODELOS DE DESENVOLVIMENTO DE SOFTWARE

1. A escolha do modelo deve levar em conta fatores como: cultura da empresa, nível de maturidade dos processos, natureza da aplicação e do projeto, entre outros. Dentre os modelos de desenvolvimento de software abaixo quais você mais utiliza?

- ☐ Cascata
- ☐ Prototipação
- ☐ Evolutivo
- ☐ Incremental
- ☐ Transformação
- ☐ Espiral
- ☐ Iterativo
- ☐ Baseado em Componentes
- ☐ Programação Extrema - XP
- ☐ Processo Unificado da Rational - RUP
- ☐ Arquitetura Dirigida pelo Modelo - MDA
- ☐ Desenvolvimento Rápido de Aplicação - RAD
- ☐ Praxis
- ☐ Em V

2. Em relação aos modelos correlacionados abaixo responda quais características levam a utilizá-lo:

Linhas	colunas
1.Cascata 2.Prototipação 3.Evolutivo 4.Incremental 5.Transformação 6.Espiral 7.Iterativo 8.Baseado em Componentes 9.Programação Extrema - XP 10.Processo Unificado da Rational - RUP 11.Arquitetura Dirigida pelo Modelo - MDA	1. As fases são bem definidas. 2. Permite o planejamento das fases seguintes. 3. As etapas são divididas. 4. A construção de um sistema menor é sempre menos arriscada que a construção de um grande. 5. Requer pequeno investimento inicial. 6. É possível obter uma versão mesmo que simplificada do que será o sistema. 7. Permite controle departamental e gerencial. 8. As fases seguem sem sobreposições.

12.Desenvolvimento Rápido de Aplicação - RAD 13.Praxis 14.Em V	9.Cobre vasta gama de atividades. 10.Permite encontrar defeitos de forma antecipada. 11.Aplicado a projetos de alto risco. 12.Conjunto bem definido de regras. 13.Flexibilidade. 14.Sem estimativa de custo e prazos. 15.Alta capacidade de reutilização. 16.Os requisitos e as táticas de alteração são acomodados. 17.Possui interfaces explicitas. 18.Não utilizo esse modelo.
--	--

2. Em relação aos modelos correlacionados abaixo responda quais características levam a utiliza-lo:

4. Você conhece algum modelo que não foi citado na pergunta 1? Se sim, qual (is)?

5. Por favor, informar qualquer outra consideração que julgue ser relevante:

3. ANEXOS

3.1. ANEXO 1 - Plano de Ensino da disciplina de Engenharia de Software



MINISTÉRIO DA EDUCAÇÃO
 SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
 TOCANTINS
 CAMPUS PORTO NACIONAL
 GERÊNCIA DE ENSINO
 COORDENAÇÃO DO CURSO SUPERIOR DE LICENCIATURA EM COMPUTAÇÃO

PLANO DE ENSINO ENGENHARIA DE SOFTWARE

Componente: **Engenharia de Software**

Carga horária **Total: 80 Carga horária Teórica: 40 Carga horária Prática: 40**

Ano/Semestre: **2019/1 Período: 6º**

Professor(a): **Rafael Miranda Correia** Curso: **Licenciatura em Computação**

1. EMENTA:

Introdução à Engenharia de Software: Conceitos e definições. Sistemas Computacionais. O que é software? O que é engenharia de software? O Ciclo de Vida do Software. Qualidade de Software. Processo de Software: Modelos de processo: Cascata, Prototipação, Evolutivo, Incremental, Transformação, Espiral. Slides. Modelos de processo: Processo Unificado.

Atividades do Processo: Especificação, Design e implementação, Validação e Evolução.

Métricas, Planejamento e

Gerenciamento de Software: Elaboração do cronograma. Planejamento da equipe. Estimativas e Métricas. Análise de riscos. Requisitos de Software: Requisitos e Engenharia de Requisitos. Definindo Requisitos com Casos de Uso. Slides. Modelos de Software. Design de Software: Design Conceitual, Prototipação. Arquitetura de

Software conceitos, visão tradicional e visão emergente. Visões arquiteturais. Linguagens de Descrição Arquitetural. Padrões de Projeto. Framework. Verificação e Validação de Software: Formas de verificação e validação de programas. Técnicas de testes. Manutenção e Evolução de Software.

2. COMPETÊNCIAS:

Apresentar, analisar e discutir o corpo de conhecimento que constitui a Engenharia de

Software, seus princípios, métodos e ferramentas. Identificar as técnicas da Engenharia de Software para o desenvolvimento de um produto; Ser capaz de descrever os métodos, ferramentas e procedimentos associados; Conhecer e utilizar as técnicas de trabalho em grupo, especificamente para desenvolvimento de softwares; Saber identificar os princípios da ética profissional do engenheiro de software.

3. HABILIDADES:

Durante o desenvolvimento desta disciplina o aluno terá a oportunidade de aprender os conceitos e praticas da engenharia de software e gerencia de projetos. Com isso, será capacitado a fazer analise documentada de um softwares por completo e também gerenciar o desenvolvimento de softwares.

4. BASES TECNOLÓGICAS:

4.1. Introdução à Engenharia de Software:

4.1.2. Conceitos e definições.

Entender os conceitos básicos de engenharia de software 4.1.3. Sistemas Computacionais.

Entender os tipos de sistemas computacionais 4.1.4. O que é software?

Entender o que é um software

4.1.5. O que é engenharia de software? Entender o que é a engenharia de software 4.1.6. O Ciclo de Vida do Software.

Entender o funcionamento de cada um dos ciclos de vida de um software 4.1.7. Qualidade de Software. Compreender o a qualidade de software 4.2. Processo de Software:

4.2.1. Modelos de processo:

4.2.2. Cascata, Prototipação,

Entender o tipos de processo de software cascata e prototipação

4.2.3. Evolutivo, Incremental,

Entender o tipos de processo de software evolutivo e incremental

4.2.4. Transformação,

Entender o tipos de processo de software transformação 4.2.5. Espiral.

Entender o tipos de processo de software espiral 4.3. Modelos de processo:

4.3.1. Processo Unificado. Compreender o processo unificado 4.4.

Atividades do Processo:

4.4.1. Especificação,

Entender a especificação de processo

4.4.2. Design e implementação,

Entender sobre design e implementação de processos 4.4.3. Validação e

Evolução.

Entender como fazer a validação e evolução 4.4.4. Métricas,

Entender as métricas.

4.5. Planejamento e Gerenciamento de Software:

4.5.1. Elaboração do cronograma. Aprender a elaborar cronograma

4.5.2. Planejamento da equipe. Entender sobre planejamento de equipe

4.5.3. Estimativas e Métricas. Entender as estimativas e métricas 4.5.4.

Análise de riscos.

Fazer a analise dos riscos

4.6. Requisitos de Software:

4.6.1. Requisitos e Engenharia de Requisitos. Aprender a identificar os

requisitos de softwares 4.6.2. Definindo Requisitos com Casos de Uso.

Utilizar a uml para definir os casos de uso 4.6.3. Modelos de Software.

Entender sobre os modelos de software 4.7. Design de Software:

4.7.1. Design Conceitual, Entender sobre o design conceitual 4.7.2.

Prototipação.

Entender sobre prototipação

4.7.3. Arquitetura de Software conceitos,

Entender os conceitos da arquitetura de software 4.7.4. visão tradicional e

visão emergente.

Entender sobre a visão tradicional e a visão emergente.

