CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
UNIDADE DE PÓS-GRADUAÇÃO, EXTENSÃO E PESQUISA
MESTRADO PROFISSIONAL EM GESTÃO E TECNOLOGIA EM SISTEMAS PRODUTIVOS

FLÁVIO AUGUSTO SILVA

AVALIAÇÃO DO USO E ACEITAÇÃO DE UM PROCESSO CORPORATIVO DE DESENVOLVIMENTO ÁGIL DE SOFTWARE NO SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS - SERPRO

São Paulo
Junho/2020

FLÁVIO AUGUSTO SILVA

AVALIAÇÃO DO USO E ACEITAÇÃO DE UM PROCESSO CORPORATIVO DE DESENVOLVIMENTO ÁGIL DE SOFTWARE NO SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS - SERPRO

Dissertação apresentada como exigência parcial para a obtenção do título de Mestre em Gestão e Tecnologia em Sistemas Produtivos do Centro Estadual de Educação Tecnológica Paula Souza, no Programa de Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos, sob a orientação da Profa. Dra. Marília Macorin de Azevedo

São Paulo
Junho/2020

FICHA ELABORADA PELA BIBLIOTECA NELSON ALVES VIANA
FATEC-SP / CPS CRB8-8390

Silva, Flávio Augusto
S586a Avaliação do uso de aceitação de um processo corporativo de desenvolvimento ágil de software no Serviço Federal de Processamento de Dados – SERPRO / Flávio Augusto Silva. – São Paulo: CPS, 2020.
115 f. : il.

Orientadora: Profa. Dra. Marília Macorin de Azevedo
Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). – Centro Estadual de Educação Tecnológica Paula Souza, 2020.

1. Sistemas Produtivos. 2. Engenharia de software. 3. Métodos ágeis. 4. Desenvolvimento de Software Lean. 5. UTAUT. I. Azevedo, Marília Marcorin de. II. Centro Estadual de Educação Tecnológica Paula Souza. III. Título.

FLÁVIO AUGUSTO SILVA

AVALIAÇÃO DO USO E ACEITAÇÃO DE UM PROCESSO CORPORATIVO DE DESENVOLVIMENTO ÁGIL DE SOFTWARE NO SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS - SERPRO

Profa. Dra. Marília Macorin de Azevedo
Centro Estadual de Educação Tecnológica Paula Souza

Prof. Dr. Mauro De Mesquita Spinola
Escola Politécnica da Universidade de São Paulo

Prof. Dr. Napoleão Verardi Galegale
Centro Estadual de Educação Tecnológica Paula Souza

São Paulo, 10 de junho de 2020

A minha esposa e filhos pela paciência e apoio.

AGRADECIMENTOS

Gostaria de agradecer à minha família pelo apoio, incentivo e compreensão pelo tempo necessário para concluir esta jornada.

Agradeço à professora Marília pelas orientações e sugestões para que este trabalho fosse realizado.

Agradeço também ao prof. Giordano pelas orientações no desbravamento do mundo estatístico.

Também agradeço aos membros da banca, prof. Dr. Mauro De Mesquita Spinola pelas sugestões sempre pertinentes para que o trabalho fosse concluído da melhor maneira possível e ao Prof. Dr. Napoleão Verardi Galegale pela sugestão de uso do modelo UTAUT e por compartilhar os “causos” tecnológicos, sempre muito interessantes.

Por fim, agradeço ao SERPRO por permitir que eu realizasse o curso no período laboral e fizesse a pesquisa no ambiente corporativo.

RESUMO

Silva, F. A. **Avaliação do uso de aceitação de um processo corporativo de desenvolvimento ágil de software no Serviço Federal de Processamento de Dados – SERPRO.** 115 f. Dissertação (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2020.

A adoção de um processo de desenvolvimento corporativo permite uma uniformidade no desenvolvimento de software dentro da organização e facilita a comunicação entre equipes de regiões diferentes. O presente trabalho teve como objetivo explorar o nível de uso e aceitação de um processo ágil de desenvolvimento de software corporativo (UNIFICA) no SERPRO, a partir do ponto de vista dos desenvolvedores e gestores. O estudo se baseou na Teoria Unificada de Aceitação de Uso da Tecnologia (UTAUT) de Venkatesh et al (2003) para avaliar fatores antecedentes que podem levar a intenção e ao comportamento de uso do UNIFICA. Por meio de uma pesquisa de abordagem mista, exploratória e descritiva, utilizando-se de uma *Survey*, foi possível coletar dados de 44 respondentes, usuários do UNIFICA de diferentes regiões do Brasil, e avaliar a relação dos constructos do modelo de pesquisa. Os resultados obtidos confirmaram a influência dos fatores antecedentes Expectativa de Esforço e Influência Social na intenção de uso do UNIFICA. O estudo concluiu que deve haver um trabalho junto às pessoas com menos de 10 anos de trabalho no SERPRO para aumentar o grau de concordância em relação aos constructos Expectativa de Desempenho e Expectativa de Esforço. O estudo também permitiu a criação de um modelo de avaliação do uso e aceitação do UNIFICA que pode ser utilizado em diferentes equipes de desenvolvimento de software.

Palavras-chave: Sistemas Produtivos; Engenharia de *software*; Métodos Ágeis; Desenvolvimento de Software *Lean;* UTAUT

ABSTRACT

Silva, F. A. **Evaluation of the use of acceptance of a corporate process of agile software development at Serviço Federal de Processamento de Dados – SERPRO.** 115 f. Dissertation (Mestrado Profissional em Gestão e Tecnologia em Sistemas Produtivos). Centro Estadual de Educação Tecnológica Paula Souza, São Paulo, 2020.

The adoption of a corporate development process allows for a standardized software development within the organization, and facilitates communication among teams from different regions. This work aimed at exploring the level of use and acceptance of an agile enterprise software development process (UNIFICA) at SERPRO, from the point of view of developers and managers. The study was based on the Unified Theory of Acceptance of Use of Technology (UTAUT) by Venkatesh et al (2003) to evaluate previous factors that can lead to UNIFICA's intention and usage behavior. Through a mixed, exploratory and descriptive research, using a Survey, it was possible to collect data from 44 respondents, users of UNIFICA, from different regions of Brazil, and to evaluate the relationship of the constructs of the research model. The results obtained confirmed the influence of the previous factors Expectation of Effort and Social Influence on the intention to use UNIFICA. The study concluded that there should be a work with people who have been working for SERPRO for fewer than 10 years in order to increase the degree of agreement in relation to the Expectations of Performance and Expectations of Effort. The study also allowed the creation of a model to evaluate the use and acceptance of UNIFICA, that can be used in different software development teams.

Keywords: Productive Systems; Software Engineering; Agile Methods; Lean Software Development; UTAUT

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AC	Autoridade Certificadora
CMM	*Capability Maturity Model*
CMMI	*Capability Maturity Model Integration*
C-TAM-TPB	*Combined TAM and TPB Model* - Modelo Combinado TAM-TPB
ICP Brasil	Autoridade Certificadora da Infraestrutura de Chaves Pública Brasileira
IDT	*Innovation Diffusion Theory* - Teoria da Difusão da Inovação
MM	*Motivation Model* - Modelo Motivacional
MPCU	*Model of Personal Computer Utilization* - Modelo de Utilização de Computador Pessoal
MUSE	Metodologia Única de Serviços
PMBoK	*Project Management Body of Knowledge*
PMod	Processo de Modernização do Desenvolvimento
RUP	*Rational Unified Process*
SCT	*Social Cognitive Theory* - Teoria Social Cognitiva
SERPRO	Serviço Federal de Processamento de Dados
SW-CMM	*Software Engineering Institute - Software Capability Maturity Model*
TAM	*Technology Acceptance Model* - Modelo de Aceitação da Tecnologia
TAM-2	*Extension of the Technology Acceptance Model* - Extensão do Modelo de Aceitação de Tecnologia
TRA	*Theory of Reasoned Action* - Teoria da Ação Racional
TPB	*Theory of Planned Behavior* - Teoria do Comportamento Planejado
UTAUT	*Unified Theory of Acceptance and Use of Technology* - Teoria Unificada de Aceitação de Uso da Tecnologia
XP	*Extreme Programming*

SUMÁRIO

INTRODUÇÃO

O projeto de desenvolvimento de *software* é uma atividade complexa e exige uma série de habilidades por parte dos desenvolvedores e gestores para ser bem executado. Brooks (1995) afirma que a complexidade é uma característica essencial do *software*, e essa complexidade leva à dificuldade de comunicação entre os membros da equipe. Isso implica em falhas, custos excessivos e atrasos no cronograma.

A complexidade do software aumenta proporcionalmente com o tamanho do projeto de *software.* No *CHAOS report* de 2015 (STANDISH GROUP, 2015) é possível observar que a taxa de sucesso, que leva em consideração prazo, escopo, custo e satisfação do cliente, para projetos pequenos é de 61%, 24% para projetos moderados, 12% para projetos médios, 11% para projetos grandes e 6% em projetos enormes. Dessa forma, é imprescindível que a equipe de desenvolvimento de *software* busque novas formas de desenvolvimento que tragam benefícios tanto para o projeto de desenvolvimento quanto para os clientes.

O método de desenvolvimento ágil veio para ajudar o time de desenvolvimento melhorando a comunicação e consequentemente a qualidade do *software* entregue. O método de desenvolvimento ágil foi introduzido formalmente em 2001 por meio de um conjunto de quatro valores e doze princípios estabelecidos no manifesto ágil. Nesse tipo de abordagem, o ciclo de desenvolvimento de software é quebrado em pequenos ciclos, cada um contendo as mesmas etapas de desenvolvimento. O objetivo não é entregar um sistema completo com todas as funcionalidades requisitadas ao final do ciclo de desenvolvimento, mas implementar as funcionalidades que apresentam maior valor para os usuários (BECK, 1999).

De acordo com Hoda et al. (2017) a agilidade, ou habilidade de adaptar-se rapidamente à volatilidade dos requisitos, é um dos pilares do desenvolvimento ágil. Essa forma de desenvolvimento contrasta fortemente com as abordagens mais tradicionais de desenvolvimento, como o modelo de desenvolvimento cascata (HODA et al., 2017).

Rigby, Sutherland e Takeuchi (2016) afirmam que a abordagem de gerenciamento ágil traz uma série de vantagens ao projeto, como: aumento de produtividade, melhoria de qualidade do *software*, melhoria de satisfação do cliente, aumento de colaboração entre a equipe e aumento da motivação dos membros da equipe. A colaboração estimula a interação entre os membros da equipe e consequentemente melhora a comunicação entre eles.

Ao longo do tempo, os benefícios trazidos pelos métodos ágeis foram largamente estudados e documentados. No estudo de Hoda et al. (2017) foram identificadas 28 revisões

sistemáticas da literatura e estudos de mapeamento (RSL) no período de 1990 até 2015. Segundo Hoda et al. (2017) os trabalhos de RSL possuíam foco em dez áreas diferentes de pesquisa em desenvolvimento de software ágil: adoção, métodos, práticas, aspectos humanos e sociais, CMMI, usabilidade, engenharia de software global, agilidade organizacional, sistemas embarcados e engenharia de linha de produtos de *software*.

Apesar dos benefícios trazidos pelos métodos ágeis, podem ocorrer dificuldades de adoção em organizações grandes e complexas (DYBÅ e DINGSØYR, 2008). De acordo com Dikert, Paasivaara e Lassenius (2016), o método de desenvolvimento ágil foi facilmente adotado em organizações pequenas enquanto organizações maiores, já estabelecidas, têm tido grandes desafios tentando ajustar as práticas já existentes na organização aos métodos ágeis. A mudança no processo de tomada de decisão das organizações é difícil e frequentemente notam-se colaboradores resistentes às mudanças (DIKERT, PAASIVAARA, e LASSENIUS, 2016).

A aceitação dos métodos ágeis de desenvolvimento pelos colaboradores permite que a organização lide melhor com o ambiente dinâmico de negócios (CHAN e THONG, 2009).

Este estudo tem, portanto, como questão de pesquisa: sob o ponto de vista dos desenvolvedores e gestores, qual é o nível de aceitação de um processo corporativo de desenvolvimento ágil de *software*?

O objetivo do presente trabalho é explorar uso e aceitação de um processo ágil de desenvolvimento de *software* corporativo em um ambiente específico - no SERPRO, a partir do ponto de vista dos desenvolvedores e gestores.

A escolha do SERPRO se justifica por se tratar de uma empresa líder de mercado de TI para o setor público com presença nacional e ampla experiência em grandes sistemas da Administração Pública Federal como, por exemplo, Siscomex, RAIS, Renach, Renavam, Siafi, Siape e IRPF.

O trabalho tem como objetivos específicos:

- Descrever os métodos ágeis *Scrum*, XP, Kanban e Desenvolvimento *Lean*
- Identificar modelo de aceitação de tecnologia
- Identificar o nível de aceitação do processo corporativo de desenvolvimento de *software* pelos desenvolvedores e gestores, baseado no modelo de aceitação selecionado

O presente trabalho está dividido da seguinte forma:

No capítulo 1, é apresentado o referencial teórico a respeito de processo de desenvolvimento de *software*.

No capítulo 2, é apresentado o referencial teórico sobre o modelo de aceitação de tecnologia.

O capítulo 3 apresenta o ambiente de pesquisa e o método utilizado na pesquisa, bem como os procedimentos metodológicos para o desenvolvimento da pesquisa empírica.

O capítulo 4 apresenta as análises e os resultados da pesquisa.

Então, são apresentadas as considerações finais.

1 DESENVOLVIMENTO DE *SOFTWARE*

Software é uma instrução (programa de computador) que, quando executada, fornece características, funções e performances desejadas, estrutura de dados que permitem os programas manipularem informação adequadamente e informações descritivas armazenadas em meios físicos e virtuais que descrevem a operação e uso dos programas (DEMARCO,1995).

Segundo Pressmam e Maxin (2016), os *softwares* podem ser classificados em sete categorias:

a) *Software* de sistema: uma coleção de *softwares* criados para servirem outros programas. Os compiladores, por exemplo, são *softwares* de sistema que utilizados por outros programas para processar informações complexas e estruturadas.

b) *Software* de aplicação: um conjunto de programas isolados que resolvem problemas específicos de um determinado negócio. Os aplicativos dessa categoria processam dados de negócio ou dados técnicos de modo a facilitar a operação de negócio ou ajudar o processo de tomada de decisão.

c) Aplicativos de engenharia e científicos: programas computacionais de processamento numérico, voltados para aplicações científicas.

d) Aplicativos embarcados: programas implementados residentes em produtos e sistemas. Possuem funções limitadas: são utilizados para controlar e limitar funções para usuários finais.

e) Aplicativos de prateleira: programas desenvolvidos para resolver problemas específicos e que são consumidos por clientes em massa.

f) Aplicativos Web/Mobile: programas baseados na rede que englobam aplicações baseadas em navegadores e aplicativos para telefone celular.

g) Aplicativos de inteligência artificial: programas que utilizam algoritmos não numéricos utilizados para resolver problemas complexos.

Cada categoria de programas apresentada exige uma habilidade específica do programador para ser desenvolvida e ao mesmo tempo compartilha de um conjunto de técnicas que são recomendadas para que o desenvolvimento de *software* seja feito com qualidade,

escopo, prazo e custo esperados. Esse conjunto de técnicas é chamado de Engenharia de *Software*.

Boehm (1976) define Engenharia de *Software* como a aplicação prática de conhecimento científico no planejamento e na construção de programas de computador, bem como a documentação associada necessária para desenvolver, operar e manter tais programas. O conhecimento científico pode ser organizado em modelos para facilitar sua utilização.

Segundo Gao (2010), o modelo de desenvolvimento de *software* é uma das partes mais importantes da Engenharia de *Software*, pois descreve o *framework* estrutural de todas as tarefas e processos no desenvolvimento de *software*. A escolha correta do modelo de processo de desenvolvimento de *software* é importante para o sucesso do projeto de *software* (VIJAY ASARATHY E BUTLER ,2016; GAO, 2010).

Boehm (1988) afirma que as principais funções de um modelo de processo de *software* são determinar a ordem ou evolução dos estágios envolvidos no desenvolvimento de *software* e estabelecer os critérios de transição para se progredir de um estágio para o outro. Assim, os modelos incluem critérios de conclusão para o estágio atual, critérios condicionais de escolha e critérios de entrada para progressão ao próximo estágio.

Pressmam e Maxin (2015) definem cinco atividades estruturais que podem ser aplicadas a todos os projetos de *software*:

a) Comunicação: Antes de iniciar o trabalho é importante entender as necessidades do cliente e das partes interessadas no projeto. Dessa forma, a comunicação e a colaboração são importantes para o sucesso do projeto.

b) Planejamento: O planejamento do projeto de *software* serve para auxiliar a equipe de desenvolvimento na execução do mesmo, pois descreve atividades técnicas que devem ser realizadas, riscos identificados, recursos que serão requeridos, produtos de trabalho que devem ser elaborados e um cronograma do trabalho.

c) Modelagem: A modelagem de *software* auxilia o engenheiro de *software* a entender melhor os requisitos de *software* e o design que irá atender esses requisitos.

d) Construção: A etapa de codificação envolve a codificação do que foi elaborado na fase de modelagem, bem como os testes do código para identificar eventuais erros de implementação.

e) Entrega: Nesta etapa, o *software* é entregue ao cliente. Este avalia o produto entregue e fornece *feedback* para a equipe de desenvolvimento.

Os mesmos autores afirmam que, além das atividades estruturais, existem atividades de apoio típicas que ajudam a equipe de desenvolvimento a gerenciar e controlar o progresso, os riscos, a qualidade e as mudanças do projeto de *software*. Essas atividades são:

a) Controle e acompanhamento do projeto: permite que o time de desenvolvimento afira o progresso do projeto de acordo com o que foi planejado inicialmente.

b) Gerenciamento de riscos: avalia os riscos que podem afetar o progresso do projeto de *software*.

c) Garantia da qualidade de *software*: define e conduz atividades requeridas para garantir a qualidade do *software*.

d) Revisões técnicas: avaliam os produtos de engenharia de *software* com a intenção de evitar a propagação de erros para as atividades seguintes.

e) Medição: define e coleta métricas (de processo, do projeto e do produto) que auxiliarão a equipe de desenvolvimento a verificar se o *software* a ser entregue está de acordo com as necessidades das partes interessadas.

f) Gerenciamento e configuração de *software*: gerencia os efeitos das mudanças em artefatos desenvolvidos ao longo do projeto de *software*.

g) Gerenciamento de reusabilidade: define critérios para reuso de artefatos (inclusive componentes de *software*) e estabelece mecanismos para buscar componentes reutilizáveis.

h) Preparo e produção de artefatos de *software*: engloba atividades requeridas para criar produtos de trabalho como modelos, documentos, *logs*, formulários e listas.

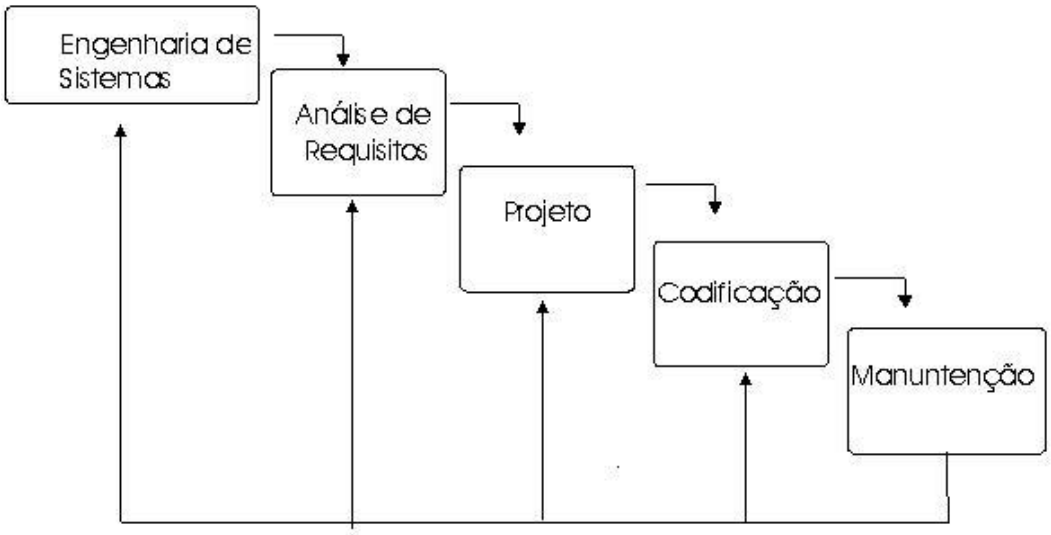
O modelo de *software* é uma representação simplificada de um processo de *software* e cada modelo representa uma perspectiva particular de um processo. Os modelos gerais de

processos a partir da perspectiva de arquitetura são conhecidos como paradigmas de processos. Os paradigmas de processos não são descrições definitivas dos processos, mas sim abstrações que podem ser usadas para explicar diferentes abordagens de desenvolvimento de *software*, e podem ser ampliados e adaptados para criar processos de engenharia de *software* mais específicos (SOMMERVILLE, 2011).

1.1 Modelo Cascata

O modelo do ciclo de vida clássico ou modelo cascata, conforme Figura 1, utiliza uma abordagem sistemática e sequencial ao longo do desenvolvimento do *software*. Geralmente, este ciclo tem início no nível do sistema e avança ao longo de análise, projeto, codificação, teste e manutenção. A utilização deste modelo requer cuidado, pois exige que todas as incertezas estejam resolvidas no início do projeto.

Figura 1 Modelo cascata



Fonte: Adaptado Pressmam e Maxin (2015)

O modelo cascata está dividido nas seguintes etapas:

- Engenharia de Sistema: nesta etapa é realizada a coleta de requisitos em nível de sistema.
- Análise de Requisitos: etapa na qual os requisitos são documentados e revistos com o cliente.
- Projeto: tradução dos requisitos do *software* para um conjunto de representações que podem ser avaliadas quanto a qualidade.

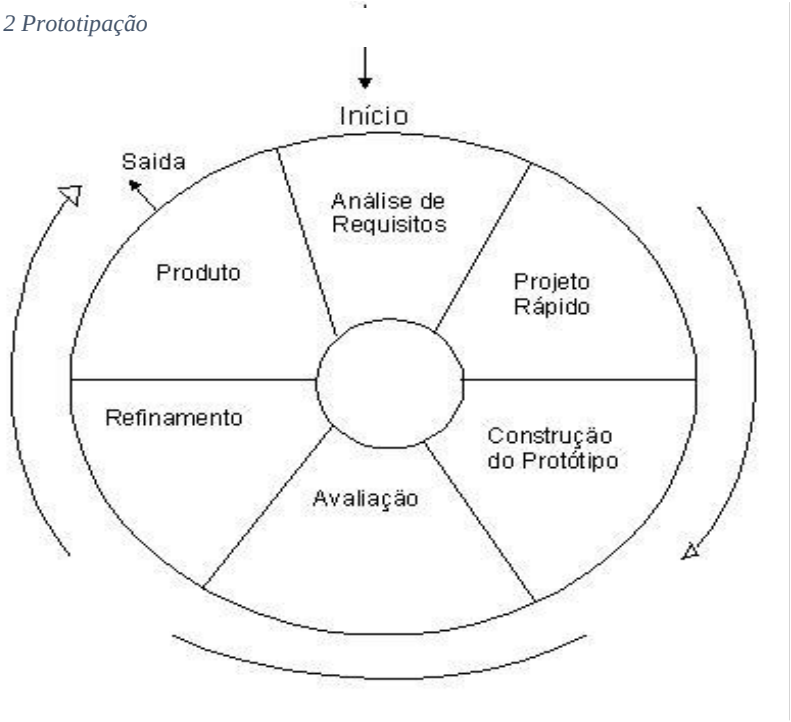
- Codificação: tradução do projeto para uma linguagem que resulte em instruções executáveis pelo computador.
- Manutenção: etapa responsável por efetuar mudanças no *software* depois que este for entregue ao cliente. As mudanças podem ocorrer devido a identificação de erros, mudanças no ambiente externo, acréscimos funcionais ou melhorias no desempenho.

O modelo cascata é simples e de fácil compreensão, porém apresenta alguns problemas como por exemplo a demora de entrega de um *software* funcional, que só estará pronto para uso num ponto tardio do cronograma. Outra questão é que projetos reais dificilmente sequem o fluxo sequencial e se erros forem detectados tardiamente o resultado pode ser desastroso (PRESSMAM e MAXIN, 2015).

1.2 Prototipação

A prototipação (Figura 2) é um processo que permite ao desenvolvedor criar um modelo de *software*.

Figura 2 Prototipação



Fonte Adaptado Pressmam e Maxin (2015)

O modelo pode assumir três formas. O primeiro modelo pode ser um protótipo que representa a iteração homem-máquina para facilitar o entendimento do cliente quanto a utilização do produto. O segundo consiste em um protótipo que implemente um subconjunto de funções exigidas pela demanda do cliente a fim de testar algoritmos. O último compreende um

programa que executa parte da função desejada, mas que possui outras características que podem ser melhoradas posteriormente.