4.7.5. Visões arquiteturais.

Entender sobre as visões de arquitetura

4.7.6. Linguagens de Descrição Arquitetural. Entender sobre as linguagens de descrição da arquitetura 4.7.7. Padrões de Projeto.

Entender sobre os padrões de projetos 4.7.8. Framework.

Entender sobre framework

4.7.9. Verificação e Validação de Software:

4.7.10. Formas de verificação e validação de programas. Entender as formas de verificação e validação de programas 4.7.11. Técnicas de testes.

Entender as técnicas de testes

4.7.12. Manutenção e Evolução de Software. Entender sobre a manutenção e evolução de software

5. METODOLOGIA:

Aula teórica e prática através de exposição dialogada utilizando recurso como pincel, quadro, material impresso, data show, hardwares e computador. Realização de exercícios e trabalhos sobre os conceitos apresentados.

Aulas expositivas

- Aulas práticas e demonstrativas
- Exercícios práticos
- Textos Técnicos

6. RECURSOS DIDÁTICOS:

Pincel, quadro, material impresso, data show, hardwares e computadores

7. AVALIAÇÃO:

1º Bimestre uma prova valendo 7,0 e um trabalho valendo 3,0 assim totalizando 10,0 2º Bimestre uma prova valendo 7,0 e um trabalho valendo 3,0 assim totalizando 10,0

8. BIBLIOGRAFIA:

8.1 Básica

BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio Janeiro: Campus, 2006. BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. UML: Guia do Usuário. 2. ed.

Rio Janeiro: Campus, 2006. PAULA FILHO, W. P. Engenharia de software: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2003. REFERÊNCIAS BIBLIOGRÁFICAS PFLEEGER, S. L. Engenharia de software: teoria e prática. São Paulo: Pearson, 2004.

PRESSMAN, R. Engenharia de software. Rio de Janeiro: MacGraw-Hill, 2006.

SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Addison Wesley, 2007.

8.2 Complementar

FURLAN, J. D. Modelagem de objetos através da UML. São Paulo: Makron Books, 1998. LARMAN, C. Utilizando UML e padrões: um guia para a análise e projeto orientados a objetos. 3. Porto Alegre: Editora Bookman, 2007.

Rafael Miranda Correia

Coordenador do Curso Superior de Licenciatura em Computação

Portaria nº 109/2016/IFTO/ Campus Porto Nacional



Documento assinado eletronicamente por **Rafael Miranda Correia**, Coordenador, em 21/02/2019, às 21:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.iftoparodonta.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0598054** e o código CRC **E47A8E30**.

Av. Tocantins, Loteamento Mãe Dedé

Setor - Jardim América

Referência: Processo nº 23337.001925/2019-90 SEI nº 0598054

3.2. |

3.3. ANEXO 2 – Artigo Científico

TENDÊNCIA DO MERCADO QUANTO AO USO DOS MODELOS DE DESENVOLVIMENTO DE SOFTWARE: Uma Experiência Tocantinense

MARKET TRENDS ON THE USE OF SOFTWARE DEVELOPMENT MODELS: A Tocantinian Experience

Juliana de Souza Pinheiro

Acadêmica do Curso Superior de Licenciatura em Computação do Campus Porto Nacional do Instituto Federal do Tocantins - IFTO
E-mail: julianadespinheiro@gmail.com

.....

Rafael Miranda Correia

Coordenador do Curso Superior de Licenciatura em Computação do Campus Porto Nacional do Instituto Federal do Tocantins - IFTO
E-mail: rafaelmiranda@ifto.edu.br

Resumo

Este trabalho aborda sobre a tendência do mercado quanto ao uso dos modelos de desenvolvimento de software: Uma experiência tocantinense, objetivando analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software. Para a realização deste estudo foram utilizadas pesquisas bibliográficas através de livros, artigos científicos e sites para obtenção de informações e a pesquisa de campo considerada quantitativa, utilizada para coleta de dados. O universo da pesquisa em que o estudo foi desenvolvido está definido com 61 participantes, deste total participaram diversos profissionais da área de desenvolvimento de software conforme discriminados a seguir: 15 da Tecnologia da informação, 14 de Banco de Dados, 8 de Engenharia da computação, 10 de Ciências da computação, 9 de Análise de sistema, 2 de Técnico em informática e 3 de Sistemas para internet. Os resultados obtidos mostram que 69% dos respondentes utilizam modelos tradicionais, enquanto 31% utilizam modelos ágeis. Os modelos ágeis ainda são menos usados, a pesar da literatura aponta-los como tendência. A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade, mostrando com isso a relevância desse trabalho.

Palavras-chave: Engenharia. Desenvolvimento. Modelos. Processo. Software.

Abstract

This paper discusses the market trend in the use of software development models: A Tocantins' experience, aiming to analyze the current market trend in the use of software development models. For the accomplishment of this study were used bibliographical research through books, scientific articles and sites to obtain information and field research considered quantitative, used for data collection, The universe of the research in which the study was developed is defined with 61 participants, of this total, several professionals from the area of software development participated as follows: 15 of Information Technology, 14 of Database, 8 of Computer Engineering, 10 of Computer Science, 9 of System Analysis, 2 of Technician in computer science and 3 of Systems for internet. The results show that 69% of respondents use traditional models, while 31% use agile models. Agile models are still less widely used, despite the literature pointing them as a trend. The research also showed that 10% of respondents know another model called Scrum, which according to the literature is the model that follows the trend of contemporaneity, thus showing the relevance of this work.

Key words: Engineering. Development. Models. Process. Software.

Introdução

Na velocidade com que a tecnologia da informação se desenvolve e expande, novos serviços e produtos de Software surgem a todo o momento, fazendo com que o grau de complexidade das aplicações aumente.

O desenvolvimento de software é uma tarefa muito complexa, difícil e de alto risco Vinicius (2016). Exemplos disso são as aplicações web que estão crescendo e se tornando cada vez mais complexas e importantes no dia-a-dia das pessoas.

Assim, Silva, Souza e Camargo (2013), preconiza que as organizações enfrentam certas dificuldades diante das metodologias disponíveis para o gerenciamento de projetos de software.

Por isso, adotar e utilizar processos de desenvolvimento maduro e consistente constitui o ponto chave para o sucesso de projetos. Neste sentido, o presente trabalho visa explorar a sua temática: tendência do mercado quanto ao uso dos modelos de desenvolvimento de software: Uma Experiência Tocantinense.

Nesse contexto, a problemática evidenciada foi se o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do campus Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados?

Assim, os objetivos gerais e específicos a serem alcançados através desta pesquisa são, respectivamente: Analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software; a) Identificar os modelos de desenvolvimento de software; b) Verificar junto as especialistas tocantinenses as respectivas utilizações dos modelos de processo de desenvolvimento de software; c) Comparar a atual tendência do mercado sobre os modelos de desenvolvimento de software com a disciplina de Engenharia de Software do Campus Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia.

Sendo assim, levantou-se os aspectos teóricos sobre as áreas envolvidas no presente trabalho, tais como, engenharia de software; os modelos de desenvolvimento de software; a atual tendência do mercado quanto ao uso dos modelos de desenvolvimento de software; ementas das disciplinas de engenharia de software.

Nessa lógica, partiu-se para os procedimentos metodológicos, apresentando os materiais e métodos, extração e identificação dos modelos de processo de software e suas características de usabilidade, por fim foi realizada uma pesquisa de campo. Como instrumento de coleta dos dados, utilizou-se um questionário aplicado via Web. confeccionado utilizando-se da plataforma google form. O formulário foi submetido ao universo da pesquisa, onde houve um retorno satisfatório das respostas.