Na Figura 2 é possível verificar que existem seis atividades que são definidas no modelo de prototipação:

1. Análise dos requisitos: nessa fase o cliente e desenvolvedor definem os objetivos gerais do *software*. Identificam os requisitos conhecidos e as áreas que necessitam de informações adicionais.
2. Projeto rápido: nessa fase, é feita a representação dos aspectos de *software* que são visíveis ao usuário.
3. Construção do protótipo: nessa fase, é feita a implementação rápida do projeto.
4. Avaliação do protótipo: nessa etapa, o cliente e o desenvolvedor avaliam o protótipo.
5. Refinamento: nessa etapa, o cliente e o desenvolvedor refinam os requisitos do *software* que será desenvolvido.
6. Produto: nessa fase, os requisitos são identificados. O protótipo é descartado e a versão de produção é construída considerando os critérios de qualidade.

Por ser um método rápido, a prototipação pode gerar alguns inconvenientes, como, por exemplo, o cliente achar que o produto já está desenvolvido. Outro erro que pode ocorrer é o desenvolvedor querer entregar o produto em um prazo menor do que o necessário e fazer algumas adaptações no protótipo que podem ser prejudiciais. Neste caso, podem existir grandes erros de implementação que passam despercebidos pelo desenvolvedor que foca no desenvolvimento rápido do projeto.

1.3 Modelo Espiral

O Modelo Espiral foi desenvolvido com base nas características do Modelo Cascata e o Modelo de Prototipação. Além disso, foi acrescentada uma nova etapa que é a análise de riscos. O Modelo Espiral, representado na Figura 3, define quatro atividades importantes:

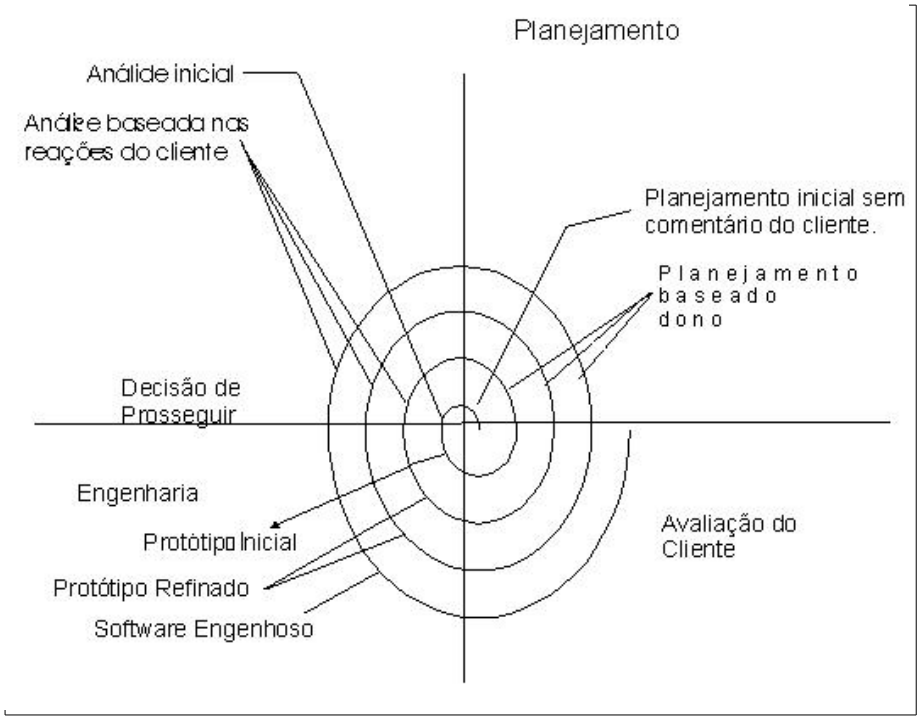
a) Planejamento: determina objetivos, alternativas e restrições;

b) Análise dos Riscos: identifica os riscos e as resoluções para estes;

c) Engenharia: desenvolvimento do produto no nível seguinte;

d) Avaliação: avaliação feita pelo cliente dos resultados da engenharia.

Figura 3 Modelo espiral



Fonte Pressmam e Maxin (2015)

A cada iteração ao redor da espiral, as versões do *software* são mais completas. Se, por exemplo, no primeiro giro for identificada incerteza no requisito, uma prototipação na fase de engenharia será utilizada para ajudar o cliente e o desenvolvedor. Na fase de avaliação, o cliente pode apresentar sugestões para modificações, que serão utilizadas na entrada da fase seguinte, de planejamento e análise de riscos.

O modelo espiral é a abordagem mais realista para o desenvolvimento de sistemas e de *software*. É importante uma grande experiência na avaliação dos riscos, pois um erro nessa fase pode acarretar sérios problemas.

1.4 Métodos Ágeis

Os métodos tradicionais de desenvolvimento de *software* enfatizam o planejamento extensivo e racional. Neste tipo de planejamento, o sucesso depende de dois fatores: a habilidade de prever as mudanças e o desenvolvimento de planos para tratar essas mudanças (DYBÅ, 2000). Segundo o mesmo autor, essa previsibilidade funciona apenas para sistemas simples. No caso dos sistemas complexos, como o desenvolvimento de *software*, as mudanças não previsíveis não são totalmente conhecidas e não são controladas pela equipe de desenvolvimento, portanto, as mudanças devem ser tratadas de outra forma (DYBÅ, 2000).

Em fevereiro de 2001, um grupo formado por 17 profissionais renomados de desenvolvimento de *software* se reuniu para discutir ideias e procurar uma alternativa aos processos burocráticos e às práticas adotadas nas abordagens tradicionais da Engenharia de *Software* e Gerência de Projetos. Dessa reunião surgiu o Manifesto do Desenvolvimento Ágil de *Software*, que define os seguintes valores: Indivíduos e interações são mais importantes que processos e ferramentas; *Software* funcionando é mais importante que documentação completa e detalhada; Colaboração com o cliente é mais importante que negociação de contratos e Adaptação às mudanças é mais importante que seguir um plano (BECK et al, 2001).

A interação entre indivíduos facilita a troca de informações e permite que as mudanças no processo ocorram tempestivamente. O trabalho focado no *software* permite apurar a produtividade e provê um *feedback* rápido, e a interação frequente entre os membros da equipe compensa a documentação mínima. A colaboração com os clientes significa que todos os envolvidos (patrocinadores, clientes, usuários e desenvolvedores) estão na mesma equipe e a colaboração entre eles pode produzir resultados mais apropriados. Os contratos com os clientes são necessários, mas sem colaboração, insuficientes (HIGHSMITH e COCKBURN, 2001).

Uma série de metodologias ágeis foram desenvolvidas para pôr em prática a filosofia ágil. Dentre elas, podemos citar *Scrum*, *eXtreme Programming* (XP), desenvolvimento de *Software Lean* e Kanban.

De acordo com Schwabber e Beedle (2001), a primeira referência do termo *Scrum* na literatura apareceu no artigo de Takeuchi e Nonaka (1986), no qual é apresentado um processo de desenvolvimento de produto originário do Japão, que é adaptativo, rápido, e auto organizado. O termo *Scrum* deriva originalmente de uma estratégia do jogo *rugby* que tem como objetivo a reposição da bola que estava fora do jogo utilizando o trabalho em equipe (SCHWABBER E BEEDLE, 2001).

A abordagem *Scrum* foi desenvolvida para gerenciar o processo de desenvolvimento de sistemas. É um método empírico que aplica ideias da teoria do processo industrial para o desenvolvimento de sistemas, reintroduzindo ideias de flexibilidade, adaptabilidade e produtividade. O *Scrum* não define nenhuma técnica específica de desenvolvimento de *software*, mas concentra-se em como os membros da equipe devem trabalhar de forma a produzir sistemas flexíveis em um ambiente em constante mudanças (SCHWABBER e BEEDLE, 2001).

A metodologia Scrum prega o desenvolvimento de *software* em incrementos (*sprints*) por uma equipe auto organizada. As necessidades do cliente são registradas em *backlogs*. Antes de cada sprint, os itens de *backlog* são priorizados e aqueles que trazem mais valor ao cliente são implementados dentro da sprint. A cada sprint é feito um planejamento e existem reuniões diárias para acompanhar a evolução do desenvolvimento no *sprint* e, ao final, é feita uma reunião para consolidação das lições aprendidas no *sprint* (DYBÅ e DINGSØYR, 2008).

De acordo com Schwabber (2004), no *Scrum* são identificados três papéis principais (*scrum master*, *product owner* e Time *Scrum*) que têm diferentes propósitos e funções durante o processo de desenvolvimento.

O *Scrum Master* é responsável por assegurar que o projeto seja executado de acordo com as práticas, valores e regras definidas no *Scrum.* O *Scrum Master* interage com o time de desenvolvimento assim como com clientes e gestor durante o projeto. É responsabilidade do *Scrum Master* que qualquer impedimento seja removido para manter o time de desenvolvimento mais produtivo possível.

O *Product Owner* é oficialmente responsável pelo projeto, gerenciando, controlando e tornando visível a lista de *backlog* do produto. Ele é escolhido pelo Scrum Master, cliente e gerente. O *Product Owner* tem a decisão final sobre as atividades relacionadas ao *backlog* de produto, participa na estimativa de esforço de desenvolvimento dos itens de *backlog* e transforma os problemas encontrados no *backlog* em funcionalidades para serem desenvolvidas.

O Time de Scrum é responsável por se auto organizar e decidir que ações são necessárias para atingir os objetivos estabelecidos para a *sprint*. O Time Scrum está envolvido na estimativa de esforço, criação do *backlog* da *sprint*, revisão da lista do *backlog* de produto e sugestão de impedimentos que precisam ser removidos do projeto.

De acordo com Beck (1999), o método XP evoluiu dos problemas causados pelos longos ciclos de desenvolvimento de modelos tradicionais de desenvolvimento. O XP foi elaborado sobre princípios e práticas que foram consideradas eficazes nos processos de desenvolvimento de *software* durante as décadas anteriores. O termo *extreme* deriva de levar o uso dos princípios e práticas a níveis extremos (Beck 2000).

O método XP consiste em doze práticas: o jogo de planejamento (*planning game*), pequenos lançamentos, metáforas, design simples, testes, refatoração, programação em pares, propriedade coletiva, integração contínua, ritmo sustentável (40 horas de trabalho por semana), clientes no local e padrões de codificação (DYBÅ; DINGSØYR, 2008).

O ciclo de vida do XP consiste em cinco fases: exploração, planejamento, iteração, produção, manutenção e morte (Beck 2000).

Na fase de exploração os clientes escrevem em cartões (também conhecido como história de usuário) o que eles desejam que seja entregue na primeira release. Cada história de usuário descreve uma funcionalidade a ser adicionada no sistema. Nessa mesma fase os membros do time do projeto se familiarizam com as ferramentas, tecnologias e práticas que serão utilizadas no projeto. Também são elaborados protótipos para testar a arquitetura e tecnologias escolhidas (Beck 2000).

Na fase de planejamento é defina a ordem de prioridades das histórias de usuário e um acordo de quais funcionalidades serão desenvolvidas na primeira release. Nessa fase, os desenvolvedores definem o esforço de cada uma das histórias de usuário definidas pelos clientes e o cronograma de entregas é acordado (Beck 2000).

A fase de iteração inclui várias iterações sobre o sistema até que a primeira release seja entregue. O cronograma definido na fase anterior é quebrado em pequenas iterações. Na são escolhidas histórias de usuário que exijam a construção de toda estrutura necessária para o sistema. Os clientes escolhem as histórias de usuário que serão desenvolvidas em cada iteração. Os testes funcionais elaborados pelos clientes são executados ao final de cada iteração para assegurar a qualidade do que foi desenvolvido. Ao final da última iteração, o sistema está pronto para a produção (Beck 2000).

A fase de produção exige uma execução exaustiva de testes funcionais e de performance antes que o sistema seja entregue ao cliente. Nessa fase, podem surgir modificações e deve-se decidir se serão executadas na release atual ou posteriormente. As ideias adiadas são documentadas e desenvolvidas em outras fases como, por exemplo, a manutenção (Beck 2000).

Depois que o sistema é disponibilizado no ambiente de produção para o cliente, a equipe XP deve manter o sistema rodando em produção enquanto trabalha em novas iterações. Essa fase é caracterizada como manutenção, na qual são exigidas tarefas de suporte ao usuário. Dessa forma, a velocidade de entrega de releases pode cair. Novas pessoas são necessárias na equipe e pode haver mudanças na estrutura do time (Beck 2000).

A fase de morte está próxima quando existem mais histórias de usuário para serem implementadas. Dessa forma, o sistema satisfaz todas as necessidades dos clientes (funcionais, e não funcionais). Nessa fase, a documentação do sistema é escrita e nenhuma alteração de código, arquitetura ou design é feita. Essa fase ocorre também quando o sistema não estiver entregando os resultados esperados ou se o sistema se tornar muito caro para desenvolvimento adicional (Beck 2000).

O pensamento *Lean* foi utilizado inicialmente em Womack, Jones e Ross (1990). Os autores relatam a evolução da indústria automobilística. O processo de produção automobilístico passou da produção artesanal para produção em massa e posteriormente para a produção *Lean*. Os autores afirmam que as empresas no Japão não dispunham de capital suficiente para investir na produção em massa, assim foi criada uma forma alternativa de trabalho: baixo estoque e processo de decisão atribuído aos trabalhadores da fábrica.

Segundo Womak e Jones (2010), *Lean* é um modelo gerencial aplicado por meio de princípios e técnicas operacionais. Os objetivos do *Lean* são reduzir o desperdício e melhorar a qualidade e a maximização do valor entregue ao cliente. Em sua essência, é uma filosofia orientada a eficiência e eficácia de processos, centrada em criar mais valor com menos trabalho.

Womack e Jones (1996) definem o *Lean Thinking* (LT) como maneira de especificar valor, criar uma sequência otimizada de ações de geração de valor, conduzir estas ações sem interrupção e otimizar cada vez mais as ações. Cinco princípios foram introduzidos no LT para lidar com diferenças culturais e processos gerenciais dentro das organizações:

1. Valor: o valor é definido pelo cliente em forma de um produto ou serviço específico que atende à demanda ou problema.
2. Fluxo de valor: definido como o conjunto de ações requeridas para mover um produto específico através das três tarefas críticas de gerenciamento de uma unidade de negócio: processos de resolução de problemas, processos de gerenciamento de informação e processos de transformação física. A análise do

fluxo de valor identifica atividades que efetivamente geram valor. As atividades que não geram valor devem ser eliminadas para evitar o desperdício.

3. Fluxo contínuo: após a identificação da cadeia de valor e a análise do processo de formação de valor, deve-se introduzir modificações no processo de produção que ocorram em fluxo contínuo. A implementação de sistemas de produção com processos em fluxo contínuo causa impacto direto no *lead time*.
4. Puxar: nesse princípio, os processos posteriores sinalizam aos processos anteriores sobre suas necessidades de produção. Desta forma, tenta-se eliminar os desperdícios associados à superprodução.
5. Perfeição: busca contínua da aproximação entre o produto ou serviço criado e a necessidade do mercado.

Segundo Thangarajoo e Smith (2015), os cinco princípios do LT guiam a administração no processo de desenvolver a cultura *Lean* nas empresas e cria uma jornada contínua de eliminação de desperdícios através da revisão de cada atividade na cadeia de valor para identificar oportunidades de melhorias adicionais.

Poppendieck e Poppendieck (2003) observaram a possibilidade de aplicação da abordagem *Lean* para o desenvolvimento de *software*. Essa adaptação foi definida como Desenvolvimento de *Sofware Lean* (*Lean Sofware Development).* Nessa obra, os autores definiram princípios, práticas e filosofia para a construção de sistemas de *software*.

Segundo Poppendieck e Poppendieck (2006), princípios são verdades subjacentes que não mudam no tempo e no espaço, enquanto práticas são a aplicação dos princípios a uma situação particular. As práticas podem mudar à medida que a situação evolui.

Poppendieck e Poppendieck (2003) definiram sete princípios que norteiam o Desenvolvimento de *Software Lean* (DSL). Posteriormente, em Poppendieck e Poppendieck (2006), os princípios foram revistos e tiveram seus nomes modificados, mas a essência continuou inalterada.

O Desenvolvimento de *Software Lean* define sete princípios básicos que devem ser observados:

1. Eliminação de desperdício: a eliminação do desperdício diz respeito a observar quais são as atividades e recursos que são absolutamente necessários para entrega de valor. Tudo o mais que não estiver relacionado a isso deve ser eliminado. Para eliminar o

desperdício é necessário reconhece-lo. O primeiro passo para utilizar o *Lean* é entender o que é valor e quais atividades e recursos são absolutamente necessários para cria-lo.

2. Construa com qualidade: Segundo Shingo (1989), existem dois tipos de inspeção: a inspeção que ocorre após a ocorrência do defeito e a inspeção para prevenir o defeito. O comprometimento com a qualidade exige que exista qualidade no processo de desenvolvimento desde o início não permitindo que o erro ocorra. Se isso não for possível é necessário inspecionar o produto a cada pequeno passo, para que os defeitos sejam detectados rapidamente. Quando um defeito é encontrado, é necessário encontrar a causa e corrigir o erro rapidamente.
3. Crie conhecimento: o processo de desenvolvimento de *software* é uma atividade complexa e o desenvolvedor nem sempre sabe como fazer o trabalho da melhor maneira. Dessa forma, são necessários pequenos experimentos e experiência adquirida por resultados. O uso de iterações curtas e frequentes está fortemente ligado a esse princípio. Através dos ciclos rápidos, aumenta-se o feedback sobre o que está sendo feito, permitindo-se assim corrigir qualquer coisa que não esteja de acordo com o especificado. É importante ter um processo de desenvolvimento que incentive o aprendizado sistemático ao longo do ciclo de desenvolvimento, porém é necessário melhorar sistematicamente esse processo de desenvolvimento. Os esforços de melhoria de processos devem ser de responsabilidade das equipes de desenvolvimento e cada equipe deve reservar um tempo para trabalhar regularmente na melhoria de processos.
4. Respeite as pessoas: esse princípio afirma que as pessoas envolvidas no processo de desenvolvimento devem ser responsáveis pela melhoria do processo de desenvolvimento do time. Isso implica no desenvolvimento da equipe para o aprimoramento do conhecimento das pessoas que codificam o *software* a partir de treinamentos. Quando mudanças ocorrem muito rapidamente, é necessário trocar a liderança diretiva por uma liderança liberal. Na liderança diretiva, o líder determina as providências e as técnicas para a execução das tarefas. Na liderança liberal, as pessoas possuem mais liberdade na execução dos seus projetos. A função dos gerentes é dar suporte a equipe e não dar ordens sobre o que deve ser feito.
5. Decida o mais tarde possível: esse princípio visa desenvolver funcionalidades somente quando são demandadas e depois de se obter informações suficientes a

respeito das necessidades do cliente. Esse princípio também tem o objetivo de maximizar o fluxo de informações e a entrega de valores. Deve-se eliminar documentação desnecessária, que não agrega valor. Testes automatizados são implementados para assegurar a qualidade do *software* nas entregas iniciais e em entregas futuras, quando mudanças inevitáveis são necessárias.

6. Entregue o mais rápido possível: esse princípio está relacionado a entrega rápida de *software* funcional. O desenvolvimento rápido permite que as necessidades do cliente sejam atendidas prontamente, dando a ele a oportunidade de adiar a tomada de decisão até o momento mais propício. Empresas que competem com base na velocidade possuem vantagens de custo em relação a seus competidores, pois possuem um processo com menos desperdício, maior qualidade de produtos e um profundo conhecimento dos seus clientes. A rapidez no ciclo de desenvolvimento é feita de forma iterativa e permite que o cliente refine as necessidades para as próximas entregas.
7. Otimize toda a organização: esse princípio implica em envolver toda a organização no processo de adoção das práticas *Lean*. Deste modo, a organização deve ser estruturada em equipes que mantêm a responsabilidade por toda a cadeia de valor *Lean* de negócio, sempre mantendo o objetivo de ajudar o cliente obter valor com o *software* desenvolvido. Quando uma organização se concentra em otimizar as partes sem possuir uma visão global, é comum que o fluxo de valor sofra.

Além de definirem os sete princípios para o DSL, Poppendieck e Poppendieck (2006) mapearam sete tipos de desperdícios encontrados no desenvolvimento de *software*:

1. Trabalho parcialmente feito - o código deve ser integrado, testado, documentado e implantado em um único fluxo.
2. Funcionalidade extra – funções que não são necessárias para o cliente.
3. Reaprendizado – tempo gasto para aprender algo que já era conhecido pela equipe.
4. Transferência de controle – perda de conhecimento com a transferência de controle devido à dificuldade de transmitir o conhecimento tácito.
5. Multitarefa – a troca de contexto entre projetos causa um desperdício de tempo por parte do desenvolvedor.

6. Tempo de espera/ Atrasos - a demora para iniciar atividades ou organizar equipes, por exemplo, corresponde a atrasos na perspectiva do cliente e impede que *feedback* e decisões ocorram no melhor momento possível.
7. Defeitos – qualquer atividade que não deveria ser realizada. Por exemplo, quando o bug é detectado na fase de produção em vez de ser detectado na fase de desenvolvimento, causa desperdício de tempo durante o rastreamento e o conserto do *bug*.

De acordo com Poppendieck e Poppendieck (2006), a proposta *Lean* é considerada radical porque é mais que uma mudança de hábitos; chega a ser uma mudança de cultura.

De acordo com Sunner (2016), o método Kanban é basicamente um método criado para organizar o caos em torno do time de desenvolvimento.

Anderson (2010) descreve o Kanban como base de um sistema de desenvolvimento de *software* eficaz e baseado em fluxo. Em um sistema Kanban, o valor fluxo é mapeado em um gráfico (de preferência um gráfico físico em uma parede) com colunas para cada etapa no valor corrente.

Cartões Kanban são colocados na coluna, representando o estado atual do trabalho. Quando o trabalho atende as regras estabelecidas para conclusão, o cartão passa para a próxima coluna (da esquerda para a direita). A ideia principal do sistema Kanban é que o trabalho em qualquer coluna (representando uma etapa no fluxo de valor) é limitado (POPPENDIECK e CUSUMANO, 2012).

O método Kanban utilizado em desenvolvimento de *software* permite ao time de projeto visualizar o fluxo de trabalho, limitar o trabalho em progresso em cada estágio de trabalho e medir o tempo do ciclo (AHMAD, MARKKULA e OVIO, 2013).

Os sistemas Kanban oferecem uma boa estrutura para organizações que estão iniciando a utilização de princípios enxutos. O quadro Kanban, por exemplo, pode começar em um ambiente de desenvolvimento de *software*, mas pode ser facilmente expandido para incluir mais etapas no fluxo de valor, como marketing e operações. Isso faz do Kanban uma boa ferramenta para equipes de fluxo de valor. Os sistemas Kanban se concentram especificamente no fluxo de valor no qual fluxo e gargalos são o principal tópico das reuniões diárias (POPPENDIECK e CUSUMANO, 2012).

Por fim, o quadro 1 apresenta um resumo das características dos métodos ágeis apresentados no trabalho.

Quadro 1 Comparativo métodos ágeis

Método	Características	Forças	Fraquezas
Scrum	- desenvolvimento iterativo - trabalho em *timebox* (*sprints*) - reuniões diárias - time auto organizado - tarefas administradas usando *backlogs*	- entrega de produtos em ciclos curtos - rápido *feedback* - rápida adaptação a mudanças	- falta de documentação -requer desenvolvedores experientes - difícil de realizar estimativas no início do projeto
Extreme programming	- programação em pares - testes unitários - entregas rápidas consecutivas - propriedade coletiva - trabalho próximo ao *Product Owner* - ambiente de trabalho aberto - o *Project Owner* decide a prioridade das tarefas	- aplicação disponibilizada rapidamente em produção - entregas frequentes de código - número reduzido de falhas - integração de código suave - feedback contínuo *do Product Owner*	- falta de documentação - relutância em programa por pares - resistência de escrever testes antes de codificar - requer encontros frequentes - falta de comprometimento definição de produto leva a um product owner relutante
Lean software development	- desenvolvimento iterativo - descarte de tudo que não agregar valor ao produto - conhecimento ampliado - foco no cliente - empoderamento do time - melhoria contínua	- redução de custo e tempo de projeto - entrega antecipada de código - time de projeto motivado	- o projeto é altamente confiado as pessoas da equipe - requer membros da equipe com amplos conhecimentos em análise de negócio.
Kanban	-visualização de fluxo de trabalho -limita o trabalho em progresso	- aumenta a colaboração -permite medir o tempo de trabalho - flexível	- consumo de tempo para administrar o quadro - capacidade limitada de rastreamento e monitoração - escalabilidade limitada para lidar com picos de demandas

Fonte: Adaptado de Rajagopalan e Mathew (2016); Wan e Chen (2007)

O quadro 1 permite observar que os métodos ágeis são eficientes, porém apresentam falhas. Segundo Chow e Cao (2008), a escolha do método ágil mais adequado depende das características da organização.