Em suma, o método de pesquisa utilizado empregou-se uma pesquisa no enfoque quantitativo. A pesquisa foi realizada com o universo de 70 participante, 61 responderam ao questionário, sendo que estes respondentes atuam em diversas áreas do desenvolvimento de software, onde, 25% são da área da Tecnologia da informação, 23% de bancos de dados, 16% da área de Ciência da computação, 15% de análise de sistemas, 13% da área de engenharia da computação, 5% de Sistema para internet e por fim 3% de Técnico em Informática.

Além dos seis modelos de processos de software, Ementa/Plano de Ensino da Disciplina de Engenharia de Software, que são: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral, a pesquisa apresentou mais sete modelos que são: Iterativo, Baseado em Componentes, Programação Extrema – XP, Processo Unificado da Rational – RUP, Arquitetura Dirigida pelo Modelo – MDA, Desenvolvimento Rápido de Aplicação – RAD e Em V.

A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade. Portanto conclui-se que a ementa do Curso de Licenciatura em Computação do campus Porto Nacional, atende parcialmente a tendência do mercado quanto aos modelos de desenvolvimento de software utilizados.

Referencial teórico

Este capítulo apresentará os aspectos teóricos sobre as áreas envolvidas no presente trabalho. São apresentados conceitos sobre a engenharia de software; os modelos de desenvolvimento de software; a atual tendência do mercado quanto ao uso dos modelos de desenvolvimento de software; ementas das disciplinas de engenharia de software.

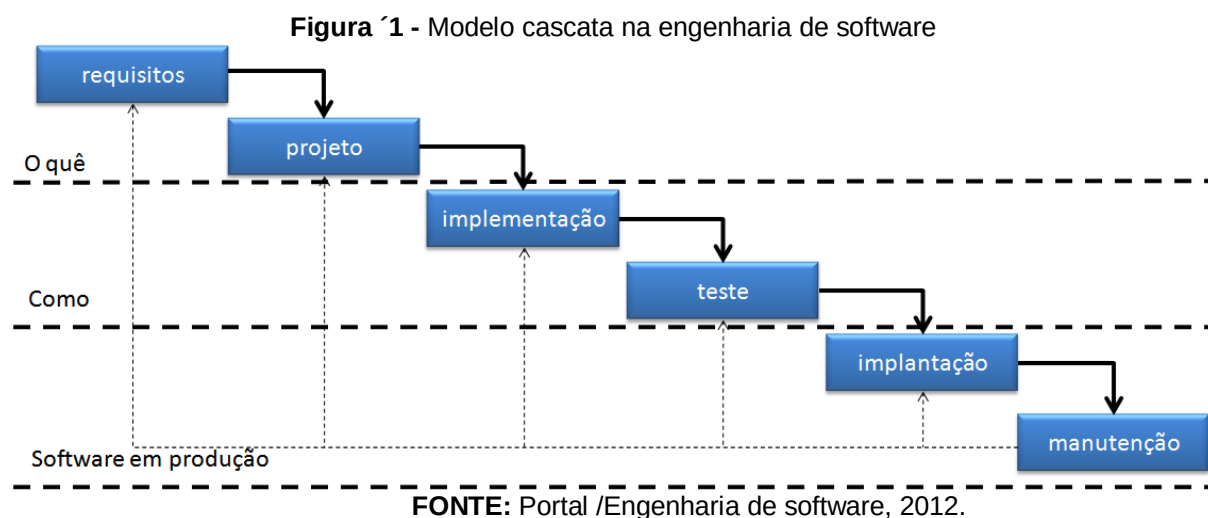
Engenharia de software

A engenharia de software é essencial no desenvolvimento de sistemas ao possibilitar que o analista se aproprie e utilize metodologias assertivas para projetos de software.

Sommerville (2011) afirma que a engenharia de software é responsável por todos os aspectos da produção de software, iniciando na especificação de sistema até a manutenção desse sistema depois que ele já está em operação. É uma disciplina que tem como meta apresentar técnicas de desenvolvimento de softwares com alta qualidade a um baixo custo.

Segundo Pressman (2011), a engenharia de software é formada por processo, métodos (práticas) e ferramentas as quais permitem construir sistemas dentro do prazo e com qualidade.

Em síntese a engenharia de software é, portanto, o conjunto de processos, métodos e ferramentas que podem ser representadas na forma de camadas (Figura 1), comprometida com a qualidade.



Os modelos de desenvolvimento de software

Um processo de software corresponde a um conjunto de passos formados por atividades, métodos, práticas e transformações parcialmente ordenados, os quais são usados para desenvolver e manter software e produtos associados (Sommerville, 2003).

Conforme Pressman (2011), processo de software é um conjunto de atividades de trabalho, ações e tarefas realizadas quando algum artefato de software deve ser criado. O processo não é uma prescrição rígida de como desenvolver um

software. Ao contrário, é uma abordagem adaptável que possibilita a equipe de software realizar o trabalho de selecionar e escolher o conjunto apropriado de ações e tarefas.

Existem muitos processos de *software* diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de *software* (SOMMEVILLE 2011). Conforme descrito na Tabela 1.

Tabela1– Atividades para processo de software

Atividades	Descrição
2. Especificação de software.	A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
2. Projeto e implementação de software.	O software deve ser produzido para atender às especificações.
3. Validação de software.	O software deve ser validado para garantir que atenda às demandas do cliente.
4. Evolução de software.	O software deve evoluir para atender às necessidades de mudança dos clientes.

Fonte: Santos (2018), Apud SOMMERVILLE, 2011.

A atual tendência de mercado quanto ao uso dos modelos de processos de software

Na literatura especializada pode-se encontrar duas frentes de processo de software no tocante à execução. A primeira, definida por Pressman (2011) como um processo tradicional, ou também chamada de processo pesado, que se caracteriza por possuir como foco principal a previsibilidade dos requisitos do sistema o comando e o controle destes a partir de um prévio planejamento antes do início do desenvolvimento.

Já a outra frente, chamada de processo ágil ou processo leve, possui seu foco na eficiência, abordando como premissa o compromisso entre “nada de processo” e processos rigorosos Beck (2000). Esta frente propõe que a análise de requisitos seja extremamente mutável, “abraçando” mudanças como principal área de atuação da metodologia.

Machado (2018), os segmentos de mercado acompanham as tendências incorporadas aos novos ciclos econômicos, sendo, nesse contexto, necessário ter medidas adaptáveis, rápido aprendizado e adoção de estratégias modernas. A metodologia ágil é, portanto, um processo cada vez mais relevante no gerenciamento de projetos corporativos, portal (GAEA, 2018).

A mudança para os Métodos Ágeis de desenvolvimento de software teve início em torno da década de 1990, e foi proposta por muitos praticantes de diversas nacionalidades e em perspectivas de projetos diferentes (LEFFINGWELL, 2007).

Em uma edição da Revista Computer, no ano 2000, Boehm (2002) afirmou que era necessário mudar o desenvolvimento de software de um processo lento para um mais rápido e dinâmico. No ano 2001, os métodos ágeis tornaram-se mais conhecidos devido à publicação do Manifesto Ágil (BECK, 2001). Nele, houve a união de dezessete especialistas em diversas abordagens de processos de desenvolvimento de software que buscaram estabelecer princípios comuns em práticas ágeis e compartilharam todas essas perspectivas.