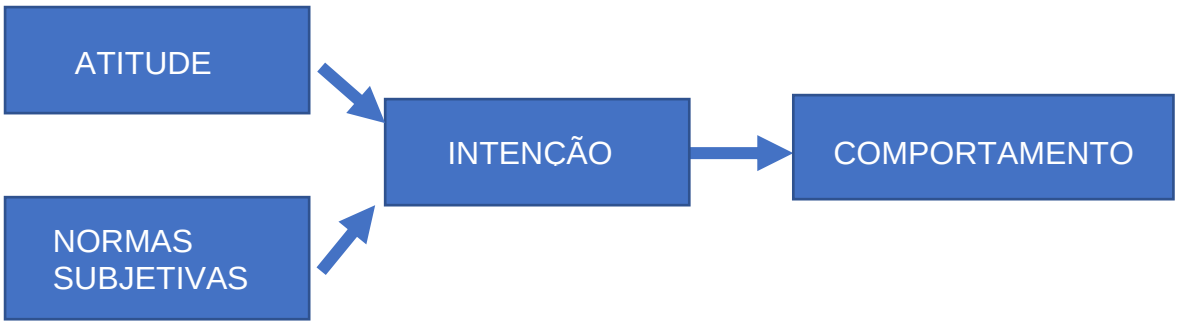
2 MODELOS DE ACEITAÇÃO E USO DA TECNOLOGIA

Os modelos de aceitação de tecnologia foram elaborados com base em duas teorias principais: a Teoria da Ação Racional - *Theory of Reasoned Action* (TRA) (FISHBEIN; AJZEN, 1975) e a Teoria do Comportamento Planejado - *Theory of Planned Behaviour* (TPB) (AJZEN, 1985).

2.1 Teoria da Ação Racional - *Theory of Reasoned Action* (TRA)

A Teoria da Ação Racional (TRA), apresentada na Figura 4, define que o comportamento do indivíduo é determinado pelas intenções comportamentais que são resultantes de suas atitudes e percepções sobre as normas subjetivas que são incorporadas ao seu comportamento (FISHBEIN; AJZEN, 1975).

Figura 4 Teoria da Ação Racional - TRA



Fonte: Adaptado de Fishbein e Ajzen (1975)

Segundo Fishbein e Ajzen (1975), o constructo Atitude em relação ao Comportamento é definido como sentimentos negativos ou positivos em relação a um determinado Comportamento. Já, o constructo Normas Subjetivas é a percepção do indivíduo sobre a opinião, de aprovação ou reprovação, de determinado comportamento das pessoas que ele considera importante.

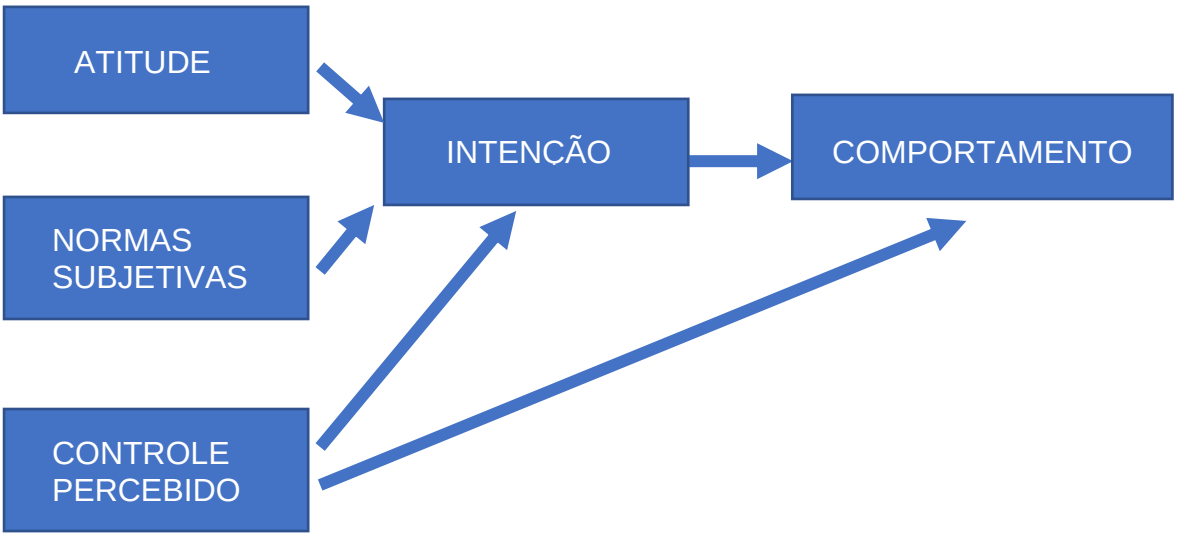
2.2 Teoria do Comportamento Planejado - *Theory of Planned Behaviour* (TPB)

A Teoria do Comportamento Planejado (TPB), proposta por Ajzen (1985), incorpora um elemento adicional que determina a intenção, conforme demonstra a Figura 5. Esse

elemento é o comportamento percebido de controle ou a crença de que ele é bem-sucedido na tarefa que tem em mãos.

Para Ajzen (2002), a TPB baseia-se no pressuposto de que os indivíduos tomam suas decisões de forma eminentemente racional e utilizam sistematicamente as informações que estão disponíveis, considerando as implicações de suas ações antes de decidirem se devem ou não se comportar de determinada forma.

Figura 5 Teoria do Comportamento Planejado - TPB



Fonte: Adaptado de Ajzen (1985)

Segundo Ajzen (2002), para modificar o comportamento, as intervenções podem ser direcionadas a um ou mais de seus três determinantes: atitude, norma subjetiva ou controle percebido. Uma vez que os indivíduos tenham verdadeiro controle sobre o comportamento, novas intenções comportamentais podem ser produzidas e convertidas em um comportamento real.

2.3 Modelo de aceitação da tecnologia – *Technology Acceptance Model* (TAM)

O Modelo de aceitação da tecnologia (TAM) foi proposto por Davis (1989). O modelo TAM foi adaptado da TRA e tem como objetivo predizer as variáveis antecedentes da intenção de comportamento para o uso da tecnologia da informação (DAVIS, 1989).

Segundo Venkatesh et al. (2003), o modelo TAM, ao contrário do modelo TRA, exclui o constructo atitude em relação ao comportamento para explicar melhor a intenção de uso da tecnologia da informação.

A Figura 6 demonstra o modelo TAM.

Figura 6 Modelo de Aceitação de Tecnologia - TAM

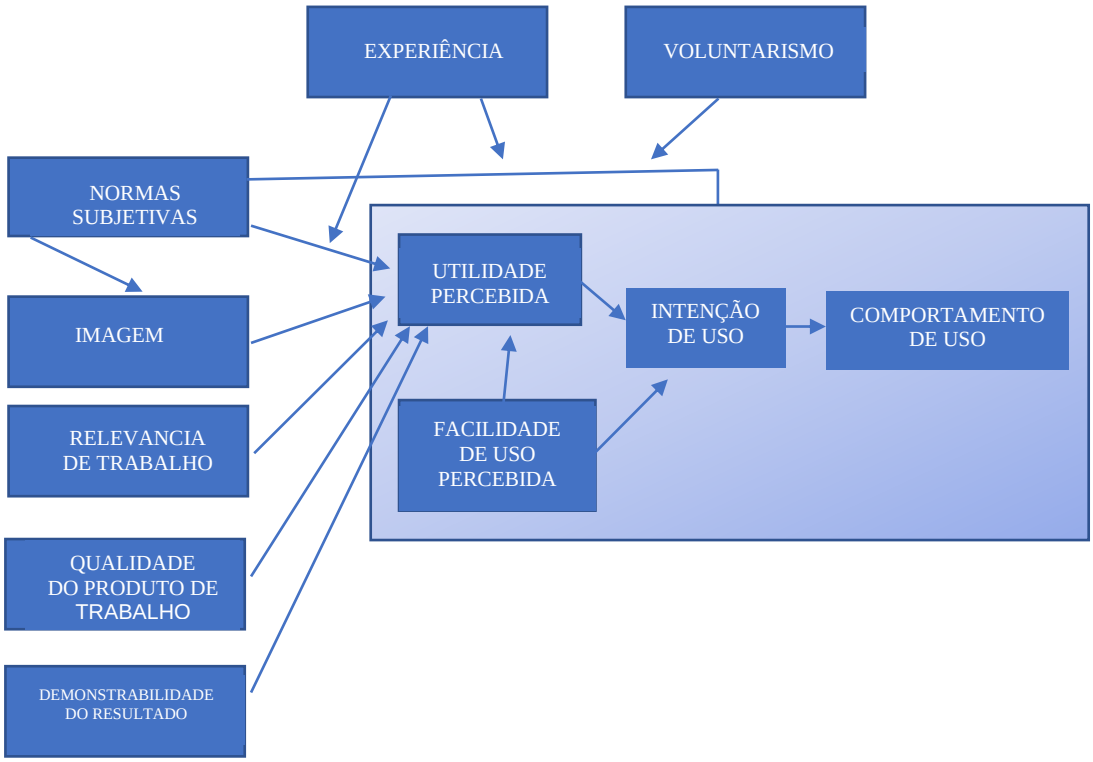
Fonte: Adaptado de Davis et al. (1989)

Os constructos dos modelos são utilidade percebida e facilidade de uso. O constructo utilidade percebida é o grau em que uma pessoa compreende que o uso de um sistema irá melhorar seu trabalho e o constructo facilidade de uso é o grau em que o indivíduo entende que é fácil utilizar um sistema (DAVIS, 1989).

2.4 Extensão do Modelo de Aceitação de Tecnologia (TAM-2)

O modelo TAM-2, demonstrado na Figura 7, surgiu com o objetivo de estender o Modelo de Aceitação de Tecnologia, para explicar a utilidade e intenção percebida de uso relacionada a influência social e processos cognitivos instrumentais. Foram adicionados os constructos normas subjetivas, imagem, relevância do trabalho, qualidade do produto do trabalho e demonstrabilidade do trabalho. Nesse modelo, a voluntariedade e a experiência são considerados moderadores para o modelo (VENKATESH; DAVIS, 2000).

Figura 7 Extensão do modelo de Aceitação de Tecnologia - TAM2



Fonte: Adaptado de Venkatesh e Davis (2000)

O constructo Norma Subjetiva está relacionado à percepção do indivíduo sobre a opinião, de aprovação ou reprovação, de determinado comportamento, das pessoas que ele considera importante. A Imagem é o grau em que o uso de um sistema favorece a melhoria de imagem ou no status de um indivíduo dentro de um sistema social. A Relevância de Trabalho é a percepção do indivíduo quanto ao grau para o qual o sistema de destino é aplicável ao seu trabalho. A Qualidade do Trabalho é a forma como o sistema executa as tarefas, especificamente da percepção da qualidade da saída da tarefa. A Demonstrabilidade do Resultado é a tangibilidade dos resultados utilizando a inovação, incluindo a sua observância e comunicabilidade. A Utilidade Percebida é o grau em que uma pessoa compreende que o uso de um sistema irá melhorar seu desempenho no trabalho.

Já a Facilidade Percebida é o grau em que a pessoa entende que é fácil utilizar um determinado sistema.

2.5 Modelo Combinado TAM-TPB (C-TAM-TPB)

O modelo de aceitação de tecnologia TAM-TPB foi desenvolvido por Taylor e Todd (1995) para medir a experiência anterior do usuário em tecnologia da informação. Esse modelo originou-se da combinação da Teoria do Comportamento Planejado (TPB) e o Modelo de Aceitação da Tecnologia (TAM) e possui os seguintes constructos:

- Atitude em relação ao comportamento
- Norma subjetiva
- Controle percebido
- Facilidade de uso

Para Taylor e Todd (1995), o modelo TAM concentra-se em características de design do sistema e do uso particular como um guia para projetar esforços. Já, o modelo TPB tem foco nos fatores normativos e de controle com os quais uma organização pode trabalhar para facilitar a implementação do sistema. Dessa forma, a combinação dos dois modelos proporciona uma compreensão melhor da intenção e do comportamento.

2.6 Modelo Motivacional – *Motivation Model* (MM)

Para Venkatesh et al. (2003), um número significativo de pesquisas em psicologia tem apoiado a teoria geral motivacional para explicar o comportamento determinado pela motivação intrínseca e a motivação extrínseca.

Sob o ponto de vista do uso de tecnologia, a motivação intrínseca se refere à percepção de como os usuários irão executar uma ação motivados pelo próprio processo de execução da ação. Por sua vez, a motivação extrínseca está relacionada à percepção de que os usuários irão executar determinada ação porque ela é percebida como importante para alcançar os resultados em decorrência de recompensa externas. No Modelo Motivacional, a utilidade percebida quanto a aprazibilidade influencia a intenção de uso, porém a utilidade percebida apresenta um peso maior que a segunda variável (DAVIS; BAGOZZI; WARSHAW, 1992).

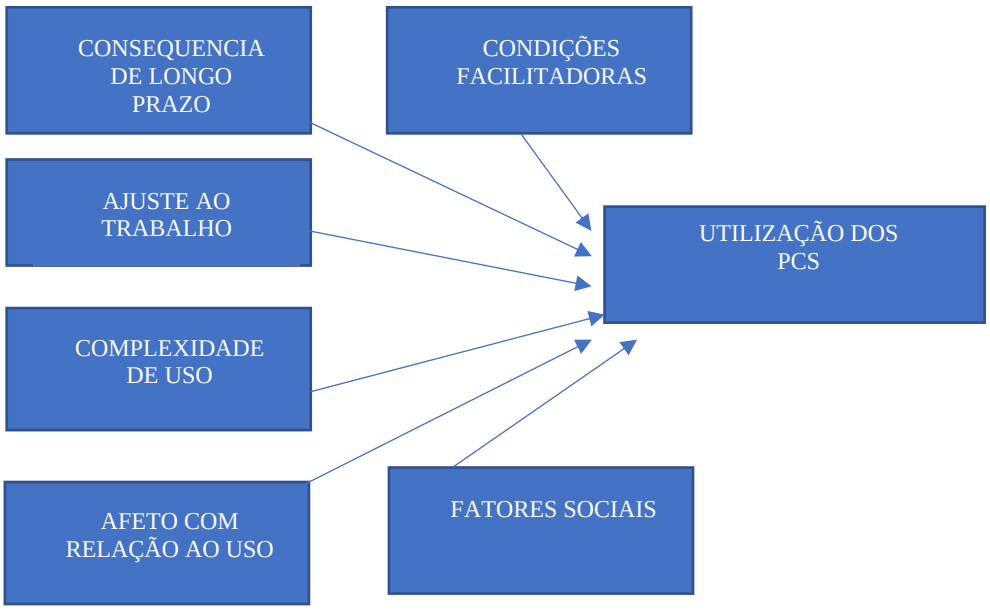
2.7 Modelo de Utilização de Computador Pessoal – *Model of Personal Computer Utilization* (MPCU)

O Modelo de Utilização de Computador Pessoal foi desenvolvido por Thompson, Higgins e Howell (1991) e foi baseado no trabalho de Triands (1980).

Triands (1980) propõe uma teoria que incorpora aspectos do modelo TRA, porém modifica e redefine os constructos. Existe uma distinção entre crenças que ligam emoções aos atos e crenças que ligam o ato com futuras consequências. As intenções de uso são determinadas por sentimentos que o indivíduo tem em relação a comportamento, fatores sociais e consequências esperadas do comportamento Triands (1980).

O modelo MPCU (Figura 8) está fundamentado em seis constructos que explicam a utilização do computador pessoal.

Figura 8 Modelo de Utilização de PC - MPCU



Fonte: Adaptado de Thompson, Higgins e Howell (1991)

O constructo Condições Facilitadoras está relacionado a fatores objetivos no ambiente que observadores concordam como facilitadores de um ato. No contexto de sistema de informação, a provisão de suporte para usuários de Computador Pessoal é um exemplo de condição facilitadora que influencia a utilização do sistema de informação.

A Consequência de Longo Prazo representa os resultados que têm uma compensação futura. O Ajuste ao Trabalho é o grau no qual o indivíduo acredita que utilizar uma tecnologia pode melhorar o seu desempenho no trabalho. E a Complexidade de Uso refere-se ao grau de inovação que é percebida como relativamente difícil de entender e usar.

No constructo Afeto com Relação ao Uso, um indivíduo associa os sentimentos de alegria, prazer, depressão, desgosto, desagrado ou ódio a um determinado ato. O constructo Fatores Sociais refere-se à internalização da referência da cultura subjetiva do grupo pelo indivíduo e relações interpessoais específicas que o indivíduo tem com outros nas relações sociais.

2.8 Teoria da difusão da Inovação – *Innovation Diffusion Theory* (IDT)

A IDT tem origem na sociologia e tem como objetivo investigar a reação de indivíduos a novos produtos e processos, além de explicar por que algumas inovações são difundidas e outras não (ROGERS,1983).

Segundo Rogers (2003), a difusão é o processo pelo qual uma inovação é comunicada por meio de determinados canais durante um período entre os componentes de um sistema social.

A IDT está baseada em cinco constructos que influenciam a difusão:

- Vantagem relativa: grau em que uma inovação é percebida como superior às tecnologias anteriores
- Compatibilidade: grau em que uma inovação é percebida como consistente com os valores existentes, experiências passadas e normas entre seus usuários
- Complexidade: grau de dificuldade de utilização da inovação
- Experimentação: oportunidade de testar a inovação antes de se comprometer a utilizá-la
- Observabilidade: grau em que os resultados da inovação são visíveis para os outros.

Além dos constructos definidos, outras variáveis determinam a taxa de adoção da inovação: tipos de decisão da inovação, canais de comunicação, natureza do sistema social e intensidade de esforço.

Os tipos de decisão da inovação podem ser opcionais, ou seja, feitos por indivíduos independentemente da decisão de outros membros: coletivas, tomadas por consenso entre membros de um sistema; e autoridade, que são feitas por pessoas que possuem algum tipo de influência (ROGERS, 2003).

A respeito dos canais de comunicação, as mídias de massa são mais eficazes na criação de conhecimento de inovação e os canais de comunicação pessoal são mais eficazes na formação e na mudança de atitude. Ambos influenciam a decisão de aprovar ou rejeitar a informação (ROGERS, 2003).

Segundo Rogers (2003), a natureza do sistema social é composta de normas e regras sociais, que geram e estabelecem o comportamento dos membros do sistema social e a intensidade de esforço é determinada pela existência de agentes que influenciam os indivíduos (formal ou informalmente) na tomada de decisão de adoção da inovação.

Moore e Benbasat (1991) adaptaram a IDT para entender os diferentes graus de aceitação da Tecnologia da Informação por indivíduos. O modelo proposto por Moore e Benbasat (1991) definiu os constructos Vantagem Relativa, Facilidade de Uso, Imagem, Visibilidade, Compatibilidade, Demonstrabilidade de Resultados e Voluntariedade de Uso.

A Vantagem Relativa indica o grau em que uma inovação é percebida como sendo melhor que sua precursora. A Facilidade de Uso está relacionada ao grau em que uma inovação é percebida como de difícil uso.

A Imagem é o grau em que o uso de uma inovação é percebido como benéfico para a própria imagem ou status no sistema social. A Visibilidade é o grau em que se pode ver outras pessoas usando o sistema na organização.

A Compatibilidade indica o grau em que uma inovação é percebida como sendo consistente com os valores existentes, as necessidades e experiências passadas dos adotantes da inovação.

A Demonstrabilidade de Resultados é o grau em que os resultados da utilização da inovação são observados e comunicados a outros. E a Voluntariedade de Uso é o grau em que a utilização da inovação é percebida como voluntária ou por vontade própria.

2.9 Teoria Social Cognitiva – *Social Cognitive Theory* (SCT)

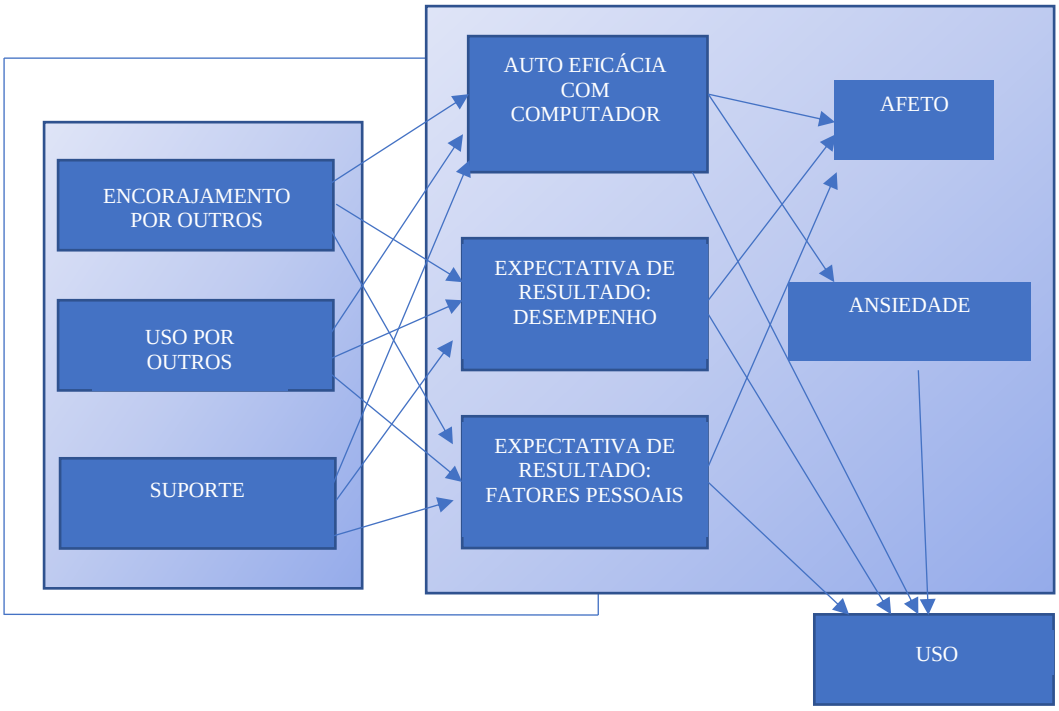
A Teoria Social Cognitiva foi proposta por Bandura (1986) e estabelece que o indivíduo tem um papel central nos processos cognitivos, vicários, auto reguladores e auto reflexivos na adaptação e nas mudanças humanas. O funcionamento humano é visto como um produto de uma interação dinâmica de influências pessoais (na forma de cognição, afetação e eventos biológicos, comportamentais e ambientais, essa interação é chamada de reciprocidade tríade (BANDURA 1986).

Segundo Bandura (1986) o comportamento motivado é um comportamento dirigido a um objetivo e esse é sustentado e ativado pela expectativa do resultado esperado da ação e pela percepção da auto eficácia para executar a ação. A auto eficácia é definida como o juízo pessoal sobre as próprias capacidades para executar as atividades requeridas para atingir determinado nível de desempenho. A expectativa de resultado está relacionada à crença nos resultados antecipados dessas ações (BANDURA, 1986).

No contexto da tecnologia da informação, Compeau e Higgins (1995) realizaram estudo sobre a aceitação do uso de computador utilizando SCT, destacando a expectativa de resultado e a auto eficácia como fatores que influenciam positivamente no uso de computadores.

Compeau e Higgins (1995) propõem um modelo (Figura 9) para explicar o uso do computador baseado na reciprocidade tríade.

Figura 9 Aplicação da SCT na avaliação de adoção de tecnologia



Fonte: Adaptado de Compeau e Higgins (1995)

O modelo definido por Compeau e Higgins (1995) possui os constructos Encorajamento por outros, Uso por Outros, Suporte, Auto Eficácia, Expectativa de Resultados relacionados a desempenho e fatores pessoais: Afeto e Ansiedade.

O constructo Encorajamento por Outros indica que pessoas que pertencem ao grupo de referência do indivíduo apoiam determinado comportamento.

O constructo Uso por Outros é a percepção de que há outros indivíduos utilizando a tecnologia. O constructo Suporte está relacionado à disponibilidade de ajuda para o indivíduo usar o computador, caso necessite, no ambiente corporativo.

A Auto eficácia refere-se ao julgamento do indivíduo de sua habilidade de usar tecnologia.

A Expectativa de Resultados referente ao desempenho está ligada às consequências do comportamento relacionadas ao desempenho. Neste caso especificamente, a expectativa de desempenho lida com os resultados relacionados a função. Já a Expectativa de Resultados, sob o ponto de vista de fatores pessoais, está relacionada as consequências pessoais do comportamento. A Expectativa Pessoal está relacionada a autoestima e senso de realização.

O Afeto é caracterizado pela apreciação de um indivíduo a um comportamento em particular. E a Ansiedade refere-se a reações emocionais evocadas quando se trata de adotar um comportamento (ex.: uso do computador).

2.10 Teoria Unificada de Aceitação de Uso da Tecnologia – *Unified Theory of Acceptance and Use of Technology* (UTAUT)

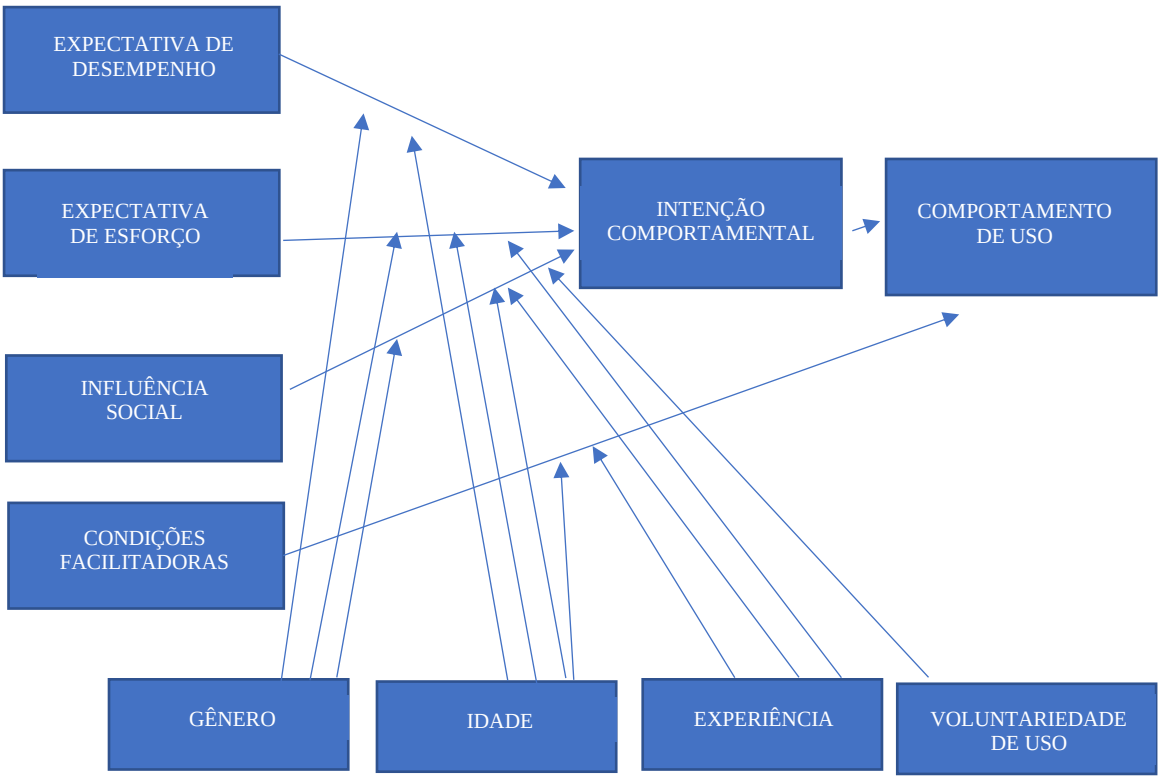
A Teoria Unificada de Aceitação de Uso da Tecnologia foi proposta por Venkatesh et al. (2003) e surgiu da junção de modelos de aceitação e uso da tecnologia.

No estudo de Venkatesh et al. (2003) foram examinados e comparados empiricamente oito modelos de aceitação de tecnologia (Teoria da Ação Racionalizada – TRA; Teoria do Comportamento Planejado – TPB; Modelo de Aceitação da Tecnologia – TAM; Modelo Combinado TAM-TPB; Modelo Motivacional – MM; Modelo de Utilização do PC – MPCU; Teoria da Difusão da Inovação – IDT; e Teoria Social Cognitiva – SCT) por meio de um estudo longitudinal com indivíduos de quatro organizações que foram introduzidos em uma nova tecnologia no ambiente de trabalho.

A partir da validação empírica, Venkatesh et al. (2003) formularam a Teoria Unificada de Aceitação de Uso da Tecnologia. Nesse estudo desenvolveram um modelo (Figura 10) com quatro constructos determinantes da intenção e uso da tecnologia (expectativa de desempenho; expectativa de esforço; Influência social e condições facilitadoras) e quatro variáveis moderadoras da relação entre os constructos (gênero, idade, experiência e voluntariedade), explicando 70% da variância da intenção de uso.

A Figura 10 relaciona os constructos do modelo UTAUT com a intenção e comportamento de uso de tecnologia da informação.

Figura 10 Modelo UTAUT



Fonte Adaptado de de Venkatesh et al. (2003)

Segundo Venkatesh et al. (2003), a Expectativa de Desempenho constitui o constructo com maior influência sobre a intenção comportamental. Esse constructo é acompanhado de Expectativa de Esforço e Influência Social e Condições Facilitadoras e são determinantes para a aceitação do usuário e para o comportamento de uso. Os constructos antecedentes (Expectativa de Desempenho, Expectativa de Esforço, Influência Social e Condições Facilitadoras) e consequentes (Intenção Comportamental e Comportamento de Uso), por sua vez são moderados por gênero, idade, experiência e voluntariedade de uso.

O constructo intenção comportamental, que tem origem no modelo TRA, é mediador do comportamento de uso e define o grau que um indivíduo se sente motivado a adotar o comportamento com base na expectativa de desempenho, expectativa de esforço e na influência social. A intenção comportamental é independente das condições facilitadoras, que viabilizam o comportamento de uso (VENKATESH ET AL. ,2003).

Os quadros 2, 3, 4 e 5 apresentam informações referentes às definições e origens dos constructos determinantes do modelo UTAUT.

O Quadro 2 apresenta a definição e origem dos constructos relacionados a Expectativa de desempenho.

Quadro 2 Constructos Expectativa de Desempenho

Constructo	Definição	Origem
Utilidade percebida	Mede o grau em que uma pessoa acredita que usando um sistema em particular melhora o desempenho do seu trabalho.	TAM/TAM2 e DTPB
Motivação extrínseca	A percepção de que os usuários vão querer realizar uma atividade porque contribui para a obtenção de resultados valiosos que são distintos da atividade em si.	MM
Ajuste ao trabalho	Como os recursos de um sistema melhoram o desempenho do trabalho de um indivíduo	MPCU
Vantagem relativa	O grau em que o uso uma inovação é percebida como sendo melhor do que a utilização do seu precursor.	IDT
Expectativa de resultado	Está relacionado com as consequências do um comportamento. No âmbito pessoal lidam com a autoestima e senso do dever cumprido e no âmbito de trabalho lidam com os resultados da função.	SCT

Fonte: Adaptado de Venkatesh et al. (2003)

O Quadro 3 apresenta a definição e origem dos constructos relacionados a Expectativa de esforço.

Quadro 3 Constructos Expectativa de esforço

Constructo	Definição	Origem
Percepção de facilidade de uso	O grau em que uma pessoa acredita que é fácil utilizar um sistema.	TAM e TAM2
Complexidade	O grau em que um sistema é percebido como relativamente difícil de entender e usar.	MPCU
Facilidade de uso	O grau em que o uso de uma inovação é percebido como relativamente difícil de entender e usar.	IDT

Fonte: Adaptado de Venkatesh et al. (2003)

O Quadro 4 apresenta a definição e origem dos constructos relacionados a Influência Social.

Quadro 4 Constructos Influência Social

Constructo	Descrição	Origem
Normas Subjetivas	A percepção do indivíduo sobre a opinião, de aprovação ou reprovação de determinado comportamento, das pessoas que ele considera importante.	TRA, TPB, C- TAM-TPB e TAM2
Fatores Sociais	A internalização da referência da cultura subjetiva do grupo pelo indivíduo e relações interpessoais específicas que o indivíduo tem com outros nas relações sociais.	MPCU
Imagem	O grau em que o uso de uma inovação é percebido para melhorar a imagem ou status em um sistema social	IDT e TAM2

Fonte: Adaptado de Venkatesh et al. (2003)

O Quadro 5 apresenta a definição e origem dos constructos relacionados a Influência Social.

Quadro 5 Constructos Condições Facilitadoras

Constructo	Definição	Origem
Controle Comportamental percebido	Reflete a percepção de restrições de comportamento interno e externo e abrange a auto eficácia, recursos facilitando condições e tecnologia facilitando condições.	TPB e C-TAM-TPB
Condições facilitadoras	Fatores objetivos no ambiente que observadores concordam como facilitador para a execução de um ato, incluindo a provisão de equipe de suporte no uso de computador	MPCU
Compatibilidade	O grau em que uma inovação é percebida como sendo consistente com valores existentes, necessidades e experiências de adotantes potenciais.	IDT

Fonte: Adaptado de Venkatesh et al. (2003)

Segundo Venkatesh, Thong e Xu (2012), desde a publicação original o modelo UTAUT tem sido aplicado ao estudo de uma variedade de tecnologias e em diversos contextos organizacionais.

Hong et al. (2011) desenvolveram um estudo com o objetivo de avaliar o uso e aceitação de *softwares* desenvolvidos com métodos ágeis, bem como a intenção de uso de novas funcionalidades entregues.

O estudo desenvolvido por Morrison, Smith e Williams (2017) investigou a aderência de práticas seguras de desenvolvimento de *software* utilizando um modelo de medida construído para aferir o grau de aderência às práticas de segurança. Nesse estudo foram utilizados os modelos TAM e UTAUT para medir o grau de aderência às práticas de segurança.

Čižmešija e Stapić (2018) investigaram o uso de aceitação do GithHub no contexto de aprendizagem de engenharia de *software*. O estudo foi realizado com estudantes de um curso de Engenharia de *Software*, que trabalharam em um projeto de *software* com características reais de desenvolvimento.

Por fim, Mudarikwa e Grace (2018) exploraram o uso e aceitação dos métodos ágeis por colaboradores de duas grandes organizações pertencentes ao ramo de banco e finanças na África do Sul. Esse estudo foi baseado no modelo UTAUT.

O presente trabalho tem a intenção de preencher uma lacuna na literatura de Engenharia de *Software* propondo um estudo sobre o uso e aceitação dos métodos ágeis de desenvolvimento de *software* em uma organização da indústria brasileira de *software*.

Para o presente estudo o modelo UTAUT será utilizado para avaliar o nível de aceitação do processo corporativo de desenvolvimento de *software* pelos desenvolvedores e gestores do SERPRO.

3 MÉTODO

A seguir são descritos a caracterização da pesquisa, o ambiente de pesquisa e os procedimentos metodológicos.

3.1 Caracterização da pesquisa

O método utilizado para o presente trabalho é de abordagem quali-quantitativa (ou misto), exploratória e descritiva, com análise documental e aplicação de uma *Survey* a equipes de desenvolvimento de *software* em uma empresa pública federal.

Creswell e Plano Clark (2011) definem métodos mistos como um procedimento de coleta, análise e combinação de técnicas quantitativas e qualitativas em um mesmo desenho de pesquisa. No presente estudo especificamente será utilizada a estratégia sequencial exploratória. Segundo Creswell (2012), a estratégia sequencial exploratória é especialmente adequada para desenvolver novos instrumentos de coleta ou aperfeiçoar instrumentos já existentes.

Segundo Gil (2002), a pesquisa exploratória tem como objetivo proporcionar maior familiaridade com o problema, a fim de torna-lo mais explícito ou a construir hipóteses. Ainda segundo o mesmo autor a pesquisa exploratória tem como objetivo principal o aprimoramento de ideias ou a descoberta de intuições.

De acordo com Gil (2002), a pesquisa descritiva tem o objetivo descrever a característica de determinada população ou fenômeno ou o estabelecimento de relações entre variáveis, dando um panorama em um determinado período de tempo.

Quanto aos procedimentos utilizou-se a analise bibliográfica, documental e a *Survey*. Segundo Gil (2002), a análise documental trilha os mesmos caminhos da pesquisa bibliográfica, porém a pesquisa documental utiliza fontes mais diversificadas e dispersas, sem tratamento analítico.

A *Survey* refere-se à obtenção de dados ou informações sobre características, ações ou opiniões de determinado grupo de pessoas (representantes de um público alvo) por meio de um instrumento de pesquisa, normalmente um questionário. A *Survey* apresenta como principais características o interesse em produzir descrições quantitativas de uma população e fazer o uso de um instrumento predefinido (FREITAS ET AL., 2000).

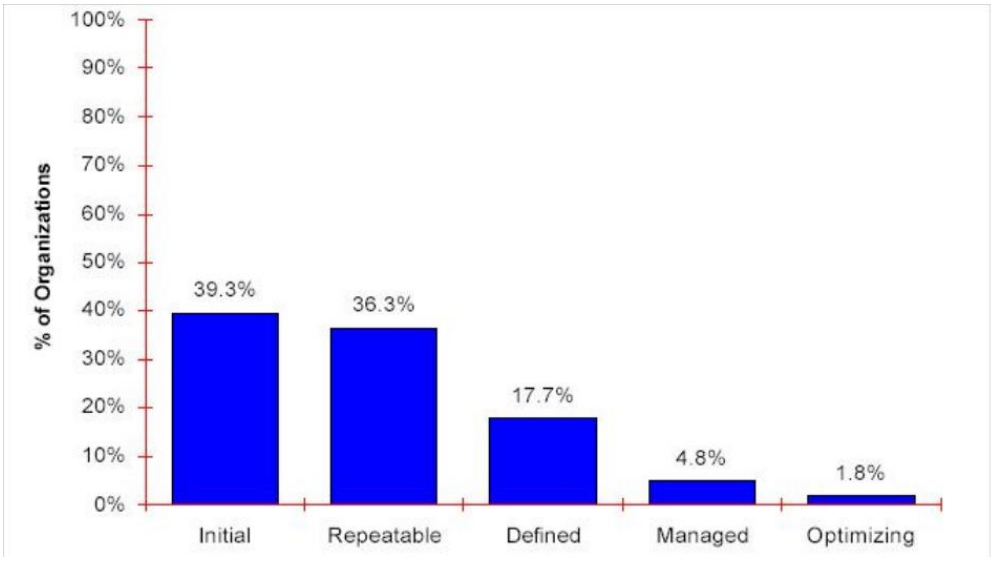
De acordo com Hair Jr et al. (2007), a população alvo é um grupo completo de elementos relevantes que apresentam informações que o projeto se propõe a coletar. Para o presente trabalho, o público alvo da pesquisa é desenvolvedores e gestores de equipes que tenham atuado em projetos que utilizaram o programa de desenvolvimento de *software* corporativo.

A escolha dos respondentes foi por conveniência. Hair Jr et al. (2007) afirmam que a amostra por conveniência envolve a seleção de elementos de amostras que estejam mais disponíveis para tomar parte no estudo e que podem oferecer informações necessárias.

3.2 Ambiente de pesquisa

O SERPRO decidiu adotar um processo de desenvolvimento de *software* acompanhando a tendência das empresas no final da década de 90 e início do ano 2000. Analisando a Figura 11, obtida no relatório *Software Engineering Measurement and Analysis Team* (2000), observa-se que o mercado ao longo do tempo estava buscando métodos mais eficientes de desenvolvimento de *software*, mais especificamente o *Software Engineering Institute - Software Capability Maturity Model* (SW-CMM).

Figura 11 Nível de maturidade das empresas



Fonte: Software Engineering Measurement and Analysis Team (2000)

A pesquisa apontava que, do total de 870 empresas pesquisadas, 39,3% estavam no nível 1 de maturidade do SW-CMM, o Inicial. Dessas, aproximadamente 56% eram empresas militares ou federais. Essa pesquisa também apontou que a tendência de empresas aderentes ao SW-CMM nível 2 aumentava gradativamente desde 1987.

Na área de gestão, mais especificamente, a pesquisa mostrou que quase 90% das empresas atendiam as áreas-chave de planejamento e acompanhamento de projetos de *software*. Essa mudança de atitude das empresas ocorreu devido ao fato de o cliente estar mais exigente quanto à qualidade do *software*. Além disso, a complexidade das empresas estava aumentando e a concorrência estava mais acirrada devido ao avanço da globalização.

O SERPRO é uma empresa pública de prestação de serviços de tecnologia da informação para o Governo Federal. Foi criado em 1o. de dezembro de 1964, para ajudar no processo de arrecadação fiscal do Ministério da Economia.

A empresa possui cerca de 9.000 empregados distribuídos por mais de 300 municípios do país. Seu mercado de atuação é o segmento de finanças públicas. Também atua nos segmentos das ações estruturadas e integradoras da Administração Pública Federal. Além disso, a empresa é credenciada como Autoridade Certificadora (AC) da Infraestrutura de Chaves Pública Brasileira – ICP Brasil.

Possui uma sede central (Brasília) e 10 regionais, situadas em: Brasília, Belém, Belo Horizonte, Curitiba, Fortaleza, Porto Alegre, Recife, Rio de Janeiro, Salvador, São Paulo.

A década de 70 foi marcada pelo aumento do nível de sofisticação dos *softwares* e hardwares e o consequente aumento de esforço para a manutenção e o desenvolvimento dos mesmos. Nessa época, o SERPRO já havia implantado o Sistema Automatizado de Imposto de Renda e outros grandes sistemas como o Cadastro Nacional de Veículos Automotores.

Não existia, na empresa, um processo unificado de desenvolvimento de *software*, mas começavam a surgir as primeiras tentativas de se criar métodos para auxiliar esse processo.

Com o passar do tempo e com a evolução das técnicas de desenvolvimento de *software*, surgiu na empresa a necessidade de um grupo de trabalho composto por técnicos de várias unidades da empresa para a criação de uma metodologia de desenvolvimento de sistemas.

Uma série de trabalhos e documentos foram feitos, mas na época o projeto não teve continuidade.

Na década de 90, porém, outro grupo de trabalho foi instituído para a criação de um processo de desenvolvimento de *software*, chamado MUSE - Metodologia Única de Serviços.

Neste caso, a ideia era que o processo atendesse as áreas de desenvolvimento de *software* e a área de negócio. Segundo diagnóstico realizado pela empresa, o processo não obteve sucesso devido a três aspectos principais:

- Gerencial – a estratégia de implantação teve algumas falhas na divulgação da metodologia e no treinamento do corpo técnico. Não houve um acompanhamento eficaz quanto ao uso da MUSE.
- Comportamental – houve uma resistência considerável quanto ao uso da nova metodologia.
- Técnico – a nova metodologia não estava totalmente adaptada às necessidades da empresa, pois não contemplava a manutenção de sistemas, o que na época representava cerca de 80% do trabalho realizado na empresa.

A decisão de implantação de um processo de desenvolvimento ocorreu na metade do ano 2000. Com o objetivo de acompanhar as tendências do mercado, aprimorar a qualidade dos serviços, reduzir custos e contornar a defasagem do conhecimento técnico, foi criado o Processo de Modernização do Desenvolvimento (PMod). Esse projeto adotou o método IDEAL como metodologia para implantar o SW-CMM e tinha como objetivo sistematizar o processo de desenvolvimento, criar uma base de soluções empresariais, instituir a gestão do processo de desenvolvimento e explicitar a concepção da solução.

Baseando-se no relatório *Software Engineering Measurement and Analysis Team* (2000), no qual apontava que a média para as empresas atingirem o CMM nível 2 era aproximadamente 25 meses, foi feito um planejamento para que a empresa estivesse aderente ao nível 2 do CMM em dois anos (julho de 2000 à dezembro de 2002).

Foram definidas cinco fases distintas para o PMod:

1. Infraestrutura: com o objetivo de mostrar a importância do processo de desenvolvimento esta fase caracterizou-se pela contratação de consultoria de CMM, etapas de sensibilização de funcionários envolvidos no processo de desenvolvimento de *software*, definição de grupos de trabalho com experiência nas áreas-chave de desenvolvimento de *software*, definição de políticas para orientar o

processo de gestão e engenharia de *software*, identificação de processos e padrões que atendessem aos requisitos do CMM e a necessidade da empresa, e finalmente um levantamento das práticas existentes.

2. Definição Corporativa: foram definidos modelos e normas, grupos de suporte, base histórica, ferramentas e técnicas.
3. Definição por unidade de gestão: Cada unidade ficou responsável por executar o PMoD através da criação de grupos. Os profissionais dessas áreas foram treinados em planejamento e acompanhamento de *software* com foco no CMM. As unidades foram avaliadas para auxiliar na priorização das ações dentro das superintendências, identificando pontos fortes e pontos de melhoria. Cada superintendência era responsável pela criação de planos de ação, normas e modelos coerentes com o PMod.
4. Coordenação e supervisão do projeto PMod: Esta fase foi criada para garantir a coerência entre o PMod e seus subprojetos. Composto pelo diretor supervisor juntamente com o grupo executor do projeto, o primeiro ficou responsável pela coordenação geral do programa de melhoria e o segundo conduzia as ações de melhoria ao acompanhar o andamento dos grupos de trabalho de cada unidade.

Paralelamente à instalação da infraestrutura do PMoD, foi organizado um workshop que percorreu todas as regionais divulgando as melhores práticas preconizadas pelo projeto.

Atendendo à terceira atividade prevista no plano de ação, em cada unidade foi criado um plano de ação que focava nas oportunidades de melhoria identificadas no diagnóstico de acordo com a área-chave correspondente. Neste plano de ação, foram definidas quatro fases distintas.

- Conscientização: foi utilizado um workshop para mostrar aos funcionários das superintendências os objetivos da gerência de projetos, papéis e responsabilidades.

- Implantação: foram documentadas e institucionalizadas as práticas existentes, definidas e implantadas as práticas não existentes de acordo com o CMM nível 2.
- Consolidação: acompanhar a evolução do processo para tornar os projetos 100% aderentes ao CMM.
- Refinamento: ajustar e melhorar as práticas existentes.

Na fase de implantação, em cada uma das superintendências, foi feito um levantamento prévio das práticas existentes na empresa. Este levantamento levou em consideração os compromissos, as atividades, as medições e as verificações exigidas para as empresas aderentes ao CMM nível 2.

O diagnóstico identificou que a maioria das unidades de gestão não possuíam uma atividade de estimativa e métrica desenvolvida. Assim, a atividade de acompanhamento, quando existia, era geralmente utilizada apenas para gerir o tempo de desenvolvimento dos projetos de *software*.

O relatório também apontou a necessidade de treinamento para maioria dos desenvolvedores em técnicas e ferramentas que auxiliassem o planejamento e o acompanhamento de projetos de *software*. Posteriormente, os envolvidos com a macro atividade foram treinados em conceitos relacionados a atividade, técnicas de estimativa, MS Project e ferramentas internas utilizadas para auxiliar na atividade.

Durante a adoção do SW-CMM no ambiente corporativo, o processo de melhoria de *software* continuou evoluindo culminando na criação do Processo Serpro de Desenvolvimento de Soluções (PSDS), que foi implantado em 2001.

O objetivo do PSDS era fornecer um processo de *software* padronizado às equipes de desenvolvimento.

Era um processo prescritivo, com a estrutura baseada em macro atividades e foi baseado em elementos do *Rational Unified Process* (RUP) - conjunto de processos de engenharia de *software*, de propriedade da IBM, que oferece uma abordagem baseada em disciplinas para atribuir tarefas e responsabilidades dentro de uma organização de desenvolvimento; *Project Management Body of Knowledge* (PMBoK) – principalmente na macroatividade de gestão de

projetos e práticas definidas nos modelos *Capability Maturity Model* (CMM) e *Capability Maturity Model Integration* (CMMI).

O PSDS foi utilizado pelo SERPRO até agosto de 2019, quando foi substituído pelo processo 3.9 - Criar e Sustentar Soluções Digitais. O novo processo faz parte da cadeia de valor do Serpro, inspirando-se em métodos ágeis, nas lições aprendidas com o Programa UNO e em outras iniciativas organizacionais do passado. É menos prescritivo e tem os objetivos de manter a governança, entregar valor e reduzir os desperdícios, mantendo o foco nos elementos principais do provimento de soluções digitais.

O UNIFICA, originalmente nomeado UNO, foi oficializado em outubro de 2017 e foi idealizado como uma forma de trabalho baseada em princípios ágeis na entrega de *software* sob medida.

A ideia evoluiu ao longo de 2017 e surgiu a partir da pesquisa de satisfação dos clientes do ciclo estratégico de 2016, que apontou oportunidades de melhoria na entrega de *softwares* sob medida. A proposta do UNIFICA foi trabalhar com soluções ponta a ponta; desta forma, as pessoas envolvidas na solução deveriam atuar junto de pessoas de diversas áreas dentro da empresa, convergindo para um time auto organizado e adaptado às necessidades dos clientes.

O processo UNIFICA tem os seguintes direcionadores: visão ponta a ponta, orientação para o cliente, quebra de silos e barreiras internas, base em métodos ágeis, orientador e pouco prescritivo e ênfase em automação.

Ele serve para apoiar os times de solução para atender os clientes nas fases de prospecção, concepção e implantação da solução de *software*, que agreguem valor de negócio para os clientes. Os times auto organizados e multidisciplinares, compostos por cliente e profissionais, atuam de forma integrada e garantem que as entregas sejam feitas com qualidade, dentro do prazo e com valor agregado.