O Manifesto Ágil propõe a filosofia de que no desenvolvimento do software os recursos são escassos e que os requisitos são altamente voláteis. Práticas ágeis

podem auxiliar o cumprimento do desenvolvimento do software devido às práticas de gerenciamento, de processo e de desenvolvimento de software (LEE; YONG, 2010).

O portal LuizToos afirma que desde 13 de novembro de 2001, quando o Manifesto Ágil foi assinado por dezessete especialistas da indústria de software da época, que o uso dos métodos ágeis só cresce. Em parte pelo demérito dos métodos tradicionais de desenvolvimento que priorizam muitas vezes a burocracia em detrimento do resultado.

Nessa perspectiva, Duarte (2017), afirma que o Scrum é o processo de software mais utilizado atualmente. Camargo (2018), concorda que metodologia Ágil Scrum é, sem dúvida, o método mais usado nos dias de hoje, principalmente porque pode ser integrado a outros métodos ágeis com facilidade, aplicando-se não só ao desenvolvimento de softwares como a qualquer ambiente de trabalho.

Ementa e plano de ensino da disciplina de engenharia de software

A ementa é como um resumo ou sinopse de determinado tema ou área. Quando se fala em “ementa do curso”, significa a apresentação de um texto que evidencia as principais características deste curso (PORTAL SIGNIFICADOS, 2016).

O Plano de Ensino é um plano de ação; é o registro do planejamento das ações pedagógicas para o componente curricular durante o período letivo. É um instrumento didático-pedagógico e administrativo de elaboração e uso obrigatórios. Pode-se pensar, primeiramente, que:

“[...] planejamento é processo de busca de equilíbrio entre meios e fins, entre recursos e objetivos, visando ao melhor funcionamento de empresas, instituições, setores de trabalho, organizações grupais e outras atividades humanas. O ato de planejar é sempre processo de reflexão, de tomada de decisão sobre a ação; processo de previsão de necessidades (...) visando à concretização de objetivos, em prazos determinados e etapas definidas, a partir dos resultados das avaliações” (PADILHA, 2001).

Sendo plano, trata-se de um “documento utilizado para o registro de decisões do tipo: o que se pensa fazer, como fazer, quando fazer, com que fazer, com quem fazer. Para existir plano é necessária a discussão sobre fins e objetivos, culminando com a definição dos mesmos” (PADILHA, 2001).

Dentro dessas definições, o Plano de ensino da disciplina de Engenharia de Software descreve no seu tópico 4 – Bases Tecnológicas, os modelos e processos de software utilizados/ministrados nas aulas da disciplina de Engenharia de Software IFTO (2018), conforme descrito a seguir: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral. Por sua vez, o referido Plano de Ensino da disciplina de Engenharia de Software, segue como anexo A deste trabalho.

Procedimentos metodológicos

Este capítulo apresenta a metodologia científica adotada neste trabalho, de forma a sistematizar a revisão bibliográfica para descrever sobre a Tendência do mercado quanto ao uso dos modelos de desenvolvimento de software, seguida como foi realizada a pesquisa, confecção e aplicação do questionário.

Levantamento dos modelos de processo de desenvolvimento de software

Para tanto, fez-se um levantamento teórico, realizado por meio de uma pesquisa bibliográfica, uma vez que a pesquisadora busca embasamento na literatura

existente. A pesquisa bibliográfica foi realizada a partir do levantamento de referências teóricas já analisadas, e publicadas por meios escritos e eletrônicos, como livros, artigos científicos, páginas de web sites.

Na opinião de Fonseca, (2002, p. 32). “Qualquer trabalho científico inicia-se com uma pesquisa bibliográfica, que permite ao pesquisador conhecer o que já se estudou sobre o assunto”. Existem, porém, pesquisas científicas que se baseiam unicamente na pesquisa bibliográfica, procurando referências teóricas publicadas com o objetivo de recolher informações ou conhecimentos prévios sobre o problema a respeito do qual se procura a resposta.

Com base nesse contexto, definiu as etapas para extração dos métodos usados nessa pesquisa.

1º Etapa: A pesquisa deu-se início a partir de um problema diagnosticado: o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do campus Porto Nacional está atualizado conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados?

2º Etapa: Diante do problema diagnóstico foi feito os seguintes termos de busca: engenharia + software; os modelos + desenvolvimento + software; tendência + mercado + modelos + desenvolvimento + software; ementas + disciplinas + engenharia de software

3º Etapa: Depois de elaborar os termos de busca fez-se uma pesquisa nas bases de dados afim de extrair modelos de desenvolvimento de software em artigos, livros e revistas.

4º Etapa: Após as pesquisas realizadas nas bases de dados fez-se uma seleção dos trabalhos mais relevantes para o decorrer da pesquisa.

5º Etapa: Nesta fase realizou-se a tradução dos trabalhos que se encontravam na língua inglesa.

6º Etapa: Após a tradução dos artigos, livros e revistas foram extraídos os modelos de desenvolvimento de software, conforme descritos nas tabelas 2

Tabela 2 – Modelos de processos software

Ordem	Modelo
1	Cascata
2	Prototipação
3	Evolutivo
4	Incremental
5	Transformação
6	Espiral
7	Iterativo
8	Baseado em Componentes
9	Programação Extrema - XP
10	Processo Unificado da Rational – RUP
11	Arquitetura Dirigida pelo Modelo – MDA
12	Desenvolvimento Rápido de Aplicação – RAD
13	Praxis
14	Em V

Fonte: A autora (2019).

A pesquisa

Para a consecução do presente trabalho foi adotado uma classificação de pesquisa de campo, conforme Fonseca (2002), pesquisa de campo caracteriza-se pelas investigações em que, além da pesquisa bibliográfica e/ou documental, se realiza coleta de dados junto a pessoas, com o recurso de diferentes tipos de pesquisa (pesquisa *ex-post-facto*, pesquisa-ação, pesquisa participante, etc.)

Desta forma, fez-se o uso dos métodos estatísticos, tendo em vista a necessidade de tornar quantificável algumas informações do questionário.

Confecção do Questionário da Pesquisa

O questionário é um instrumento ou programa de coleta de dados que pode ser enviado via correio, fax, Internet, etc., o questionário pode ser estruturado não disfarçado: o respondente sabe qual é o objetivo da pesquisa, e o questionário é padronizado, usando principalmente questões fechadas.

Assim, a edição do questionário foi realizada em alguns passos: o primeiro foi pesquisa sobre como confeccioná-los, o segundo passo foi a criação dos escopos para tomada de base na edição, o terceiro passo foi as correções dos escopos, o quarto passo foi a confecção do questionário no Google Docs, ferramenta gratuita para criação de formulários, sendo fundamental para a realização da pesquisa por meio da internet.

Submissão do Questionário da Pesquisa

Após a confecção foi realizada a submissão do questionário de pesquisa, com o objetivo de coletar os dados para verificação da tendência do mercado quanto ao uso dos modelos de desenvolvimento de software no âmbito do estado do Tocantins, o questionário de pesquisa foi submetido por meio de um formulário Google Docs que permite aos participantes responderem via internet, outra vantagem deste recurso é enviar o link do formulário via e-mail, possibilitando que o mesmo seja respondido com tempo pelo participante e de qualquer lugar que disponha de internet.

Retorno e tabulação das respostas do formulário de pesquisa

Como explicado no item anterior, foi enviado um link do formulário de pesquisa via e-mail, com uma nota explicativa, isso possibilitou o retorno das respostas em tempo que o questionário era respondido pelo universo da pesquisa.