O processo UNIFICA é baseado em métodos e frameworks ágeis como *Scrum*, Kanban e XP e tem como direcionador o foco no cliente. Além disso, o processo UNIFICA está baseado em cinco princípios norteadores:

1. Foco nas pessoas: esse princípio cuida de aspectos relacionados aos membros da equipe e tem como características estímulo à confiança entre as pessoas, clareza na comunicação, empoderamento da equipe e motivação dos membros da equipe.

2. Foco na satisfação do cliente: esse princípio tem como objetivo melhorar a comunicação com o cliente, melhorar a satisfação do cliente por meio de entregas antecipadas e contínuas da solução e também efetuar ações para garantir a qualidade da solução.
3. Atingir efetividade financeira: esse princípio tem como objetivo compatibilizar as entregas da solução com a contrapartida financeira, gerando insumos que agilizem o processo de faturamento. Esse princípio também tem o objetivo de alinhar os contratos e os aditivos com as práticas e métodos ágeis.
4. Promover a inovação: esse princípio tem o objetivo de estimular o espírito propositivo dos membros do time da solução, através do uso da criatividade e da utilização de novas tecnologias para aprimorar a solução.
5. Melhoria contínua: tem como objetivo acompanhar os resultados e as medidas gerados pelo time da solução, bem como pela própria solução. Esse princípio estabelece que devem ser feitos ajustes para que o processo, o time e a solução se adaptem ao contexto que atenda às necessidades dos clientes. O time da solução deve refletir afim de buscar formas de aumentar seu desempenho.

O processo UNIFICA tem como propósito entregar soluções de *software* que gerem valor de negócio para os clientes, apoiando os times da solução e estimulando uma postura propositiva, para que as necessidades dos clientes sejam atendidas, gerando assim valor e satisfação.

O processo UNIFICA está baseado em cinco fundamentos:

1. Visão de Ponta a Ponta, quebrando silos organizacionais
2. Princípios e Práticas Ágeis
3. Pensamento Enxuto, resultando em um processo orientador e pouco prescritivo
4. Ênfase na Automação de Tarefas, processos e atividades
5. Liderança Transformadora e Inspiradora.

A Figura 12 representa os propósitos, princípios e fundamentos do programa UNIFICA. Esse modelo foi inspirado nas representações da *House of Lean* e da adaptação construída para o *SAFe*.

O *framework* SAFe, utilizado para escalar os métodos ágeis para o nível empresarial foi proposto por Leffingwell (2019) e baseado no *Lean Thinking*, *Scrum* e XP.

O SAFe encontra-se na versão 4.6 e apresenta quatro configurações possíveis: *Essential* SAFe, *Large Solution* SAFe, Portfolio SAFe e *Full* SAFe. Essa característica permite que cada organização adapte-se ao framework de acordo com suas necessidades. No *framework* SAFe, os times múltiplos são agrupados em *Agile Release Train* (ART), que é definido como equipes de equipes trabalhando junto em um fluxo de valor da empresa.

Na configuração Portifólio, o SAFe está dividido em três níveis:

- Portifólio: define o tema de investimento do projeto e as iniciativas empresariais que derivam do tema de investimento. O tema de investimento está relacionado ao local em que o cliente irá investir para ter a solução de *software*.
- Programa: define as características do sistema como serviços que deve fornecer e os requisitos não funcionais da aplicação. Neste nível, é desenvolvida a visão do sistema e são planejadas as entregas de versão de *software*.
- Time: neste nível, são especificados as histórias de usuário (funcionalidade do sistema) e os critérios de aceitação das mesmas. Também é no nível Time que é realizado o planejamento, a execução e a retrospectiva de uma Sprint.

Figura 12 Representação casa do UNIFICA

O processo UNIFICA foi validado em projetos pilotos reais junto a clientes. Os primeiros pilotos foram os projetos para os sistemas: Porto Sem Papel (PSP) – sistema estruturador utilizado para facilitar a análise e a liberação de mercadorias nos portos brasileiros; SIAPE- sistema de abrangência nacional para gestão de folha de pessoal dos servidores públicos; SCDP – sistema para gestão de controle de diárias e passagens de viagens de interesse da administração em território nacional ou internacional.

As informações do processo 3.9 - Criar e Sustentar Soluções Digitais estão disponíveis no portal UNIFICA, que é um complemento ao mapeamento da cadeia de valor e possui elementos como: princípios, restrições, fluxos, atividades, práticas, temas, artefatos, ferramentas, procedimentos operacionais e métricas, para apoiar os times das soluções do Serpro na prospecção, proposição e estruturação de ideias e tarefas que resultem na construção e sustentação de soluções digitais. O conteúdo é evoluído e atualizado constantemente a partir das experiências dentro do Serpro e dos novos conhecimentos de mercado, sempre buscando o maior valor de negócio aos clientes e ao próprio Serpro.

3.3 Procedimentos metodológicos

Para o presente trabalho, decidiu-se utilizar um questionário como instrumento de coleta de dados, o mesmo consta no Apêndice A.

Segundo Hair Jr. et al. (2007), o questionário consiste em um instrumento para medir as características de indivíduos, organizações, eventos ou outros fenômenos. Assim, o questionário utilizado no trabalho foi baseado no estudo elaborado por Venkatesh et al. (2003).

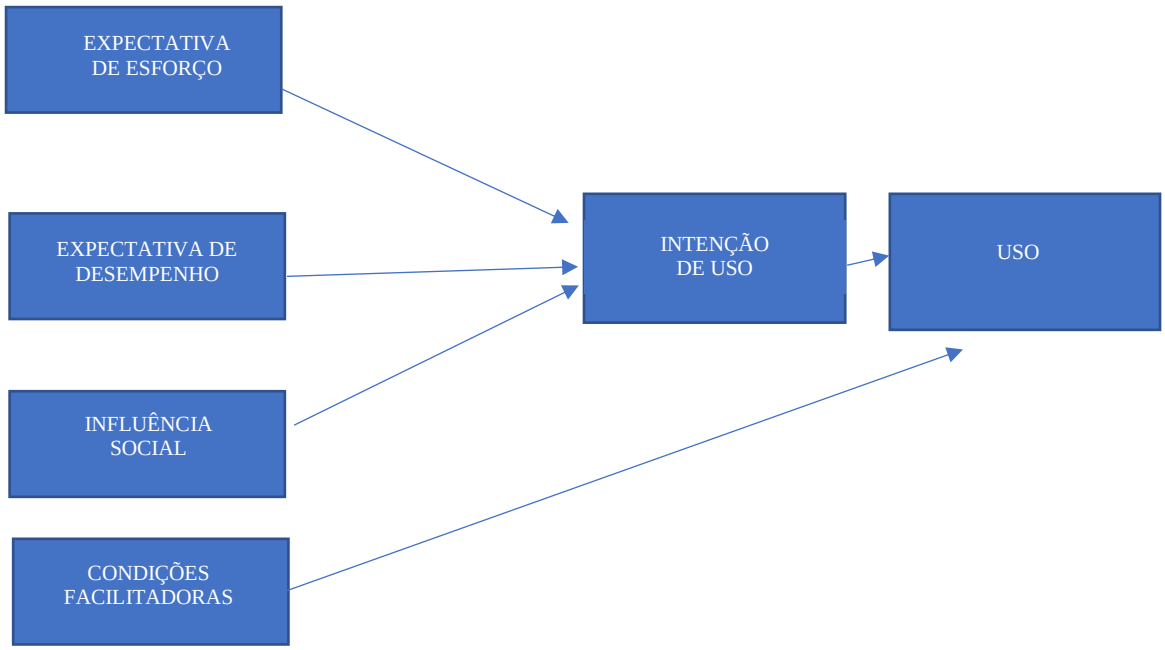
Em relação ao formato do instrumento de coleta, o mesmo foi dividido em três blocos.

O primeiro bloco contém informações sobre o termo de consentimento, objetivo da pesquisa, garantia da confidencialidade das respostas, bem como as questões relacionadas a caracterização do respondente. Nesse primeiro bloco, o objetivo foi identificar gênero, idade, grau de instrução, cargo, tempo de experiência em desenvolvimento de *software* e tempo de trabalho no SERPRO.

O segundo bloco contém questões relacionadas à atividade profissional do respondente. Essa parte do questionário contém perguntas relacionadas a linguagem de programação que utiliza, tempo de experiência em desenvolvimento ágil, tempo de experiência em desenvolvimento ágil no SERPRO, quantidade de projetos ágeis entregues no SERPRO, tempo de utilização do processo de desenvolvimento de *software* UNIFICA e quantidade de projetos entregues utilizando o processo de desenvolvimento de *software* UNIFICA.

O terceiro bloco possui questões relacionadas ao uso do modelo de desenvolvimento de *software* UNIFICA, e foi elaborado com base no estudo proposto por Venkatesh et al. (2003) para avaliar a aceitação de uso de tecnologia (Figura 13).

Figura 13 Modelo UTAUT adaptado ao estudo



Fonte: Autor

Para o presente trabalho, foram considerados os constructos Expectativa de Desempenho, Expectativa de Esforço, Influência Social, Condições Facilitadoras e Intenção de Uso e Uso. O moderador voluntariado não foi considerado, pois não existe a opção de utilizar o processo por parte de membros da equipe já que é uma decisão gerencial.

Para cada constructo foram elaboradas entre três e quatro afirmações (Quadro 6, 7,8,9 e 10), com adaptações de palavras para o contexto de desenvolvimento de *software*, a partir da proposta de Venkatesh et al. (2003). Todas as questões foram medidas com uma escala tipo Likert de 6 itens, considerando as opções “discordo totalmente”, “discordo”, “discordo parcialmente”, “concordo parcialmente”, “concordo” e “concordo totalmente”. Além disso, foi disponibilizada a opção “não sei / não tenho opinião formada” para o caso do respondente não se sentir seguro para responder a questão.

.

Quadro 6 Constructos Expectativa de desempenho e itens do instrumento de pesquisa

Constructo	Itens do instrumento de pesquisa
Expectativa de Desempenho	ED1- O modelo de desenvolvimento de *software* UNIFICA é útil para eu usar em projetos de desenvolvimento de *software*.
	ED2- O modelo de desenvolvimento de *software* UNIFICA me ajuda a executar as tarefas relacionadas ao desenvolvimento de forma mais rápida.
	ED3- Usar o modelo de desenvolvimento de *software* UNIFICA aumenta minha produtividade relacionada às atividades de desenvolvimento de *software*.

Fonte: Autor

Quadro 7 Constructos Expectativa de esforço e itens do instrumento de pesquisa

Constructo	Itens do instrumento de pesquisa
Expectativa de Esforço	EF1- Foi fácil aprender como aplicar as práticas sugeridas no modelo de desenvolvimento de *software* UNIFICA nos projetos que atuei.
	EF2- Foi fácil para eu me tornar hábil no modelo de desenvolvimento de *software* UNIFICA.
	EF3- Eu considero o modelo de desenvolvimento de *software* UNIFICA fácil de utilizar.

Fonte: Autor

Quadro 8 Constructos Influência social e itens do instrumento de pesquisa

Constructo	Itens do instrumento de pesquisa
Influência Social	IS1- As pessoas com as quais trabalho no SERPRO me incentivam a fazer uso do modelo de desenvolvimento de *software* UNIFICA.
	IS2- O gerente do setor no qual trabalho considera que eu devo utilizar o modelo de desenvolvimento de *software* UNIFICA nos projetos de *software* que atuo.
	IS3- Em geral, minha organização tem apoiado o uso do modelo de desenvolvimento de *software* UNIFICA.

Fonte: Autor

Quadro 9 Constructos Condições facilitadoras e itens do instrumento de pesquisa

Constructo	Itens do instrumento de pesquisa
Condições Facilitadoras	CF1- Eu tenho, à minha disposição, os recursos necessários para utilizar modelo de desenvolvimento de *software* UNIFICA.
	CF2- Eu tenho os conhecimentos necessários para utilizar o modelo de desenvolvimento de *software* UNIFICA nos projetos que atuo.
	CF3- O modelo de desenvolvimento de *software* UNIFICA é compatível com outras tarefas que eu utilizo no desenvolvimento de *software*.
	CF4- Posso obter ajuda de outros colegas quando encontro dificuldades no uso do modelo de desenvolvimento de *software* UNIFICA.

Fonte: Autor

Quadro 10 Constructos Intenção de uso e itens do instrumento de pesquisa

Constructo	Itens do instrumento de pesquisa
Intenção de Uso	IU1 - Eu pretendo utilizar o modelo de desenvolvimento de *software* UNIFICA em projetos futuros.
	IU2 - Eu irei utilizar o modelo de desenvolvimento de *software* UNIFICA em projetos futuros.
	IU3 - Embora possa ser favorável, o uso do modelo de desenvolvimento de *software* UNIFICA certamente não é obrigatório no meu trabalho.
	IU4 - O uso do modelo de desenvolvimento de *software* UNIFICA é pouco relevante nos projetos que atuo.

Fonte: Autor

As hipóteses do presente trabalho foram baseadas no trabalho de Venkatesh et al. (2003), e algumas adaptações foram necessárias neste caso específico.

H1: O constructo Expectativa de Desempenho influencia de forma positiva o constructo Intenção de Uso do UNIFICA.

H2: O constructo Expectativa de Esforço influencia de forma positiva o constructo Intenção de Uso do UNIFICA.

H3: O constructo Influência Social influencia de forma positiva o constructo Intenção de Uso o UNIFICA.

H4: O constructo Condições Facilitadoras influencia de forma positiva o constructo Uso do UNIFICA.

H5: A Intenção de Uso influencia de forma positiva o Uso do UNIFICA.

Após a elaboração do questionário, foi realizado um pré-teste com uma equipe de especialistas do novo processo corporativo de desenvolvimento de *software*. O questionário foi disponibilizado por e-mail por um período de 15 dias para avaliação de aspectos como layout, instruções do instrumento de pesquisa e abrangência, adequações das expressões contidas nos itens e disposição das informações.

As opiniões dos respondentes a respeito do questionário foram obtidas através de entrevistas não estruturadas por telefone em duas ocasiões.

A primeira áudio conferência ocorreu com três participantes e foram discutidos aspectos de objetivo da pesquisa, modelo de aceitação de tecnologia UTAUT, termos utilizados no instrumento de pesquisa que deveriam ser modificados, inclusão de opções de resposta que não estavam sendo utilizadas e um possível público alvo para aplicar o instrumento de pesquisa.

Também foram levantados possíveis indicadores de desempenho que poderiam ser utilizados para avaliar a utilização do novo processo de desenvolvimento ágil por parte das equipes.

A segunda áudio conferência ocorreu com um participante e foram discutidos aspectos do modelo de aceitação da tecnologia UTAUT, termos utilizados no instrumento de pesquisa que deveriam ser substituídos e opções nos itens do instrumento de pesquisa que deveriam ser modificados.

Nas duas reuniões, ficou evidente a preocupação dos especialistas com a quantidade de itens do instrumento de pesquisa que tratavam de informações pessoais dos respondentes. Foi explicado aos mesmos que os itens foram criados de acordo com os constructos propostos no modelo de aceitação UTAUT eram necessários para o grau de aceitação no novo processo de desenvolvimento de *software* pelos respondentes.

Após a alteração do questionário com as sugestões dos especialistas, o mesmo foi submetido a desenvolvedores e gestores.

A coleta de dados foi realizada por meio de um questionário eletrônico criado no Google *Forms*, e foi submetida a equipes lotadas nas regionais de Recife, Salvador, São Paulo, Curitiba, Fortaleza, Brasília e Florianópolis, totalizando um potencial de 227 respondentes, no período de 17/02/2020 até 21/02/2020.

A escolha dessas equipes deveu-se ao fato de serem as que utilizavam o UNIFICA há mais tempo.

Após a coleta de dados, foi realizada uma análise exploratória de banco de dados para verificar as respostas válidas. Para as afirmações IU3 e IU4, foi realizada uma inversão de valores, por se tratarem de afirmações negativas.

Por fim, foi feita a eliminação de respondentes com mais de nove afirmações com valor não sei, relacionadas aos constructos Expectativa de Desempenho, Expectativa de Esforço,

Influência Social, Condições Facilitadoras e Intenção de Uso. Dessa forma, consideraram-se 30 questionários como válidos para análise do modelo estrutural, dentre os 44 obtidos na coleta.

Para a avaliação dos dados coletados no questionário, foi previsto um plano de análise da descrição do perfil da amostra e uma associação entre o perfil da amostra e as dimensões Expectativa de Desempenho, Expectativa de Esforço, Influência Social, Condições facilitadoras e Intenção de Uso, assim como Avaliação da consistência interna do instrumento de pesquisa e verificação das hipóteses do modelo.

Segundo Hair Jr. et al. (2005), confiabilidade é o grau em que um conjunto de indicadores de uma variável latente (constructo) é consistente com suas mensurações. O constructo não pode ser medido diretamente, mas apenas representado ou medido por um ou mais indicadores combinados (HAIR JR. et al., 2005).

De acordo com Bland e Altman (1997) o coeficiente de Alfa de Cronbach (α) é uma medida de confiabilidade utilizada para a avaliação da consistência interna do instrumento de pesquisa para um conjunto de dois ou mais indicadores de constructo.

Para o presente trabalho, a tabela 1 será utilizada para avaliar o grau de confiabilidade interna do instrumento de pesquisa.

Tabela 1 Tabela de interpretação α de Cronbach

α de Cronbach	**Consistência interna**
$\alpha \geq 0,9$	Excelente
$0,7 \leq \alpha < 0,9$	Boa
$0,6 \leq \alpha < 0,7$	Aceitável
$0,5 \leq \alpha < 0,6$	Pobre
$\alpha < 0,5$	Inaceitável

Fonte: Adaptado de Kline (1999)

Os valores de α variam de 0 a 1 e quanto mais próximo de 1, maior é a confiabilidade entre os indicadores. Segundo Kline (1999), para testes cognitivos como o teste de inteligência, é esperado o valor 0,8 para o α de Cronbach e 0,7 para testes de habilidade. Quando se trata de

constructos psicológicos, valores abaixo de 0,7 podem ser esperados devido a diversidade dos constructos que estão sendo medidos.

Para a realização dos testes de hipóteses do presente trabalho, foram realizadas análises de correlação não paramétricas de Rô de Spearman. Segundo Dancey e Reidy (2017), Rô de Spearman é utilizada quando os dados não estão normalmente distribuídos ou quando a amostra possui um pequeno número de participantes. De acordo com Corder e Foreman (2014), a correlação de Spearman é utilizada para comparar a relação entre variáveis ordinais.

Para a interpretação dos valores obtidos do coeficiente de correlação, foi utilizada a tabela 2. A descrição para os valores apresentados na tabela 2 são válidos tanto para valores negativos quanto para positivos.

Tabela 2 Tabela de interpretação dos coeficientes de correlação de Spearman

Rô de Spearman	**Correlação**
1	Perfeita
0,7 – 0,9	Forte
0,4 – 0,6	Moderada
0,1 – 0,3	Fraca
0	sem correlação

Fonte: Adaptado de Dancey e Reidy (2017)

Um relacionamento positivo entre duas variáveis x e y significa que o aumento de pontuação de x tende a ter pontuações altas em y. E pontuações baixas em x tendem a ter baixas pontuações em y. As relações negativas significam que, com o aumento de pontuação em y, tenda a ocorrer baixa pontuação em y (DANCEY e REIDY, 2017).

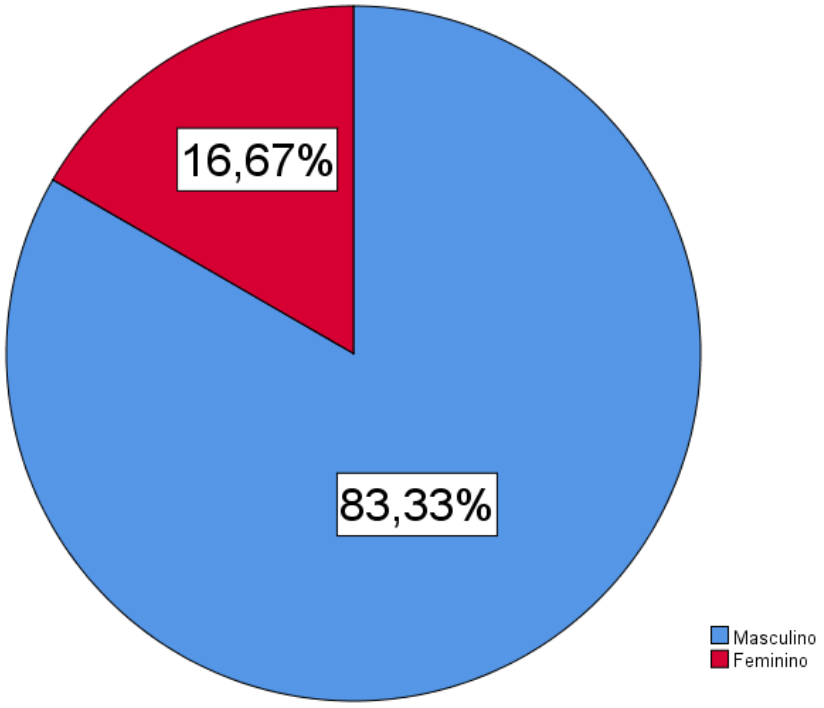
4 RESULTADOS

A seguir, os resultados obtidos nas tarefas descritas no capítulo anterior.

4.1 Caracterização da Amostra

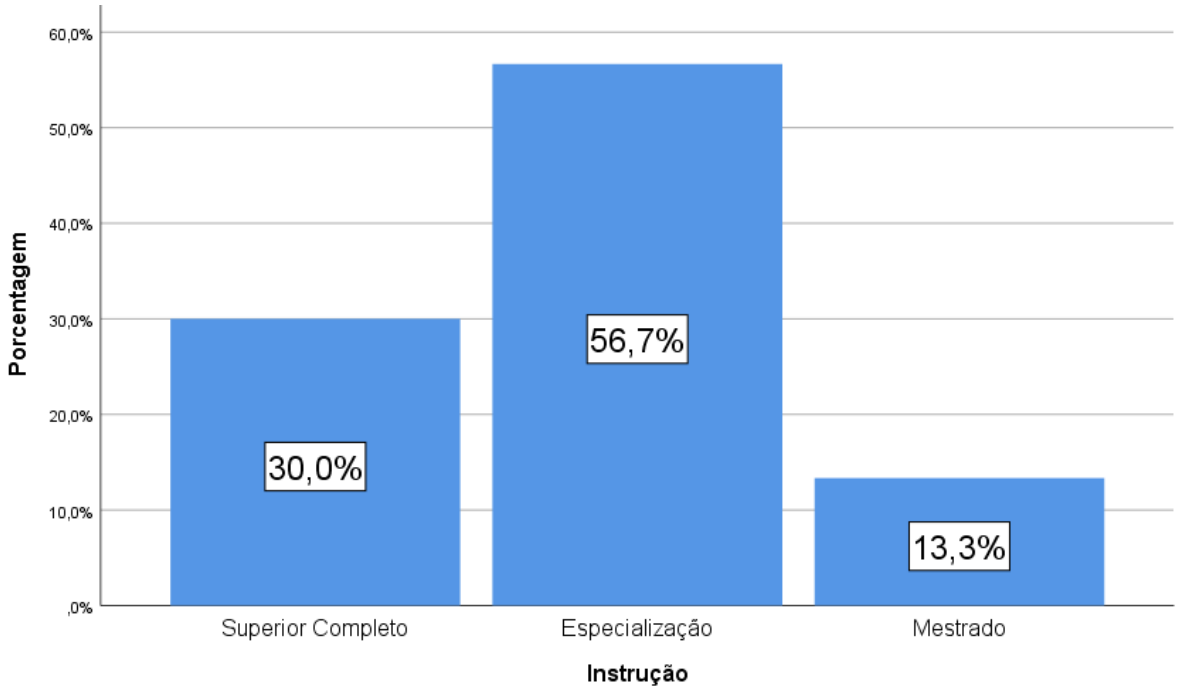
Com relação ao perfil dos respondentes, é possível observar um desequilíbrio no gênero, sendo a maioria do sexo masculino (Figura 14). Os dados corroboram com o estudo de Maia (2016) que apresenta um número decrescente de mulheres nos cursos superiores do campo da computação. Em relação ao grau de instrução dos respondentes, é possível observar que pouco mais de 70% possui um curso de pós graduação em *Lato Sensu* ou *Stricto Sensu* (Figura 15). O fato de não haver pessoas sem ensino superior neste campo é explicado por ser uma exigência do concurso público para ingresso na organização.

Figura 14 Distribuição de Gênero



Fonte: Autor

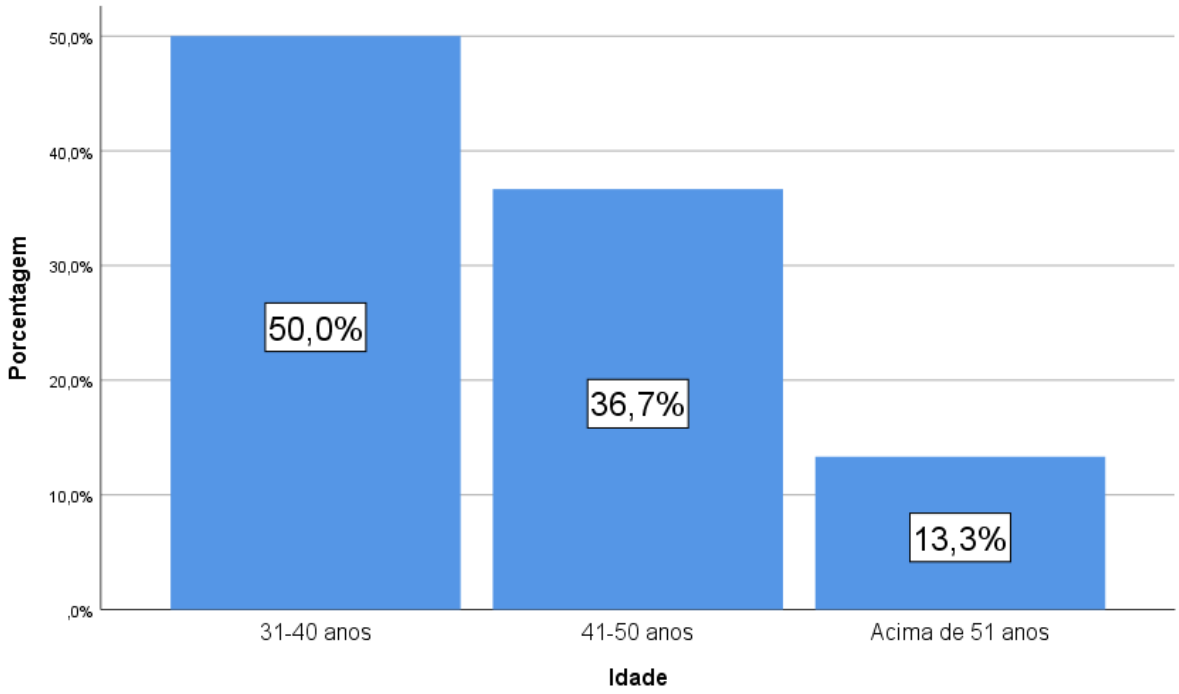
Figura 15 Distribuição por grau de instrução



Fonte: Autor

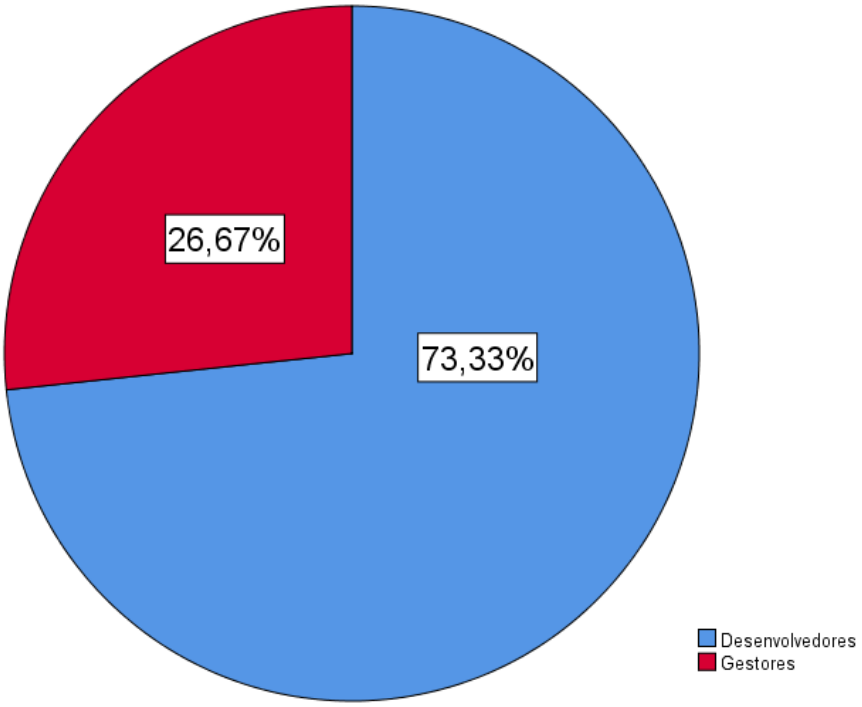
Com relação à faixa etária dos respondentes (Figura 16), pouco mais de 13% dos respondentes estão acima da faixa de 51 anos. Em relação ao quesito função gerencial (Figura 17), foi possível observar que a maioria (cerca de 73% dos respondentes) não exerce função gerencial.

Figura 16 Distribuição por idade



Fonte: Autor

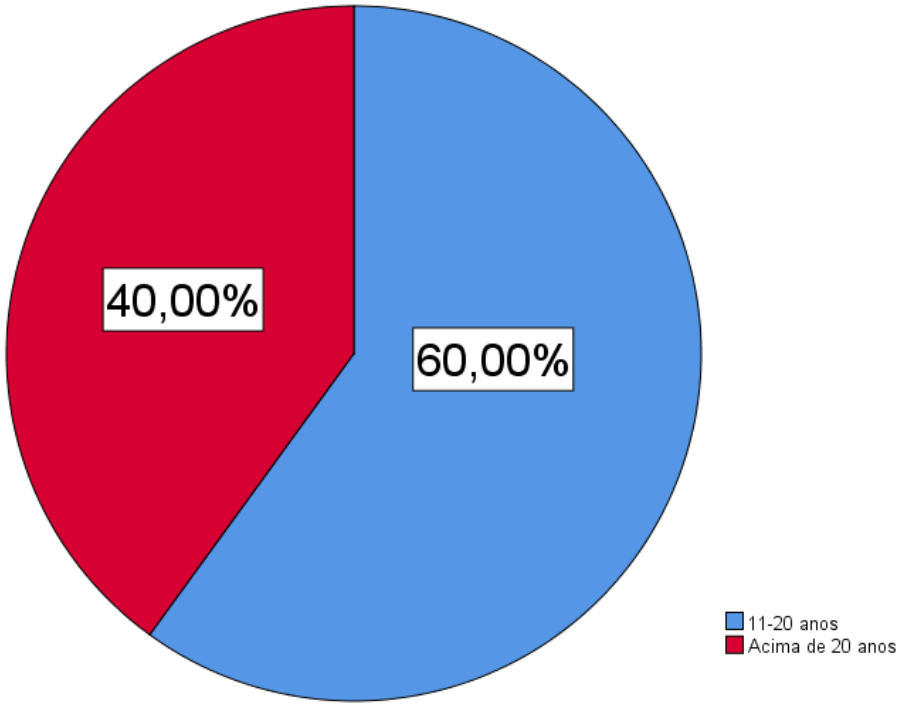
Figura 17 Distribuição por função gerencial



Fonte: Autor

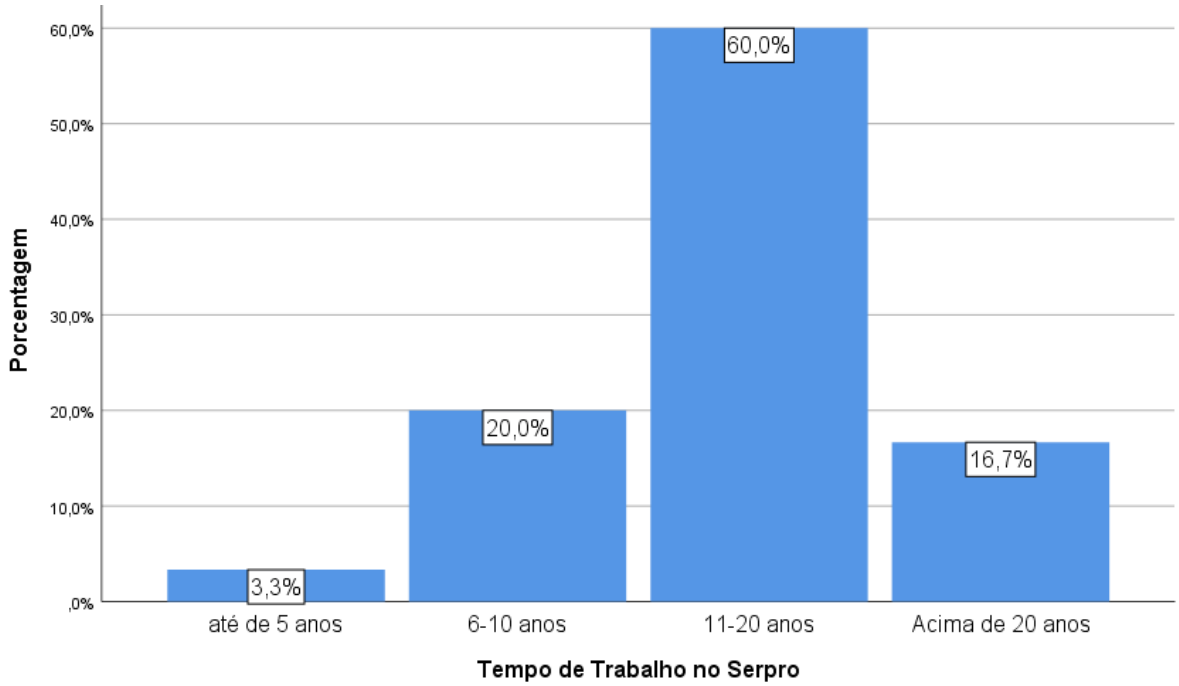
No quesito tempo de experiência em desenvolvimento de *software* (Figura 18), 60% dos respondentes se enquadra na faixa dos 11 a 20 anos. Em relação ao tempo de trabalho no Serpro (Figura 19), cerca de 60% dos respondentes estão na faixa de 11 a 20 anos.

Figura 18 Distribuição por experiência em desenvolvimento de software



Fonte: Autor

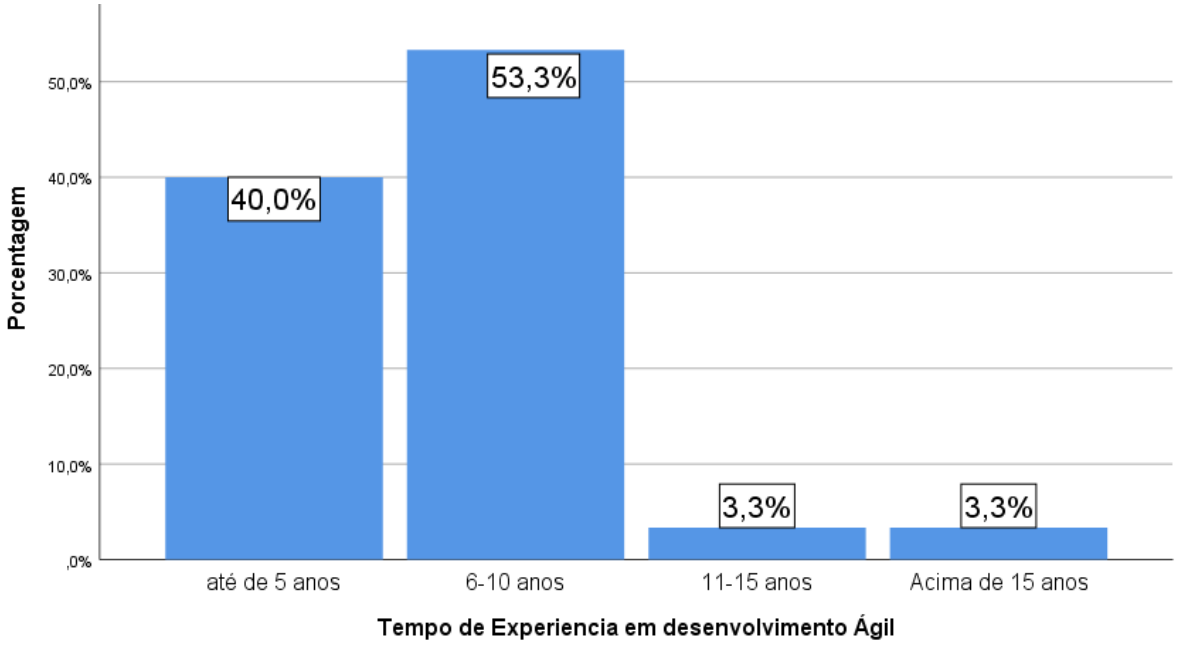
Figura 19 Distribuição por tempo de trabalho no SERPRO



Fonte: Autor

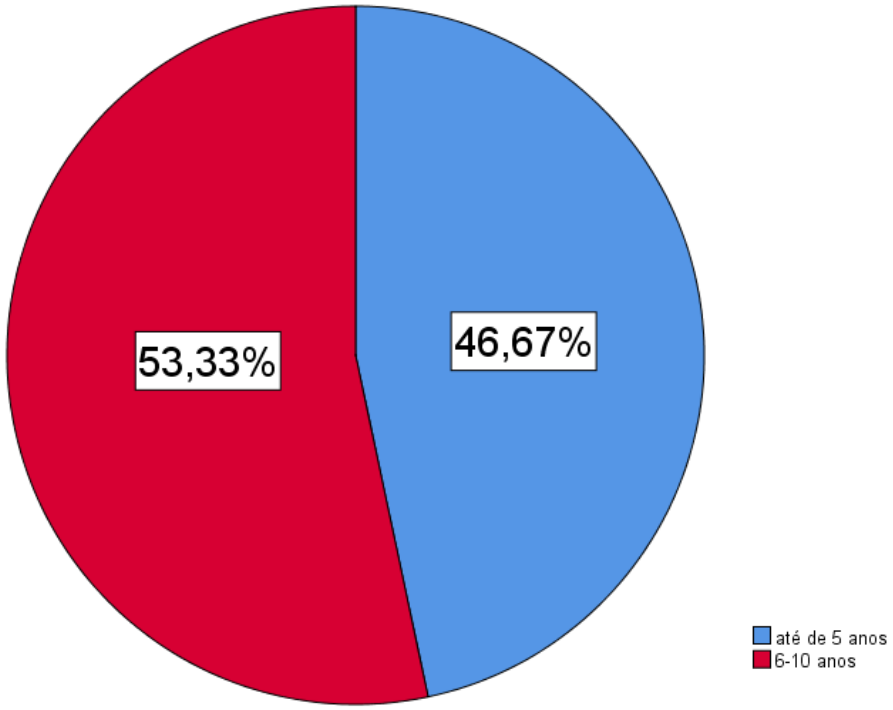
No quesito tempo de experiência (Figura 20) em desenvolvimento ágil, é possível observar que mais de 90% dos respondentes possuem até 10 anos de experiência, assim como no tempo de experiência em desenvolvimento ágil no Serpro (Figura 21).

Figura 20 Distribuição do tempo de experiência em Ágil



Fonte: Autor

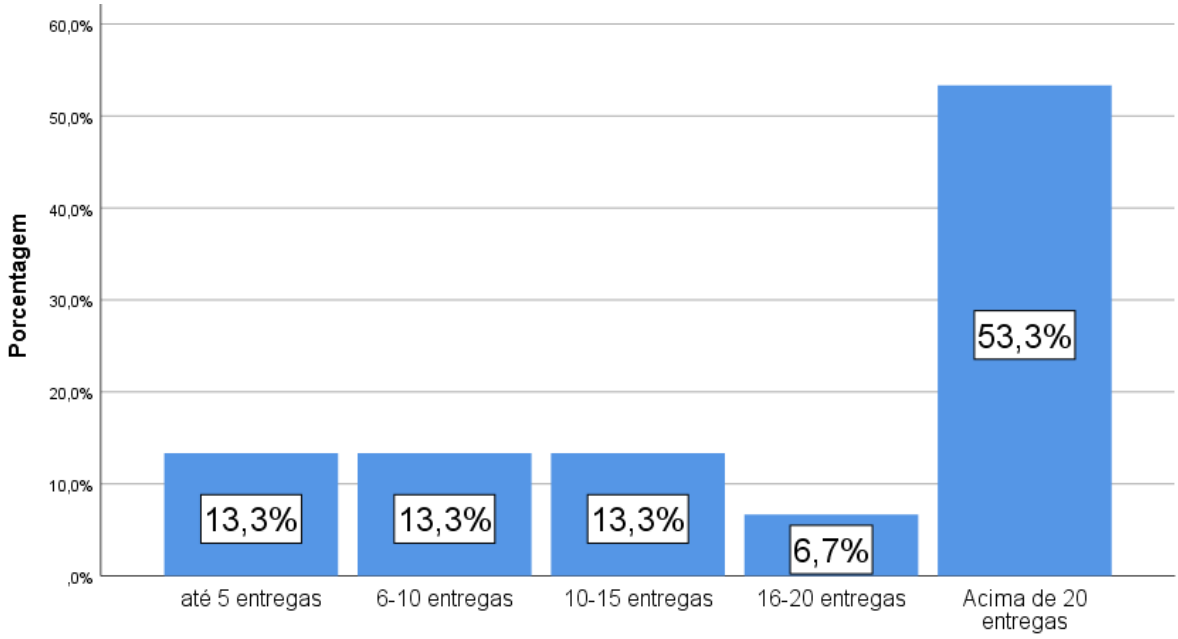
Figura 22 Distribuição de tempo de desenvolvimento Ágil no SERPRO



Fonte: Autor

Com relação à quantidade de entregas utilizando métodos ágeis no Serpro (Figura 22), aproximadamente 53% dos respondentes afirmaram que fizeram mais de 20 entregas. Em relação ao tempo de uso do UNIFICA (Figura 23), é possível observar que apenas 13% dos respondentes têm experiência acima de um ano e pouco mais de 23% dos respondentes afirmaram que fizeram mais de 20 entregas utilizando o UNIFICA (Figura 24).

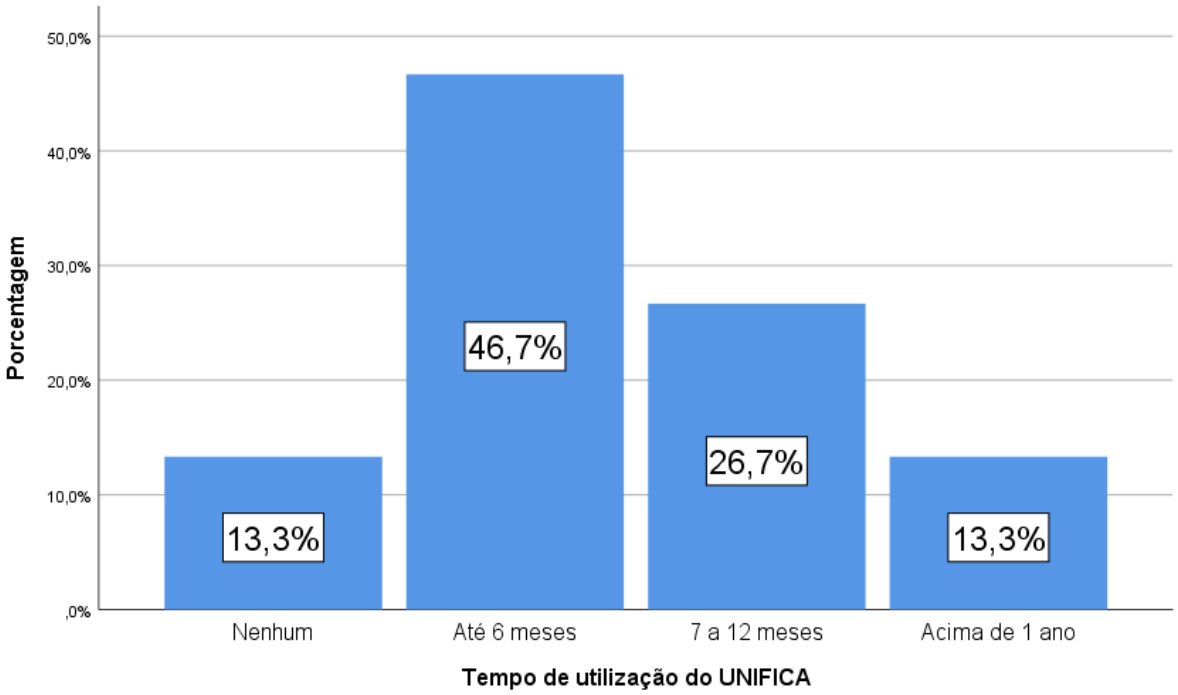
Figura 21 Distribuição de quantidade de entregas utilizando ágil no SERPRO



Fonte: Autor

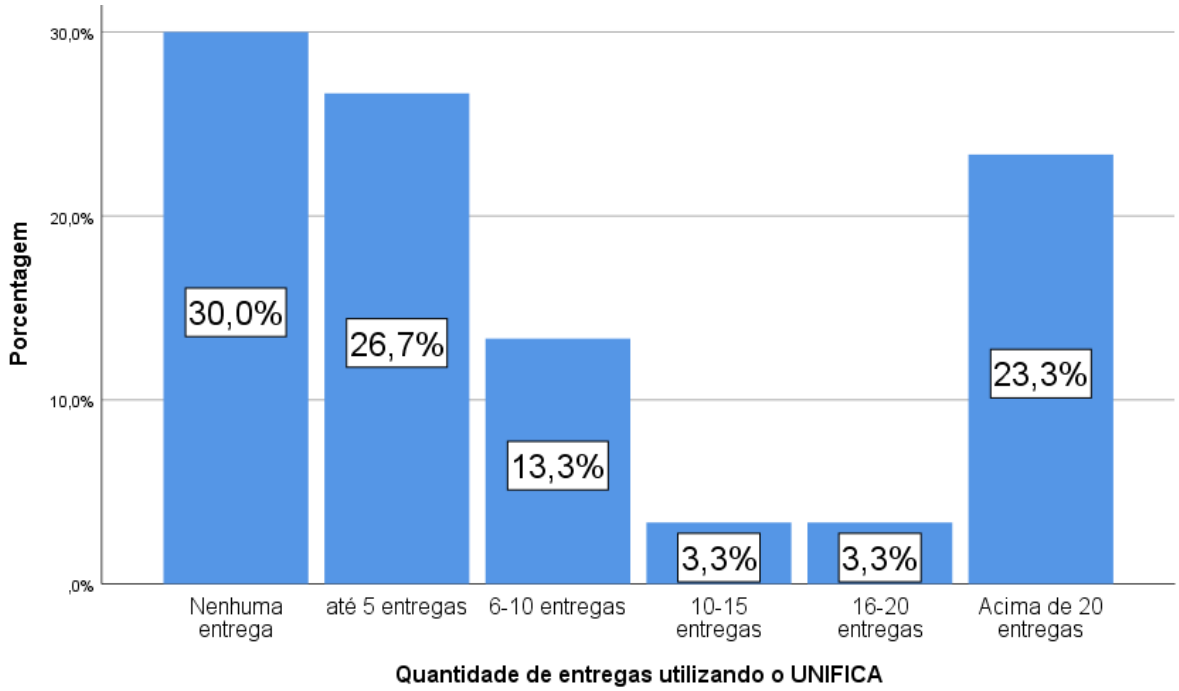
.

Figura 23 Distribuição de tempo de utilização do UNIFICA



Fonte: Autor

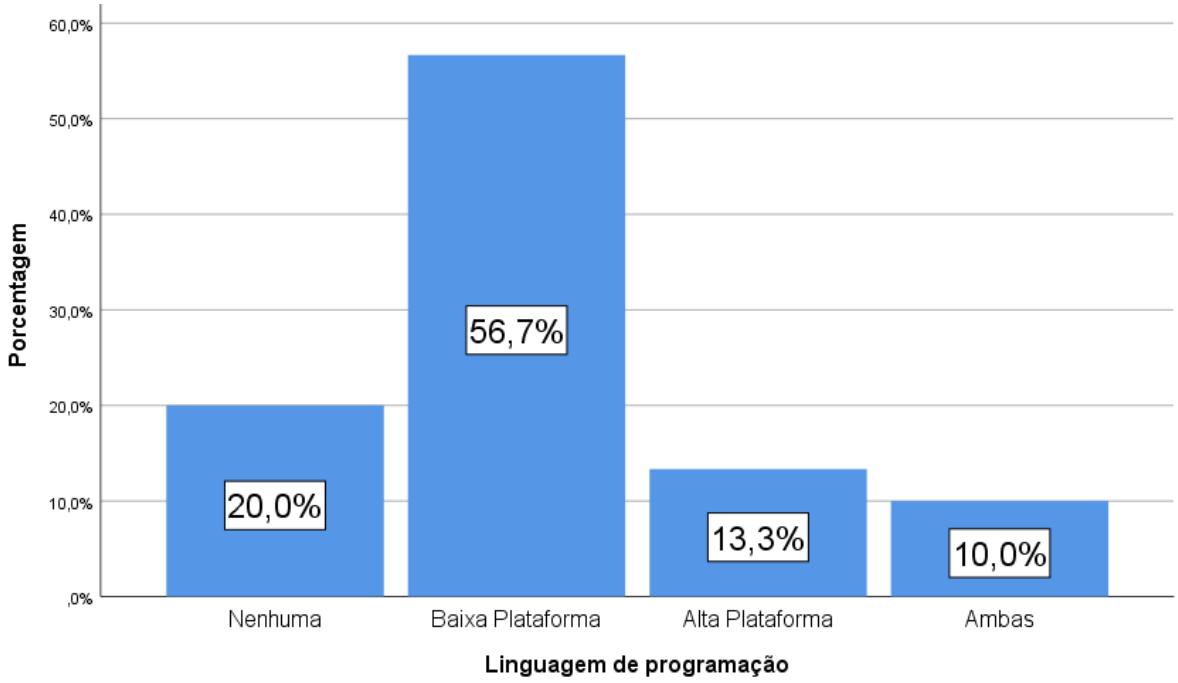
Figura 24 Distribuição de quantidade de entregas utilizando o UNIFICA



Fonte: Autor

Por fim, em relação à linguagem de programação utilizada (Figura 25), pouco mais da metade dos respondentes (56,7%) utilizam linguagem de programação de baixa plataforma em seus projetos.