Com os dados brutos em mãos, o processamento dessas informações se deu das seguintes formas: Em primeiro momento as coletas dos dados foram feitas em planilhas eletrônicas do pacote Google Docs, no segundo momento os dados foram salvos em planilhas do Microsoft Excel para facilitar a análise e tabulação dos dados, em terceiro momento foi aplicado os métodos estatísticos para a tabulação dos dados. E por fim, os resultados foram impressos em gráficos e tabelas geradas pela ferramenta utilizada, para serem apresentados no capítulo Análises e Resultados dos Dados.

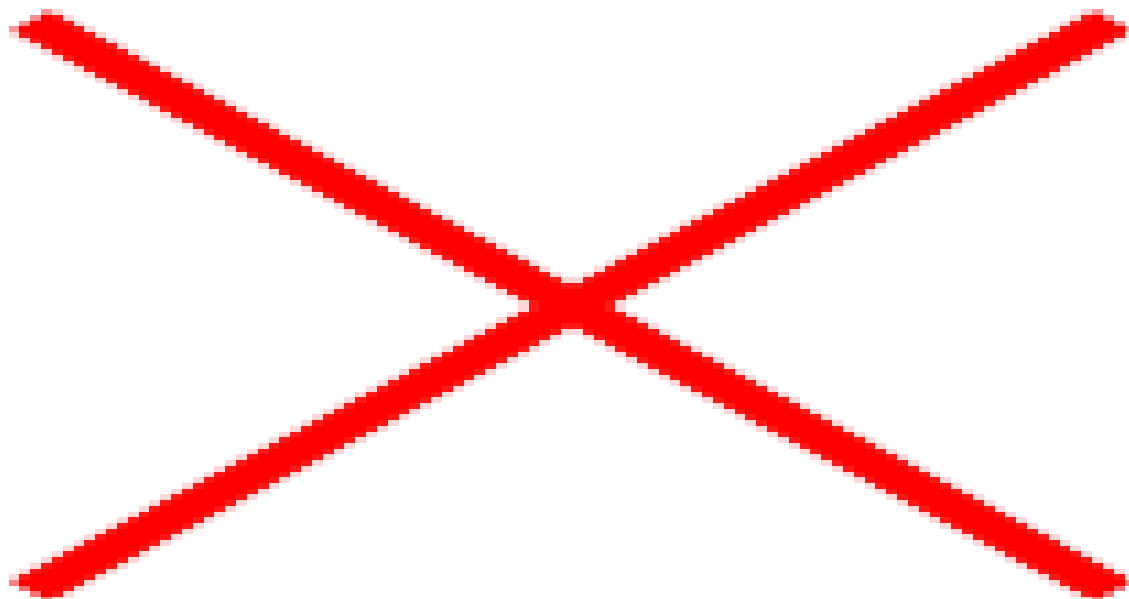
Análises e Resultados dos Dados

Neste Capítulo são apresentados os resultados e análises obtidas com a pesquisa, Tendência do mercado quanto ao uso dos modelos de desenvolvimento de

software: Uma experiência tocantinense, foram levantados os seguintes questionamentos: 1). A escolha do modelo deve levar em conta fatores como: cultura da empresa, nível de maturidade dos processos, natureza da aplicação e do projeto, entre outros. Dentre os modelos de desenvolvimento de software abaixo quais você mais utiliza? 2). Caso existam características que não foram listadas na pergunta anterior cite-as identificando o modelo; 3). Você conhece algum modelo que não foi citado na pergunta 1? Se sim, qual (is)? 4). Por favor, informar qualquer outra consideração que julgue ser relevante. Além de informações pessoais e área de atuação dos respondentes.

Do universo da pesquisa de 70 pessoas, 61 responderam ao questionário, sendo que estes respondentes atuam em diversas áreas do desenvolvimento de software, conforme descrito na figura 2, onde 25% são da área da Tecnologia da informação, 23% de bancos de dados, 16% da área de Ciência da computação, 15% de análise de sistemas, 13% da área de engenharia da computação, 5% de Sistema para internet e por fim 3% de Técnico em Informática.

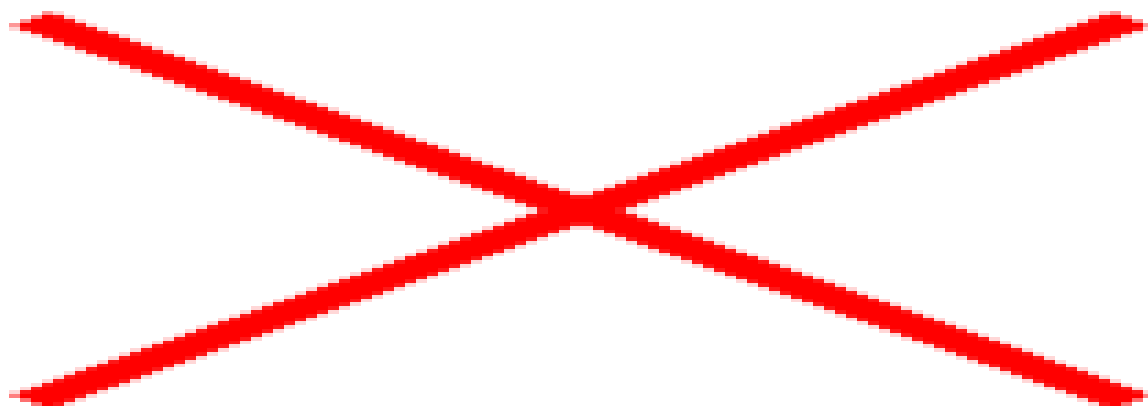
Figura 2 - Área de formação dos respondentes



Fonte: A autora (2019).

Quando questionados quanto a escolha do modelo que deve levar em conta fatores como: cultura da empresa, nível de maturidade dos processos, natureza da aplicação e do projeto, entre outros. Dentre os modelos de desenvolvimento de software abaixo quais você mais utiliza? 18 deles responderam que utilizam o Modelo Incremental, 17 respondentes utilizam o modelo Iterativo, 8 utilizam o modelo Cascata, 6 utilizam o modelo Espiral 4 utilizam Prototipação, 4 utilizam Transformação, 4 Processo Unificado da Rational – RUP, 4 Arquitetura Dirigida pelo Modelo – MDA, 3 Baseado em Componentes, 3 Programação Extrema – XP, 3 Praxis, 2 Evolutivo, 2 Em V, conforme apresentado na figura 3.

Figura 3 - Modelos de desenvolvimento mais utilizados no Tocantins



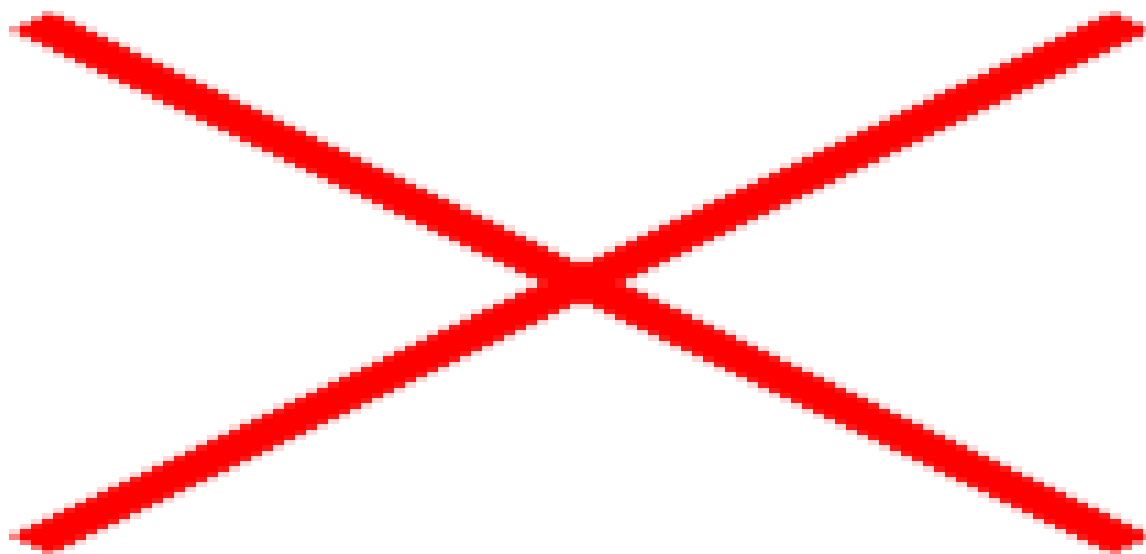
Fonte: A autora (2019).