Figura 25 Distribuição por linguagem de programação utilizada



Fonte: Autor

Ao avaliar o perfil dos respondentes, é possível verificar que se trata de pessoas com um nível elevado de maturidade em desenvolvimento de *software*, inclusive em desenvolvimento ágil de *software*. Porém, fica evidente que a utilização do UNIFICA ainda é incipiente, o que pode levar a distorções na avaliação de aceitação do processo de desenvolvimento proposto pelo SERPRO.

4.2 Avaliação das respostas por Constructo

Para cada constructo, os itens do instrumento de pesquisa foram agrupados (conforme Quadro 6, 6,7,8 e 9), e a mediana e moda da dimensão para as respostas foram calculadas (0 – não sei; 1- Discordo totalmente; 2 – Discordo; 3 – Discordo levemente; 4 – Concordo levemente, 5 – Concordo e 6 Concordo Totalmente) Os itens individuais receberam o mesmo tratamento

A tabela 3 apresenta os cálculos de mediana e moda por dimensão bem como a mediana por afirmação.

Tabela 3 Medianas e moda por itens e dimensões

	ED1	ED2	ED3	EF1	EF2	EF3	IS1	IS2	IS3	CF1	CF2	CF3	CF4	IU1	IU2	IU3	IU4
Mediana Dimensão			4			4			5				5				5
Moda Dimensão			5			4			5				5				5
Mediana Afirmação	5	4	4	4	4	4	4	5	5	4,5	4	5	5	5	5	5	5

Fonte: Autor

De acordo com a tabela 3, é possível verificar que a mediana calculada para cada uma das dimensões avaliadas variou entre 4 e 5 (concordo levemente e concordo). Esse resultado mostra que, para a amostra estudada, existe a aceitação do uso do UNIFICA como processo organizacional de desenvolvimento de *software*.

Ao calcular as medianas dos constructos por categoria (tabela 4), é possível observar diferenças em relação ao resultado geral. Para os casos em que a mediana obtida não foi um número inteiro, foi utilizado o teste de Wilcoxon para determinar a mediana com valor inteiro mais apropriado.

Tabela 4 Medianas dos constructos por categoria

Informação	Níveis	Mediana por Constructo				
		ED	EF	IS	CF	IU
Idade	31-40 anos	4	4	4	4	5
	Acima de 41 anos	4	4	5	5	5
Instrução	Superior Completo	4	4	5	5	5
	Pós graduação	4	4	4	4	5
Função Gerencial	Não	4	4	5	5	5
	Sim	4	4	5	4	5
Tempo Trabalho no SERPRO	Até 10 anos	3	3	5	4	4
	Acima de 11 anos	4	4	5	5	5
Experiência em Ágil no Serpro	Até 5 anos	4	4	5	5	5
	6 - 10 anos	4	4	5	4	5
Tempo de UNIFICA	0 - 6 meses	4	4	5	5	5
	acima de 7 meses	4	4	5	5	5

Fonte: Autor

Quando observamos os dados categorizados por faixa de idade, é possível verificar que os respondentes mais velhos apresentam maior concordância nos constructos Influência Social e Condições Facilitadoras. Esse resultado está em desacordo com o apresentado no estudo de

Venkatesh et al. (2003), o qual afirma que a Influência Social tem efeito mais forte em mulheres mais velhas em situação de uso mandatório e em estágios de pouca experiência.

Em relação ao Grau de instrução, os respondentes com ensino superior apresentam maior concordância em relação aos constructos Influência Social e Condições Facilitadoras.

Os respondentes que exercem função gerencial apresentam uma avaliação diferente dos desenvolvedores em relação ao constructo Condições Facilitadoras.

Quando os dados são analisados categorizando as respostas por tempo de trabalho no SERPRO, é possível observar que os respondentes com menor tempo de trabalho concordam menos que os respondentes com mais tempo de trabalho no SERPRO em relação aos constructos Expectativa de Desempenho, Expectativa de Esforço, Condições Facilitadoras e Intenção de Uso.

Em relação ao tempo de experiência em desenvolvimento ágil de *software* no SERPRO, os respondentes mais experientes apresentam menor concordância com o constructo Condições Facilitadoras.

Por fim, quando os dados são analisados categorizando por tempo de uso do UNIFICA, é possível observar que não houve diferença de avaliação entre os respondentes menos experientes e os mais experientes. Isso pode ser explicado pelo uso incipiente do UNIFICA.

O teste não paramétrico Rô de Spearman (tabela 5) foi utilizado para examinar as significâncias das relações entre todos os constructos.

As colunas da tabela 5 representam os constructos Expectativa de Desempenho, Expectativa de Esforço, Influência Social, Condições Facilitadoras e Intenção de Uso e Uso. As linhas contêm os mesmos constructos e o cálculo da correlação Rô de Spearman e a significância (2 extremidades). N representa o número de casos analisados. A letra ρ representa o p-value. De acordo com Dancey e Reidy (2017), o p-value para um teste estatístico inferencial é a probabilidade de encontrar padrões de resultados em um estudo particular se a hipótese nula for verdadeira.

É possível observar que o coeficiente de correlação das diagonais é igual a 1. Isso ocorre porque representa a correlação do constructo com ele mesmo.

Tabela 5 Correlação não paramétrica Rô de Spearman para os constructos

Correlações

			ED	EF	IS	CF	IU	U
rô de Spearman	ED	Coeficiente de Correlação	1,000	,684**	,528**	,563**	,356	,130
		ρ (2 extremidades)	.	,000	,003	,001	,054	,494
		N	30	30	30	30	30	30
	EF	Coeficiente de Correlação	,684**	1,000	,556**	,532**	,565**	,033
		ρ (2 extremidades)	,000	.	,001	,002	,001	,864
		N	30	30	30	30	30	30
	IS	Coeficiente de Correlação	,528**	,556**	1,000	,549**	,385*	,069
		ρ (2 extremidades)	,003	,001	.	,002	,036	,717
		N	30	30	30	30	30	30
	CF	Coeficiente de Correlação	,563**	,532**	,549**	1,000	,352	,079
		ρ. (2 extremidades)	,001	,002	,002	.	,057	,680
		N	30	30	30	30	30	30
	IU	Coeficiente de Correlação	,356	,565**	,385*	,352	1,000	,055
		ρ. (2 extremidades)	,054	,001	,036	,057	.	,774
		N	30	30	30	30	30	30
	U	Coeficiente de Correlação	,130	,033	,069	,079	,055	1,000
		ρ (2 extremidades)	,494	,864	,717	,680	,774	.
		N	30	30	30	30	30	30

**. A correlação é significativa no nível 0,01 (2 extremidades).

*. A correlação é significativa no nível 0,05 (2 extremidades).

Fonte: Autor

A tabela 5 apresenta oito correlações significativas destacadas com asterisco que estão descritas a seguir.

A correlação entre Expectativa de Desempenho com a Expectativa de Esforço tem valor positivo igual a 0,684 no nível 0,01 (2 extremidades). De acordo com a tabela 1, essa correlação é moderada e mostra que o aumento na Expectativa de Desempenho aumenta a Expectativa de Esforço no uso do UNIFICA. Isso significa que o aumento na percepção de que o uso do UNIFICA melhora o desempenho dos respondentes também aumenta a percepção de que é fácil utilizar o UNIFICA.

A correlação Expectativa de Desempenho com Influência de Social tem valor positivo igual a 0,528 no nível 0,01 (2 extremidades). De acordo com a tabela 1, essa correlação é moderada. Nessa correlação, é possível observar o aumento na percepção de que o uso do UNIFICA melhora o desempenho dos respondentes também aumenta a percepção sobre a opinião (de aprovação ou reprovação) das outras pessoas sobre o uso do UNIFICA.

A correlação entre Expectativa de Desempenho e as Condições Facilitadoras têm valor positivo igual a 0,563 no nível de 0,01 (2 extremidades). Considerada moderada (conforme tabela 1), nessa correlação é possível observar que o aumento na percepção de que o uso do UNIFICA melhora o desempenho dos respondentes também aumenta a percepção sobre as condições de suporte que facilitam o uso do UNIFICA.

A correlação entre Expectativa de Esforço e Influência Social é positiva, com valor igual a 0,556 no nível de 0,01 (2 extremidades). A correlação é considerada moderada (tabela 1), e significa que o aumento na percepção da facilidade de uso do UNIFICA aumenta a percepção da importância da opinião de outras pessoas sobre o uso do UNIFICA.

A correlação Expectativa de Esforço com Condições Facilitadoras tem valor positivo igual a 0,532 (moderada, conforme tabela 1) no nível de 0,01 (2 extremidades). Com essa correlação, é possível observar que o aumento na percepção da facilidade de uso do UNIFICA aumenta a percepção sobre as condições de suporte que facilitam o uso do UNIFICA.

A correlação entre Expectativa de Esforço com Intenção de Uso tem valor positivo igual a 0,565 (moderada, conforme tabela 1) no nível de 0,01 (2 extremidades). Nesse caso, o aumento na percepção da facilidade de uso do UNIFICA aumenta o grau em que o respondente se sente motivado a utilizar o UNIFICA.

A correlação da Influência Social com Condições Facilitadoras tem valor positivo igual a 0,565 (moderada, conforme tabela 1) no nível de 0,01 (2 extremidades). Dessa forma o aumento na percepção sobre a opinião (de aprovação ou reprovação) das outras pessoas sobre o uso do UNIFICA aumenta a percepção sobre as condições de suporte que facilitam o uso do UNIFICA.

A correlação entre Influência Social com Intenção de Uso tem valor positivo igual a 0,385 no nível de 0,05 (2 extremidades). Essa correlação é considerada fraca, conforme tabela 1. A correlação indica que o aumento na percepção sobre a opinião (de aprovação ou reprovação) das outras pessoas sobre o uso do UNIFICA aumenta o grau em que o respondente se sente motivado a adotar o uso do UNIFICA.

Não foram identificadas correlações significativas para o constructo Uso.

Quando as correlações são analisadas sob o ponto de vista de desenvolvedores e gestores, é possível observar uma diferença nas correlações encontradas (tabelas 6 e 7).

Tabela 6 Calculo de correlações para desenvolvedores

Correlações

			ED	EF	IS	CF	IU	U
rô de Spearman	ED	Coeficiente de Correlação	1,000	,827**	,538**	,593**	,421	,045
		ρ (2 extremidades)	.	,000	,010	,004	,051	,842
		N	22	22	22	22	22	22
	EF	Coeficiente de Correlação	,827**	1,000	,523*	,497*	,477*	-,101
		ρ (2 extremidades)	,000	.	,013	,018	,025	,655
		N	22	22	22	22	22	22
	IS	Coeficiente de Correlação	,538**	,523*	1,000	,480*	,289	,007
		ρ (2 extremidades)	,010	,013	.	,024	,191	,976
		N	22	22	22	22	22	22
	CF	Coeficiente de Correlação	,593**	,497*	,480*	1,000	,247	-,012
		ρ (2 extremidades)	,004	,018	,024	.	,267	,959
		N	22	22	22	22	22	22
	IU	Coeficiente de Correlação	,421	,477*	,289	,247	1,000	-,249
		ρ (2 extremidades)	,051	,025	,191	,267	.	,264
		N	22	22	22	22	22	22
	U	Coeficiente de Correlação	,045	-,101	,007	-,012	-,249	1,000
		ρ (2 extremidades)	,842	,655	,976	,959	,264	.
		N	22	22	22	22	22	22

**. A correlação é significativa no nível 0,01 (2 extremidades).

*. A correlação é significativa no nível 0,05 (2 extremidades).

Fonte: Autor

Tabela 7 Calculo de correlações para gestores

Correlações

			ED	EF	IS	CF	IU	U
rô de Spearman	ED	Coeficiente de Correlação	1,000	,110	,516	,361	,198	,410
		ρ (2 extremidades)	.	,795	,190	,379	,638	,312
		N	8	8	8	8	8	8
	EF	Coeficiente de Correlação	,110	1,000	,629	,577	,773*	,387
		ρ (2 extremidades)	,795	.	,095	,135	,024	,344
		N	8	8	8	8	8	8
	IS	Coeficiente de Correlação	,516	,629	1,000	,816*	,573	,252
		ρ. (2 extremidades)	,190	,095	.	,013	,138	,547
		N	8	8	8	8	8	8
	CF	Coeficiente de Correlação	,361	,577	,816*	1,000	,665	,304
		ρ (2 extremidades)	,379	,135	,013	.	,072	,464
		N	8	8	8	8	8	8
	IU	Coeficiente de Correlação	,198	,773*	,573	,665	1,000	,652
		ρ. (2 extremidades)	,638	,024	,138	,072	.	,080
		N	8	8	8	8	8	8
	U	Coeficiente de Correlação	,410	,387	,252	,304	,652	1,000
		ρ (2 extremidades)	,312	,344	,547	,464	,080	.
		N	8	8	8	8	8	8

*. A correlação é significativa no nível 0,05 (2 extremidades).

Fonte: Autor

As correlações significativas encontradas para desenvolvedores (tabela 6) foram: ED x EF (forte); ED x IS (moderada); ED x CF (moderada); EF x IS (moderada); EF x CF (moderada); EF x IU (moderada) e IS x CF (moderada).

Cabe destacar que, para os desenvolvedores, o aumento na percepção que o uso do UNIFICA aumenta o desempenho dos mesmos aumenta também a percepção de facilidade de uso do UNIFICA.

Para os gestores (tabela 7), as correlações encontradas foram Expectativa de Esforço e Intenção de Uso (positiva e forte) e Influência Social e Intenção de Uso (positiva e forte).

A correlação EF x IU encontrada foi positiva e forte, significando que, para os gestores da amostra observada, o aumento da percepção de facilidade de uso do UNIFICA também gera um aumento na intenção de uso do UNIFICA.

4.3 Avaliação do Modelo de Mensuração

Para a avaliação do modelo de mensuração para a medida dos constructos do UTAUT foi feito o teste de confiabilidade interna com uso do cálculo de α de Cronbach (tabela 6).

Tabela 8 Calculo de confiabilidade do instrumento de pesquisa

Estatísticas de confiabilidade		
Alfa de Cronbach	Alfa de Cronbach com base em itens padronizados	N de itens
,932	,952	19

Fonte: Autor

Comparando o resultado obtido na tabela 8 com a tabela de referência (tabela 1), a consistência interna é considerada excelente.

A tabela 9 mostra o cálculo de confiabilidade para as afirmações agrupadas por constructos.

Tabela 9 Cálculo de confiabilidade por constructo

Estatísticas de confiabilidade			
Constructos	**Alfa de Cronbach**	**Alfa de Cronbach com base em itens padronizados**	**N de itens**
ED	0,875	0,877	3
EF	0,938	0,940	3
IS	0,793	0,794	3
CF	0,887	0,889	4
IU	0,833	0,833	4
U	0,753	0,889	2

Fonte: Autor

De acordo com a tabela 9, é possível observar que as afirmações agrupadas por constructo também apresentam um nível bom de confiabilidade interna, e as afirmações relacionadas ao constructo Expectativa de Esforço apresentam um excelente nível de confiabilidade interna (conforme tabela 1).

CONSIDERAÇÕES FINAIS

O presente trabalho permitiu que fossem abordados os métodos de desenvolvimento ágil *Scrum*, XP, Kanban e desenvolvimento *Lean* bem como discorrer sobre modelos de uso e aceitação de tecnologia.

Com base no modelo de aceitação de tecnologia UTAUT foi possível explorar, sob o ponto de vista de desenvolvedores e gestores, o nível de aceitação do processo corporativo de desenvolvimento ágil de *software* UNIFICA no SERPRO por meio de uma *Survey*.

O instrumento de pesquisa utilizado na *Survey* foi baseado no trabalho de Venkatesh et al. (2003) com adaptações para a língua portuguesa e ao contexto do SERPRO. Posteriormente, o questionário foi submetido a uma equipe de especialistas para avaliação e foram feitas melhorias no questionário baseadas nas sugestões dos especialistas.

O instrumento de pesquisa definitivo foi aplicado a desenvolvedores e gestores das regionais de Recife, Salvador, São Paulo, Curitiba, Fortaleza, Brasília e Florianópolis.

A partir da análise descritiva dos dados obtidos, foi possível constatar que a maioria dos respondentes é do sexo masculino, fez pós-graduação (*Lato Sensu* ou *Stricto Sensu*), e situa-se na faixa de 30 a 50 anos.

Também foi possível observar que grande parte dos respondentes são desenvolvedores, possuem entre 11 e 20 anos de experiência em desenvolvimento de *software* e de 11 a 20 anos de trabalho no SERPRO, além de 6 a 10 anos de experiência em desenvolvimento ágil.

A maioria dos respondentes têm pouco tempo de experiência de uso do UNIFICA, isso se justifica pelo fato do UNIFICA ser relativamente recente.

Analisando mediana das respostas das dimensões Expectativa de Desempenho, Expectativa de Esforço, Influência Social, Condições Facilitadoras e Intenção de Uso, é possível observar que existe, em geral, uma concordância variando entre concordo levemente e concordo. Sendo que houve maior concordância nas dimensões Influência Social, Condições Facilitadoras e Intenção de Uso.

Quando os dados foram analisados com a segmentação das respostas por categorias (idade, Grau de Instrução, Função Gerencial, Tempo de Trabalho no SERPRO, Experiência em desenvolvimento Ágil no SERPRO e Tempo de UNIFICA), foi possível observar que não houve diferença significativa em relação às respostas analisadas de forma geral.

Cabe destacar que os respondentes com até 10 anos de trabalho no SERPRO discordam levemente das dimensões Expectativa de Desempenho, Expectativa de Esforço e Condições Facilitadoras. Dessa forma, para os respondentes dessa categoria, o UNIFICA não contribui com aumento de desempenho, não é fácil de usar e não existe suporte para o uso do UNIFICA.

Quando as respostas são analisadas dividindo os respondentes entre desenvolvedores e gestores, é possível verificar uma leve diferença nas respostas referentes ao constructo Condições Facilitadoras. Existe uma percepção maior pelos desenvolvedores da existência de recursos que ajudam a utilizar o UNIFICA.

Para validação das hipóteses do presente estudo, foram observadas as correlações dos constructos. A validação das hipóteses é válida somente para a amostra observada.

A primeira hipótese do estudo - O constructo Expectativa de Desempenho influencia de forma positiva o constructo Intenção de Uso do UNIFICA - não foi confirmada, pois não houve uma correlação significativa entre a Expectativa de Desempenho e a Intenção de Uso. Pode-se concluir que para as pessoas da amostra estudada não existe a percepção de que o uso do UNIFICA possa contribuir com o aumento de desempenho nas atividades relacionadas ao desenvolvimento de *software*.

A segunda hipótese do presente estudo - O constructo Expectativa de Esforço influencia de forma positiva o constructo Intenção de Uso do UNIFICA - foi confirmada, pois foi observada uma correlação positiva moderada entre a Expectativa de Esforço e a Intenção de Uso. Para os respondentes da pesquisa, existem aspectos relacionados à facilidade de uso do UNIFICA como, por exemplo, a facilidade de aplicar as práticas sugeridas no UNIFICA no processo de desenvolvimento de *software*.

A terceira hipótese do presente trabalho - O constructo Influência Social influencia de forma positiva o constructo Intenção de Uso o UNIFICA - foi confirmada por uma correlação positiva fraca (0,385), porém muito próxima a correlação moderada (conforme tabela 2). Para os respondentes, existe a percepção que existe o incentivo de utilização do UNIFICA dentro da organização.

A quarta hipótese do estudo - O constructo Condições Facilitadoras influencia de forma positiva o constructo Uso do UNIFICA - não foi confirmada, pois não foi possível observar uma correlação significativa entre as Condições Facilitadoras e o Uso do UNIFICA.

A quinta hipótese do presente estudo - A Intenção de Uso influencia de forma positiva o Uso do UNIFICA - não foi confirmada, pois não foi identificada uma correlação positiva entre a intenção de uso e o uso do UNIFICA.

Talvez as duas últimas hipóteses não puderam ser verificadas devido ao número reduzido de respondentes e ao uso recente do UNIFICA.

Outras correlações foram encontradas no estudo e que não estavam presentes no estudo proposto por Venkatesh et al. (2003).

Foi identificada a correlação positiva entre Expectativa de Desempenho e outros três constructos (Expectativa de Esforço, Influência Social e Condições Facilitadoras). Para a amostra estudada, à medida que aumenta a percepção de utilidade do UNIFICA, aumenta também a percepção

de facilidade de uso do UNIFICA, a existência de incentivo do uso do UNIFICA e a existência de condições que facilitam o uso do UNIFICA.

Outra correlação observada foi entre Expectativa de Esforço e outras duas dimensões (Influência Social e Condições Facilitadoras). Para a amostra estudada, o aumento ou a diminuição na percepção de facilidade de uso do UNIFICA aumenta ou diminui a percepção de existência de incentivo de uso e condições que facilitam o uso do UNIFICA.

Também foi encontrada a correlação positiva entre Influência Social e Condições facilitadoras. Deste modo, para a amostra estudada, o aumento ou diminuição na percepção de existência de uso do UNIFICA aumenta ou diminui a percepção de condições que facilitam o uso do UNIFICA.

No presente trabalho também foram analisadas as correlações entre os constructos separando os respondentes entre desenvolvedores e gestores.

Assim como a análise geral dos dados, o grupo de desenvolvedores apresentou correlações entre Expectativa de Desempenho com outras três dimensões (Expectativa de Esforço, Influência Social e Condições Facilitadoras). Enquanto na análise geral os valores apresentados foram 0,684 (ED x EF), 0,528 (ED x IS) e 0,563 (ED x CF), para o grupo de desenvolvedores os valores apresentados foram 0,837 (ED x EF), 0,538 (ED x IS) e 0,593 (ED x CF). Dessa forma, todas as correlações desse grupo apresentaram uma correlação maior que a correlação encontrada para toda a amostra. Outro fator interessante foi a existência de uma correlação forte entre Expectativa de Desempenho e Expectativa de Esforço.

Outras correlações encontradas para os desenvolvedores foram Expectativa de Esforço com outros três constructos (Influência Social, Condições Facilitadoras e Intenção de Uso). Todas apresentaram correlações menores que os dados para toda amostra, sendo que a correlação entre Expectativa de Esforço e Intensão de uso confirmou a correlação apresentada por Venkatesh et al. (2003).

Para o grupo de gestores foi identificada uma correlação positiva forte entre a Expectativa de Esforço e a Intenção de Uso, corroborando com os resultados obtidos em Venkatesh et al. (2003).

Outra correlação forte encontrada para o grupo de gestores foi entre Influência Social e Condições Facilitadoras.

Ao confrontarem-se os dois grupos (gestores e desenvolvedores), é possível observar a diferença entre a correlação de Expectativa de Esforço e Intenção de Uso. Desta forma, para a amostra estudada, a facilidade de uso é mais importante para os gestores na Intenção de Uso do UNIFICA.

Por fim, foi analisada a consistência interna do instrumento de pesquisa, que foi considerado excelente.

A partir dos resultados apresentados no presente estudo, entende-se que deve haver um trabalho para aumentar aceitação e uso do UNIFICA junto a funcionários com menos de 10 anos de trabalho no SERPRO. Principalmente em relação aos constructos Expectativa de Desempenho e Facilidade de Uso.

Em relação ao constructo Expectativa de Desempenho, é possível que seja necessário pensar em outras formas de comunicação para mostrar a importância do UNIFICA para melhorar o desempenho. Já, em relação ao constructo Expectativa de Esforço, seria necessário investir na capacitação dos funcionários que pertencem ao grupo específico para aumentar o entendimento de uso UNIFICA.

Como produto final deste trabalho, propõe-se a criação de um modelo de avaliação de uso e aceitação do UNIFICA (APÊNDICE B). Esse modelo está baseado no UTAUT com critérios adaptados aos métodos ágeis, e inclui o questionário aplicado no presente estudo e os procedimentos adotados para os cálculos estatísticos.