Ao analisar o gráfico da figura 3, pode-se perceber que a soma dos números das respostas contabiliza 78, enquanto que o número de participante são de apenas de 61, isso ocorre devido a questão 1 do formulário da pesquisa ser de múltiplas escolhas, ou seja o participante poderia escolher mais de uma opção, portanto a pesquisa mostrou que vários participantes utilizam mais de um modelo de processo de desenvolvimento de software.

Isso só confirma o que a literatura apresenta no item 2.3 do Referencial Teórico onde, Santos Silva (2013) diz: "...as duas metodologias podem ser utilizadas quando a empresa entende qual a mais produtiva para os seus projetos".

Quando indagados na pergunta 3, que se conheciam algum modelo que não foi citado na pergunta 1? Se sim, qual (is)? 10% dos respondentes informaram que conhecem o modelo Scrum conforme apresentado na figura 4.

Figura 4 - Modelos que não foram citados na pesquisa



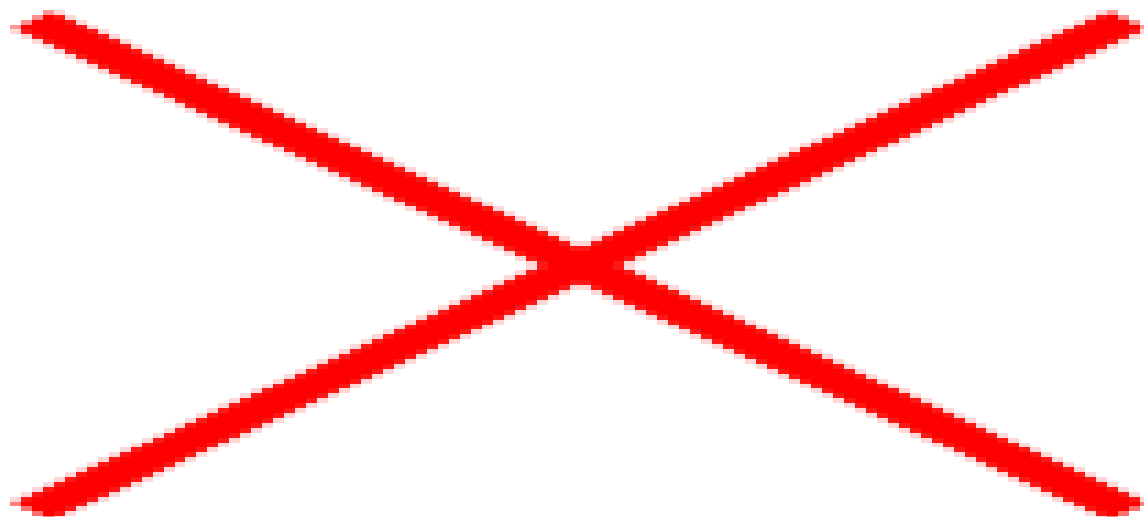
Fonte: A autora (2019).

Ainda foram questionados na pergunta 5 desta pesquisa, para informarem qualquer outra consideração que julgasse ser relevante, porém ninguém se manifestou nessa questão.

A fim de comparar a atual tendência do mercado, sobre os modelos de desenvolvimento de software com a disciplina de Engenharia de Software do Campus Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia, dividiu se os modelos em dois grupos, de acordo Pressman (2011), Ágeis e Tradicionais. Conforme

apresentado no gráfico da figura 5, observa-se que 69% dos respondentes utilizam modelos tradicionais, enquanto 31% utilizam modelos ágeis.

Figura 5 - Modelos Tradicionais x modelos Ágeis

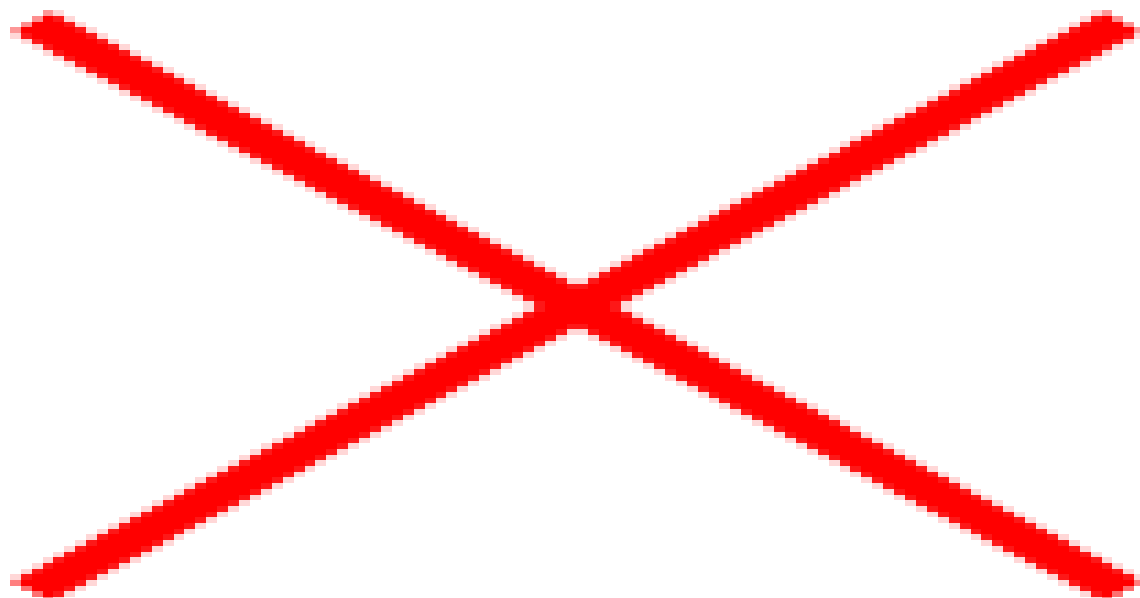


Fonte: A autora (2019).

A pesar da literatura pesquisada mostrar um crescimento das metodologias ágeis, principalmente através do método Scrum, mesmo assim, os modelos tradicionais ainda são usados pela maioria dos respondentes desta pesquisa.

Ainda, a fim de uma melhor precisão nos resultados dessa pesquisa separou-se os modelos descritos no Plano de Ensino dos modelos pesquisados e analisando o gráfico da figura 6 a seguir, comparando-o com o gráfico da figura 3, percebe-se que a ementa não contempla todos os modelos pesquisados e que apenas 1 dos modelos (Transformação), pertence aos grupos de desenvolvimentos Ágeis.

Figura 6 - Modelos descrito na ementa / plano de ensino



Fonte: A autora (2019).

Observa-se, portanto que a ementa do curso de Licenciatura em computação do IFTO de Porto Nacional – TO. atende parcialmente a tendência quanto ao uso dos modelos de processos de desenvolvimento de Software.

Conclusões e considerações

Este trabalho propôs verificar se o egresso da disciplina de Engenharia de Software do curso de licenciatura em Computação do campus Porto Nacional, do IFTO Porto Nacional - TO. está atualizado, conforme a atual tendência do mercado quanto aos modelos de desenvolvimento de software utilizados.

Portanto, a problemática da pesquisa foi respondida, a hipótese levantada foi validada, pois, a pesquisa mostrou que a ementa da disciplina de Engenharia de Software do PPP do curso de Licenciatura em Computação do IFTO Porto Nacional - TO. atende parcialmente a tendência quanto ao uso dos modelos de processos de desenvolvimento de Software.

Ainda, todos os objetivos deste trabalho foram alcançados pois identificou-se os modelos de desenvolvimento de software; verificou junto as especialistas tocantinenses as utilizações dos modelos de processo de desenvolvimento de software e ainda Comparou a atual tendência do mercado sobre os modelos de desenvolvimento de software com a disciplina de Engenharia de Software do Campus Porto Nacional do Instituto Federal de Educação, Ciência e Tecnologia.

Nesse sentido, a contribuição desta pesquisa fica evidente, pois, a atual ementa do PPP vigente do curso de licenciatura em computação do Campus Porto Nacional do IFTO, apresenta de forma geral a utilização dos modelos de desenvolvimento de software, descrito no anexo – A, Plano de Ensino da Disciplina de Engenharia de Software, que são: Cascata, Prototipação, Evolutivo, Incremental, Transformação e Espiral.

Além desses descrito a cima, a pesquisa apresentou mais sete modelos que são: Iterativo, Baseado em Componentes, Programação Extrema – XP, Processo Unificado da Rational – RUP, Arquitetura Dirigida pelo Modelo – MDA, Desenvolvimento Rápido de Aplicação – RAD e Em V. A pesquisa mostrou ainda, que 10% dos respondentes conhecem um outro modelo chamado Scrum, que segundo a literatura é o modelo que segue a tendência da contemporaneidade.

Portanto, a contribuição dessa pesquisa é sugerir a atualização da ementa e plano de ensino, acrescentando os modelos que não são contemplados na disciplina, principalmente o modelo Scrum.

Todavia a temática desse estudo se torna relevante pois, apresentou uma abordagem multidisciplinar ao analisar a tendência do mercado atual quanto ao uso dos modelos de desenvolvimento de software, uma vez que, trata-se de um tema pouco explorado pelo universo acadêmico e apresentou uma melhoria junto a disciplina de Engenharia de Software do IFTO porto Nacional – TO. Com isso, o egresso será melhor qualificado para o mercado de trabalho.

Assim, o presente trabalho ganha ênfase, pois dentro da literatura pesquisada não há outro igual ou que se assemelhe com esse, quanto a sua proposta, tema, problema e objetivos.

Por fim, este trabalho como um todo propiciou grande experiência a autora, a principal razão é o envolvimento e foco em todas as etapas de desenvolvimento e escrita do projeto. Com o foco no desenvolvimento, foi possível colocar em prática todos os conceitos levantados durante a pesquisa bibliográfica bem como as habilidades adquiridas nas disciplinas no decorrer do curso.

Trabalhos Futuros

Como trabalhos futuros a autora pretende desenvolver um artigo sobre o uso do modelo de processo de software Scrum, modelo este, apresentado pelos respondentes desta pesquisa e apontado pela literatura como sendo a tendência da atualidade para o futuro.

Pretende se também acompanhar essa tendência e observar até quando é viável utilizar-se dos modelos de processos de software tradicionais e modelos de processos de software ágeis, em cada projetos.

A autora pretende ainda aprofundar seus estudos na área de engenharia de software, realizando uma pós-graduação/mestrado nesse segmento, pois pretende se desenvolver softwares e apps, gerenciar projetos ligados aos softwares, arquitetar o design estrutural dos programas, realizar testes nos sistemas e gerenciar bancos de dados.

Referencias

VINICIUS. **Processo de teste ágil x tradicional**. Portal DevMedia, 2016. Disponível em: <<https://www.devmedia.com.br/processo-de-teste-agil-x-tradicional/36854>>. Acesso em: 19 de jun. de 2019.

DOS SANTOS SILVA, Daisy Eliana; DE SOUZA, Ingredy Thaís; CAMARGO, Talita. **Metodologias Ágeis Para O Desenvolvimento De Software: Aplicação E O Uso Da Metodologia Scrum Em Contraste Ao Modelo Tradicional**

De Gerenciamento De Projetos. Revista Computação Aplicada-UNG-Ser, v. 2, n. 1, p. 39-46, 2013.

SOMMERVILLE, Ian. **Engenharia de Software.** Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.

PRESSMAN, R. **Engenharia de Software: Uma Abordagem Profissional.** 7ª Ed., São Paulo: McGraw-Hill, 2011.

Sommerville, Ian. **Engenharia de Software.** – São Paulo, Brasil: Editora PEARSON, 2003, 353p.

BECK, Kent. **Manifesto for agile development.** 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 27 jun. 2019.

MACHADO, Felipe Nery Rodrigues; ABREU, Mauricio Pereira de. **Projeto de Banco de Dados: uma visão prática.** 16 ed. São Paulo: Érica, 2009.

PORTAL GAEA. **Metodologia Ágil x Metodologia Tradicional – entenda as diferenças.** maio/2018. Disponível em: <https://gaea.com.br/metodologia-agil-x-metodologia-tradicional-entenda-as-diferencas/>. Acesso em: 20 de jun. 2019.

LEFFINGWELL, D. **Scaling software agility: best practices for large enterprises.** [s.l.] Pearson Education, 2007.

BOEHM, B. W. **Get ready for agile methods, with care.** IEEE Computer, v. 35, p. 64–69, 2002.

LEE, S.; YONG, H.-S. **Distributed agile: project management in a global environment.** Empirical Software Engineering, v. 15, n. 2, p. 204–217, 2010. Disponível em: <<http://link.springer.com/10.1007/s10664-009-9119-7>>.

DUARTE, Luiz. **9 vantagens de usar Scrum em seus projetos,** 2017. Disponível em: <https://www.luiztools.com.br/post/9-vantagens-de-usar-scrum-em-seus-projetos/>. Acesso em: 20 de jun. de 2019.

CAMARGO, Robson. **Metodologia Agile: quais as mais utilizadas?** 2018. Disponível em: <https://robsoncamargo.com.br/blog/Metodologia-Agile-quais-as-mais-utilizadas>. Acesso em: 19 de jun. de 2019.

PORTAL SIGNIFICADOS. **Significado de Ementa,** 2016. Disponível em:<<https://www.significados.com.br/ementa/>>. Acesso em 23 de jun de 2019.

PADILHA, R. P. **Planejamento dialógico: como construir o projeto político pedagógico da escola.** São Paulo: Cortez; Instituto Paulo Freire, 2001.

FONSECA, J. J. S. **Metodologia da pesquisa científica.** Fortaleza: UEC, 2002. Apostila.

ANEXO - A Plano de Ensino da disciplina de Engenharia de Software



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO
TOCANTINS
CAMPUS PORTO NACIONAL
GERÊNCIA DE ENSINO
COORDENAÇÃO DO CURSO SUPERIOR DE LICENCIATURA EM
COMPUTAÇÃO

PLANO DE ENSINO ENGENHARIA DE SOFTWARE

Componente: **Engenharia de Software**
Carga horária **Total: 80 Carga horária Teórica: 40 Carga horária Prática:**

40

Ano/Semestre: **2019/1 Período: 6º**

Professor(a): **Rafael Miranda Correia** Curso: **Licenciatura em Computação**

1. EMENTA:

Introdução à Engenharia de Software: Conceitos e definições. Sistemas Computacionais. O que é software? O que é engenharia de software? O Ciclo de Vida do Software. Qualidade de Software. Processo de Software: Modelos de processo: Cascata, Prototipação, Evolutivo, Incremental, Transformação, Espiral. Slides. Modelos de processo: Processo Unificado.

Atividades do Processo: Especificação, Design e implementação, Validação e Evolução.

Métricas, Planejamento e

Gerenciamento de Software: Elaboração do cronograma. Planejamento da equipe. Estimativas e Métricas. Análise de riscos. Requisitos de Software: Requisitos e Engenharia de Requisitos. Definindo Requisitos com Casos de Uso. Slides. Modelos de Software. Design de Software: Design Conceitual, Prototipação. Arquitetura de Software conceitos, visão tradicional e visão emergente. Visões arquiteturais. Linguagens de Descrição Arquitetural. Padrões de Projeto. Framework. Verificação e Validação de Software: Formas de verificação e validação de programas. Técnicas de testes. Manutenção e Evolução de Software.

6. COMPETÊNCIAS:

Apresentar, analisar e discutir o corpo de conhecimento que constitui a Engenharia de

Software, seus princípios, métodos e ferramentas. Identificar as técnicas da Engenharia de Software para o desenvolvimento de um produto; Ser capaz de

descrever os métodos, ferramentas e procedimentos associados; Conhecer e utilizar as técnicas de trabalho em grupo, especificamente para desenvolvimento de softwares; Saber identificar os princípios da ética profissional do engenheiro de software.

7. HABILIDADES:

Durante o desenvolvimento desta disciplina o aluno terá a oportunidade de aprender os conceitos e praticas da engenharia de software e gerencia de projetos. Com isso, será capacitado a fazer análise documentada de um softwares por completo e também gerenciar o desenvolvimento de softwares.

8. BASES TECNOLÓGICAS:

4.1. Introdução à Engenharia de Software:

4.1.2. Conceitos e definições.

Entender os conceitos básicos de engenharia de software 4.1.3. Sistemas Computacionais.

Entender os tipos de sistemas computacionais 4.1.4. O que é software?

Entender o que é um software

4.1.5. O que é engenharia de software? Entender o que é a engenharia de software 4.1.6. O Ciclo de Vida do Software.

Entender o funcionamento de cada um dos ciclos de vida de um software

4.1.7. Qualidade de Software. Compreender o a qualidade de software 4.2. Processo de Software:

4.2.5. Modelos de processo:

4.2.6. Cascata, Prototipação,

Entender o tipos de processo de software cascata e prototipação

4.2.7. Evolutivo, Incremental,

Entender o tipos de processo de software evolutivo e incremental

4.2.8. Transformação,

Entender o tipos de processo de software transformação 4.2.5. Espiral.

Entender o tipos de processo de software espiral 4.3. Modelos de processo:

4.3.1. Processo Unificado. Compreender o processo unificado 4.4.

Atividades do Processo:

4.4.3. Especificação,

Entender a especificação de processo

4.4.4. Design e implementação,

Entender sobre design e implementação de processos 4.4.3. Validação e Evolução.

Entender como fazer a validação e evolução 4.4.4. Métricas,

Entender as métricas.

8.5. Planejamento e Gerenciamento de Software:

8.5.1. Elaboração do cronograma. Aprender a elaborar cronograma
 4.5.2. Planejamento da equipe. Entender sobre planejamento de equipe 4.5.3.
 Estimativas e Métricas. Entender as estimativas e métricas 4.5.4. Análise de riscos.

Fazer a análise dos riscos

8.6. Requisitos de Software:

8.6.1. Requisitos e Engenharia de Requisitos. Aprender a identificar os
 requisitos de softwares 4.6.2. Definindo Requisitos com Casos de Uso. Utilizar a uml
 para definir os casos de uso 4.6.3. Modelos de Software. Entender sobre os modelos
 de software 4.7. Design de Software:

4.7.1. Design Conceitual, Entender sobre o design conceitual 4.7.2.
 Prototipação.

Entender sobre prototipação

4.7.3. Arquitetura de Software conceitos,

Entender os conceitos da arquitetura de software 4.7.4. visão tradicional e
 visão emergente.

Entender sobre a visão tradicional e a visão emergente.

4.7.7. Visões arquiteturais.

Entender sobre as visões de arquitetura

4.7.8. Linguagens de Descrição Arquitetural. Entender sobre as
 linguagens de descrição da arquitetura 4.7.7. Padrões de Projeto.

Entender sobre os padrões de projetos 4.7.8. Framework.

Entender sobre framework

4.7.11. Verificação e Validação de Software:

4.7.12. Formas de verificação e validação de programas. Entender as
 formas de verificação e validação de programas 4.7.11. Técnicas de testes.

Entender as técnicas de testes

4.7.12. Manutenção e Evolução de Software. Entender sobre a manutenção
 e evolução de software

9. METODOLOGIA:

Aula teórica e prática através de exposição dialogada utilizando recurso
 como pincel, quadro, material impresso, data show, hardwares e computador.
 Realização de exercícios e trabalhos sobre os conceitos apresentados.

Aulas expositivas

- Aulas práticas e demonstrativas
- Exercícios práticos
- Textos Técnicos

9. RECURSOS DIDÁTICOS:

Pincel, quadro, material impresso, data show, hardwares e computadores

10. AVALIAÇÃO:

1º Bimestre uma prova valendo 7,0 e um trabalho valendo 3,0 assim totalizando 10,0 2º Bimestre uma prova valendo 7,0 e um trabalho valendo 3,0 assim totalizando 10,0

11. BIBLIOGRAFIA:

11.1 Básica

BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio Janeiro: Campus, 2006. BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. UML: Guia do Usuário. 2. ed.

Rio Janeiro: Campus, 2006. PAULA FILHO, W. P. Engenharia de software: fundamentos, métodos e padrões. Rio de Janeiro: LTC, 2003. REFERÊNCIAS BIBLIOGRÁFICAS PFLEEGER, S. L. Engenharia de software: teoria e prática. São Paulo: Pearson, 2004.

PRESSMAN, R. Engenharia de software. Rio de Janeiro: MacGraw-Hill, 2006.

SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Addison Wesley, 2007.

11.2 Complementar

FURLAN, J. D. Modelagem de objetos através da UML. São Paulo: Makron Books, 1998. LARMAN, C. Utilizando UML e padrões: um guia para a análise e projeto orientados a objetos. 3. Porto Alegre: Editora Bookman, 2007.

Rafael Miranda Correia

Coordenador do Curso Superior de Licenciatura em Computação

Portaria nº 109/2016/IFTO/ Campus Porto Nacional



Documento assinado eletronicamente por **Rafael Miranda Correia, Coordenador**, em 21/02/2019, às 21:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.toc.gov.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o

código verificador **0598054** e o código CRC **E47A8E30**.

Av. Tocantins, Loteamento Mãe Dedé

Setor - Jardim América

Referência: Processo nº 23337.001925/2019-90 SEI nº 0598054

Submetido em xx/xx/xxxx.

Aceito em xx/xx/xxxx.