Apesar do presente trabalho ser voltado para uma organização específica, entende-se que é possível aplicar o modelo desenvolvido em outras organizações que utilizam o método ágil como forma de desenvolvimento de *software*. Para isso, basta adaptar as questões ao processo de desenvolvimento da organização a ser estudada.

Para pesquisas posteriores, recomenda-se a aplicação do presente estudo na mesma organização em um número maior de respondentes e em diferentes períodos, conforme o estudo original proposto Venkatesh et al. (2003). Dessa forma, será possível verificar se existem todas as relações de constructos propostas no trabalho, uma vez que não foi possível observar a correlação do constructo Condições Facilitadoras e Uso do UNIFICA e Intenção de Uso e Uso do UNIFICA devido ao número reduzido de respondentes.

REFERÊNCIAS

AJZEN, I. From Intentions to Actions: A Theory of Planned Behavior. IN: KUHL, J.; BECKMAN, J. (Org.) Action Control: From Cognition to Behavior. Berlin: Springer, p. 11-39, 1985.

AJZEN, I. Residual Effects of Past on Later Behavior: Habitual and Reasoned Action Perspectives. Personality and Social Psychology Review, v. 6, n. 2, p. 107-122, 2002.

AHMAD, M. O.; MARKKULA, J.; OVIO, M. Kanban in *software* development: A systematic literature review. 39th Euromicro Conference Series on Software Engineering and Advanced Applications: IEEE, 2013.

BANDURA, A. Social Foundations of Thought and Action: A Social Cognitive Theory, Prentice Hall, Englewood Cliffs, Nova Jersey, 1986.

BECK, K. Embracing change with extreme programming. Computer, IEEE, v. 32, issue 10, p.70–77, 1999.

BECK, K. Embracing change with extreme programming. New Jersey: Addison-Wesley, 2000.

BECK, K.; BEEDLE, M.; VAN BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. Manifesto for Agile Software Development. 2001. Disponível em: http://www.agilemanifesto.org. Acessado em: 01/03/2019.

BOEHM, B. A Spiral Model of Software Development and Enhancement. Computer, IEEE, v.21, i. 5, p. 61-72, May, 1988.

BOEHM, B. Software Engineering. IEEE Transaction on Computers, December, p. 1226-1241, 1976.

BOEHM, B. TURNER, R.; Using Risk to Balance Agile and Plan-Driven Methods. Computer, IEEE, v. 36, i. 6, June, 2003.

BROOKS, F. P. The Mythical Man-Month: Essays on *Software* Engineering. Addison-Wesley, 1995.

CHAN, F.; THONG, J. Y. Acceptance of agile methodologies: A critical review and conceptual framework. Decision Support System, v 46, issue 4, p. 803 – 814, 2009.

CHOW, T.; CAO, D. A survey study of critical success factors in agile software projects. Journal of Systems and Software, v 81, i. 6, p. 961-971, 2008.

ČIŽMEŠIJA, A.; STAPIĆ, Z. GitHub as backbone in Software Engineering course: Technology acceptance analysis. 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2019.

COMPEAU. D. R.; HIGGINS, C. A. Computer Self-Efficacy: Development of a Measure and Initial Test. MIS Quarterly, v.19, n.2, p. 189-211, 1995.

CORDER, G. W.; FOREMAN, D. I. Nonparametric Statistics: A Step-by-Step Approach. 2ed. New Jersey: John Wiley & Sons, 2014.

CRESWELL, J. W. Qualitative Inquiry and Research Design: Choosing Among Five Approaches. Thousand Oaks, CA: Sage, 2012.

CRESWELL, J. W.; PLANO CLARK, V. L. Designing and conducting mixed methods research. 2nd. Los Angeles: SAGE Publications, 2011.

CROSBY, P. B. Quality is Free: The Art of Making Quality Certain, McGraw-Hill,New York, 1979.

DANCEY, C. P.; REIDY, J. Statistics without maths for psychology. 7 ed. New York: Pearson, 2017.

DAVIS, F. D. Perceived Usefulness, Perceived Ease of Use and User Acceptance of Information Technology. MIS Quarterly, v.13, n. 3, p. 319-340, Sept. 1989.

DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. Extrinsic and Intrinsic Motivation to Use Computers in the Workplace. Journal of Applied Social Psychology, v. 22, n. 14, p. 1111-1132, 1992.

DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, v. 35, n. 8, p. 982-1003, 1989.

DEMARCO, T. Why Does *Software* Cost So Much? Dorset House, 1995.

DIKERT, K.; PAASIVAARA, M.; LASSENIUS, C. Challenges and success factors for large-scale agile transformations: A systematic literature review. Journal of System and Software, 119, p. 87-108, 2016.

DYBÅ, T. Improvisation in small *software* organizations. IEEE *Software*, v.17, i. 5, p. 82-87, 2000.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile *software* development: A systematic review. Information and Software Technology, v.50, i. 9-10, p.833-859, 2008.

ELBANNA, A.; SARKER, S. The Risks of Agile *Software* Development: Learning from Adopters. IEEE Software, v. 33, i. 5, p.72-79, Sept-Oct, 2016.

FISHBEIN, M.; AJZEN, I. *Belief, Attitude, Intention and Behavior*: An Introduction to Theory and Research. Addison-Wesley: Reading, MA, 1975.

FREITAS, H.; OLIVEIRA, M.; SACCOL, A. Z.; MOSCAROLA, J. O método de pesquisa survey. Revista de Administração, v. 35, n. 3, p. 105-112, 2000.

FRIEL, C. M. Notes on Factor Analysis. Criminal Justice Centre, Sam Houston State

University, 2009.

GALIN, D.; AVRAHAMI, M. Are CMM program investments beneficial? Analyzing past studies. IEEE *Software*, v.23, i. 6, p. 81-87, 2006.

GAO, Y. Research on the rule of evolution of *software* development process model. 2nd IEEE International Conference on Information Management and Engineering, 2010.

GIL, A. C. Como elaborar projetos de pesquisa. 4 ed. São Paulo: Atlas 2002.

HAIR JR., F.; ANDERSON, R. E.; TATHAM, R. L.; BLACK, W. C. Análise multivariada de dados. Porto Alegre: Bookman, 2005.

HAIR, JR. J. F.; BABIN, B.; MONEY, A. H.; SAMOUEL, P. Fundamentos de métodos de pesquisa em administração. Porto Alegre: Bookman, 2007.

HIGHSMITH, J.; COCKBURN, A. Agile Software Development: The business of innovation. Computer, IEEE, v. 34, i. 9, p. 120 – 127, 2001.

HODA, R.; SALLEH, N.; GRUNDY, J.; TEE, H. M. Systematic literature reviews in agile software development: A tertiary study. Information and Software Technology n. 85, p. 60–70, 2017.

HONG, W.; THONG, J. Y. L.; CHASALOW, L. C.; DHILON, G. User Acceptance of Agile Information Systems: A Model and Empirical Test. Journal of Management Information Systems. v. 28, n. 1, p. 235-272, 2011.

KLINE P. The handbook of psychological testing. 2 ed. Londres: Routledge; 1999.

LEFFINGWELL, D. Scaled Agile Framework. 2019. Disponível em: http: //www.scaledagileframework.com>. Acesso em: 25.07.2019.

LUI, K.; CHAN, K. A road map or implementing extreme programming. Proceedings of international *software* process workshop. Beijing, p. 474–481, 2005.

MAIA. M. Limites de gênero e presença feminina nos cursos superiores brasileiros do campo da computação. Cadernos Pagu, n. 46. Campinas, 2016.

MOORE, G. C.; BENBASAT, I. Development of an instrument to measure the perceptions of adopting an information technology innovation. Information Systems Research. v. 2, n. 3, p. 192-222, 1991.

MORRISON, P.; SMITH, B. H.; WILLIAMS, L. Surveying Security Practice Adherence in Software Development. HoTSoS: Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp. p. 85-94, 2017.

MUDARIKWA, G.; GRACE, T. Agile system development methodologies usage and acceptance in South African banking firms: an exploratory analysis. SAICSIT '18: Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists. p. 248-257, 2018.

PACKLICK, J.; The agile maturity map: a goal-oriented approach to agile improvement. Proceedings of AGILE 2007 conference, p. 266–271, 2007.

PAPATHEOCHAROUS, E.; ANDREOU, A. S. Empirical evidence and state of practice of *software* agile teams. Journal of *Software*-Evolution and Process. v.26, i. 9, p. 855-866, 2014.

POPPENDIECK, M.; CUSUMANO, M. A. Lean Software Development: A Tutorial. IEEE Software, v. 29, i. 5, p. 26-32, 2012.

POPPENDIECK, M.; POPPENDIECK, T. Lean software development: an agile toolkit. Addison-Wesley Professional, 2003.

POPPENDIECK, M.; POPPENDIECK, T. Implementing Lean Software Development: From Concept to Cash. Addison-Wesley Professional, 2006.

PRESSMAN, R. S. Engenharia de Software: McGraw Hill/Nacional, 6 ed., São Paulo, 2006.

PRESSMAN, R., S.; MAXIN, B., R. *Software* Engineering: A Practitioner´s Approach. McGraw-Hill, 8 ed. New York, 2015.

RAJAGOPALAN, S.; MATHEW, S. K. Choice of Agile Methodologies in Software Development: A Vendor Perspective. Journal of International Technology and Information Management, v. 25, i. 1, 2016.

RIGBY, D. K.; SUTHERLAND, J.; TAKEUCHI, H. Embracing agile How to master the process that's transforming management. Harvard Business Review 94. n. 5, p. 41-57, 2016.

ROGERS, E.M. Diffusion of Innovations. 3 ed. The Free Press, New York, 1983.

ROGERS, E.M. Diffusion of Innovations. 5 ed. The Free Press, New York, 2003.

SCHWABER, K.; M. BEEDLE. Agile Software Development With Scrum. 1 ed. New Jersey: Prentice-Hall, 2001.

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

SHINGO, S. A study of the Toyota production system from an industrial engineering viewpoint. Productivity Press, 1989.

SOFTWARE ENGINEERING MEASUREMENT AND ANALYSIS TEAM. Process Maturity Profile of the *Software* Community 1999 Year End Update. Pittsburgh: Carnegie Mellon University; 2000.

SOMMERVILLE, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

STANDISH GROUP. The Chaos Report, 2015. Disponível em http://www.standishgoup.com/sample_research_files/CHAOSReport2015-Final.pdf Acessado em: 11/05/2019.

SUNNER, D. Adapting to need of the hour: Understanding Agile methodology and Agile techniques. 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT): IEEE, 2016.

TAYLOR, S.; TODD, P. A. Understanding Information Technology Usage: A Test of Competing Models. Information Systems Research, v. 6, n. 2, 1995.

THANGARAJOO, Y.; SMITH, A. Lean Thinking: An Overview. Journal of Industrial Engineering and Management. v.4, i. 2, 2015.

THOMPSON, R.; HIGGINS, C.; HOWELL, J. Personal computing: toward a conceptual model of utilization. MIS Quarterly, Minneapolis, v.15, n.1, p.124-143, Mar. 1991.

TRIANDIS, H.C. Values, Attitudes, and Interpersonal Behavior. Nebraska Symposium on Motivation, 1979: Beliefs, Attitudes, and Values, University of Nebraska Press, Lincoln, NE, p. 195-259, 1980.

VENKATESH, V., DAVIS, F. D. A theoretical extension of the technology acceptance model: four longitudinal field studies. Manage. Sci, New York, v. 46, n. 2, p. 186- 204. 2000.

VENKATESH, V.; MORRIS, M. G.; DAVIS, G. B.; DAVIS, F. D. User Acceptance of Information Technology: Toward a Unified View. MIS Quarterly, v. 27, n. 3, p. 425-478, 2003.

VENKATESH, V.; THONG, J.Y.L.; XU, X. Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. MIS Quarterly. v. 36, n.1, p. 157-178, 2012.

VIJAYASARATHY, L. R.; BUTLER, C. W. Choice of *Software* Development Methodologies: Do Organizational, Project, and Team Characteristics Matter? IEEE *Software* v. 33, i. 5, p. 86-94, 2016.

WAN, H.; CHEN, F. F. A Web-based Kanban system for job dispatching, tracking, and performance monitoring The International Journal of Advanced Manufacturing Technology, 38, p. 995–1005, 2008.

WOMACK, J. P.; JONES, D. T.; ROOS, D. The Machine That Changed the World: The Story of Lean Production, New York: Rawson and Associates; 1990.

WOMACK, J. P.; JONES, D. T. Lean Thinking: Banish Waste and Create Wealth in Your Corporation. London: Simon & Schuster, 1996.

WOMACK, J. P.; JONES, D. T. Lean Thinking: Banish Waste and Create Wealth in Your Corporation. Free Press, 2010.

APÊNDICE A – INSTRUMENTO DE COLETA DE DADOS

Pesquisa "Avaliação de uso e aceitação do UNIFICA no SERPRO".

O objetivo deste estudo é explorar uso e aceitação do UNIFICA no SERPRO, a partir do ponto de vista dos desenvolvedores e gestores.

Sua contribuição muito engrandecerá nosso trabalho ao trazer uma visão específica e única sobre o assunto.

As informações obtidas por meio desta pesquisa serão confidenciais e asseguramos o sigilo sobre sua participação. Os dados serão divulgados de forma a não possibilitar sua identificação, protegendo e assegurando sua privacidade.

Ao final desta pesquisa, o trabalho completo será disponibilizado no site do Programa de Mestrado do Centro Paula Souza.

A pesquisa é rápida (até 8 minutos) e está dividida em três grandes blocos: Caracterização do respondente, Atividade profissional e Uso do modelo de desenvolvimento de software UNIFICA.

Agradecemos desde já sua participação.

*Obrigatório

1. Você concorda em participar da pesquisa? *

Marcar apenas um oval.

- ◯ Sim, concordo em participar da
- ◯ pesquisa. Não concordo em participar da pesquisa

Caracterização do respondente

Nessa seção serão apresentadas questões de múltipla escolha para caracterização do respondente. Selecione a alternativa que descreve melhor suas características.

2. Gênero? *

Marcar apenas um oval.

- ◯ Feminino
- ◯ Masculino

3. Idade? *

Marcar apenas um oval.

- ◯ Até 30 anos
- ◯ 31-40 anos
- ◯ 41-50 anos
- ◯ Acima de 51 anos

4. Grau de instrução? *

Marcar apenas um oval.

- ◯ Ensino fundamental (até 8ª série)
- ◯ Ensino Médio
- ◯ Superior Completo
- ◯ Especialização
- ◯ Mestrado
- ◯ Doutorado

5. Exerce função gerencial?

Marcar apenas um oval.

- Sim
- Não

6. Tempo de Experiência em desenvolvimento de software (dentro e fora do Serpro)? *

Marcar apenas um oval.

- Nenhum
- até de 5 anos
- 6-10 anos
- 11-20 anos
- Acima de 20anos

7. Tempo de trabalho no SERPRO? *

Marcar apenas um oval.

- até de 5 anos
- 6-10 anos
- 11-20 anos
- acima de 20 anos

Atividade profissional

Nessa seção serão apresentadas questões de múltipla escolha sobre a atividade profissional do respondente. Selecione a alternativa que descreve melhor sua atividade profissional.

8. Linguagem de programação que utiliza no SERPRO? *

Marcar apenas um oval.

- ◯ Linguagem de programação de baixa plataforma (Java, Oracle forms, javascript,etc.)
- ◯ Linguagem de programação de alta plataforma (NATURAL, COBOL, JCL, etc.)
- ◯ Ambas
- ◯ Nenhuma

9. Tempo de Experiência em desenvolvimento ágil (dentro e fora do SERPRO)? *

Marcar apenas um oval.

- ◯ até de 5 anos
- ◯ 6-10 anos
- ◯ 11-15 anos
- ◯ acima de 15 anos
- ◯ Não tenho experiência em desenvolvimento ágil

10. Tempo de Experiência em desenvolvimento ágil no SERPRO? *

Marcar apenas um oval.

- ◯ até de 5 anos
- ◯ 6-10 anos
- ◯ 11-15 anos
- ◯ acima de 15 anos
- ◯ Não tenho experiência em desenvolvimento ágil

11. Quantidade de entregas utilizando o Ágil no SERPRO? *

Marcar apenas um oval.

- Nenhuma entrega
- até 5 entregas
- 6-10 entregas
- 10-15 entregas
- 16-20 entregas
- Acima de 20 entregas

12. Tempo de utilização do UNIFICA? *

Marcar apenas um oval.

- Nenhum
- Até 6 meses
- 7 a 12 meses
- Acima de 1 ano

13. Quantidade de entregas utilizando o UNIFICA? *

Marcar apenas um oval.

- Nenhuma entrega
- até 5 entregas
- 6-10 entregas
- 10-15 entregas
- 16-20 entregas
- Acima de 20 entregas

Uso do modelo de desenvolvimento de software UNIFICA

Nessa seção serão apresentadas afirmações relacionadas ao UNIFICA. Selecione a opção que expressa sua opinião.

14. O modelo de desenvolvimento de software UNIFICA é útil para eu usar em projetos de desenvolvimento de software. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

15. O modelo de desenvolvimento de software UNIFICA me ajuda a executar as tarefas relacionadas ao desenvolvimento de forma mais rápida. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

16. Usar o modelo de desenvolvimento de software UNIFICA aumenta minha produtividade no que diz respeito às atividades de desenvolvimento de software. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

17. Foi fácil aprender como aplicar as práticas sugeridas no modelo de desenvolvimento de software UNIFICA nos projetos que atuei. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

18. Foi fácil para eu me tornar hábil no modelo de desenvolvimento de desenvolvimento de software UNIFICA. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

19. Eu considero o modelo de desenvolvimento de software UNIFICA fácil de utilizar. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

20. As pessoas com as quais trabalho no SERPRO me incentivam a fazer uso do modelo de desenvolvimento de software UNIFICA. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

21. O gerente do setor no qual trabalho considera que eu devo utilizar o modelo de desenvolvimento de software UNIFICA nos projetos de software que atuo. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

22. Em geral, minha organização tem apoiado o uso do modelo de desenvolvimento de software UNIFICA. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

23. Eu tenho à minha disposição os recursos necessários para utilizar modelo de desenvolvimento de software UNIFICA. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

24. Eu tenho os conhecimentos necessários para utilizar o modelo de desenvolvimento de software UNIFICA nos projetos que atuo. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

25. O modelo de desenvolvimento de software UNIFICA é compatível com outras tarefas que eu utilizo no desenvolvimento de software. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

26. Posso obter ajuda de outros colegas quando encontro dificuldades no uso do modelo de desenvolvimento de software UNIFICA. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

27. Eu pretendo utilizar o modelo de desenvolvimento de software UNIFICA em projetos futuros. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

28. Eu irei utilizar o modelo de desenvolvimento de software UNIFICA em projetos futuros. *

Marcar apenas um oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Discordo levemente
- ◯ Concordo levemente
- ◯ Concordo
- ◯ Concordo totalmente
- ◯ Não sei

29. Embora possa ser favorável, o uso do modelo de desenvolvimento de software UNIFICA certamente não é obrigatório no meu trabalho. *

Marcar apenas um oval.

- ◯ Discordo totalmente
- ◯ Discordo
- ◯ Discordo levemente
- ◯ Concordo levemente
- ◯ Concordo
- ◯ Concordo totalmente
- ◯ Não sei

30. O uso do modelo de desenvolvimento de software UNIFICA é pouco relevante nos projetos que atuo. *

Marcar apenas um oval.

- Discordo totalmente
- Discordo
- Discordo levemente
- Concordo levemente
- Concordo
- Concordo totalmente
- Não sei

Agradecimento

As repostas foram recebidas com sucesso.

Obrigado por participar da pesquisa!

APÊNDICE B – MODELO DE AVALIAÇÃO DE USO E ACEITAÇÃO DO UNIFICA

Apresentação

Desde a década de 1990, tem surgido estudos relacionados a aceitação de tecnologia com objetivo de identificar fatores intrínsecos e extrínsecos envolvidos nas decisões, intenções e satisfação dos indivíduos quanto ao uso da tecnologia da informação.

O modelo proposto tem como objetivo auxiliar os gestores na avaliação do uso e da aceitação do UNFICA. Está baseado na Teoria Unificada de Aceitação de Uso da Tecnologia (UTAUT) e a avaliação é feita por meio de um questionário que deve ser enviado aos usuários do UNIFICA.

O UTAUT está baseado em quatro constructos (Expectativa de Desempenho, Expectativa de Esforço, Influência Social e Condições Facilitadoras) que têm influência direta sobre a Intenção de Uso e Uso Efetivo da Tecnologia.

Para atender as necessidades do UNIFICA, os constructos foram adaptados e estão detalhados conforme o quadro abaixo:

Quadro 1: Constructos UTAUT adaptados ao UNIFICA

Constructo	**Definição**
Expectativa de Desempenho	Mede o grau em que uma pessoa acredita que o uso do UNIFICA melhore o desempenho do seu trabalho.
Expectativa de Esforço	Mede o grau em que uma pessoa acredita que seja fácil utilizar o UNIFICA.
Influência Social	Mede a percepção de uma pessoa sobre a opinião (de aprovação ou reprovação) das pessoas que ela considera importante sobre o uso do UNIFICA.
Condições Facilitadoras	Mede a percepção das pessoas quanto à existência de condições que dêem suporte ao uso do UNIFICA.

Método

Para avaliação de uso e aceitação do UNIFICA é necessário seguir os seguintes passos:

1. Seleção de equipes a serem avaliadas. Sugere-se a escolha de uma amostra com o maior número possível de participantes para uma avaliação mais consistente.
2. Disponibilização do questionário de pesquisa (Apêndice A) para os respondentes da amostra. O questionário pode ser disponibilizado por intermédio de e-mail, utilizando-se do Google Forms ou outro aplicativo para esse fim.
3. Após a coleta,, deve-se fazer o tratamento dos dados para posterior avaliação conforme a seguir:
    a. Transposição das respostas para o LibreOffice Calc
    b. Codificação das respostas em códigos numéricos
    c. Inversões de valores das afirmações 29 e 30 por se tratarem de afirmações negativas
    d. Eliminação de respondentes com mais de 9 (nove) afirmações com o valor “não sei”.
4. Cálculos estatísticos:
    a. Cálculo de mediana agrupando as afirmações por constructo: As afirmações podem ser agrupadas por constructo (conforme quadros 2, 3, 4, 5 e 6) e para o cálculo das medianas, pode-se utilizar o LibreOffice. Para o cálculo da mediana, deve-se usar a fórmula Quartil(Dados; tipo), onde os dados representam a matriz de dados de uma amostra e o tipo representa o tipo de quartil (0 = MÍNIMO, 1 = 25%, 2 = 50% (MED), 3 = 75% e 4 = MÁXIMO.). Para uso do modelo, o tipo deve ser 2 (mediana).
    b. Cálculo de moda agrupando as afirmações por constructo: No LibreOffice, para o cálculo da moda, deve-se usar a fórmula MODO(Dados), onde os dados representam a matriz de dados de uma amostra.

c. Cálculo de correlação de Spearman para avaliação de correlações encontradas: Para o cálculo das correlações é necessário primeiro calcular a mediana das respostas por constructos por respondente e depois utilizar um software de cálculo estatístico como o SPSS para calcular a correlação entre os constructos. A correlação de Spearman no SPSS é calculada da seguinte forma: Selecionar a opção Analisar, Correlacionar, Bivariável e depois escolher as variáveis e selecionar Spearman. A ferramenta SPSS fará o destaque das correlações significativas.

Após o cálculo das correlações, é possível avaliar a força de correlações encontradas comparando os resultados obtidos com a tabela 1.

Tabela1 Tabela de interpretação dos coeficientes de correlação de Spearman

Rô de Spearman	**Correlação**
1	perfeita
0,7 – 0,9	forte
0,4 – 0,6	moderada
0,1 – 0,3	fraca
0	sem correlação

Também é possível segmentar os dados por categorias e refazer os cálculos de medianas e correlações para efetuar eventuais comparações com a amostra de forma geral.

Quadro 2 Constructos Expectativa de desempenho e itens do questionário

Constructo	Itens do instrumento de pesquisa
Expectativa de Desempenho	14- O modelo de desenvolvimento de software UNIFICA é útil para eu usar em projetos de desenvolvimento de software.
	15- O modelo de desenvolvimento de software UNIFICA me ajuda a executar as tarefas relacionadas ao desenvolvimento de forma mais rápida.
	16- Usar o modelo de desenvolvimento de software UNIFICA aumenta minha produtividade no que diz respeito às atividades de desenvolvimento de software.

Quadro 3 Constructos Expectativa de esforço e itens do questionário

Constructo	Itens do instrumento de pesquisa
Expectativa de Esforço	17- Foi fácil aprender como aplicar as práticas sugeridas no modelo de desenvolvimento de software UNIFICA nos projetos que atuei.
	18- Foi fácil para eu me tornar hábil no modelo de desenvolvimento de desenvolvimento de software UNIFICA.
	19- Eu considero o modelo de desenvolvimento de software UNIFICA fácil de utilizar.

Quadro 4 Constructos Influência social e itens do questionário

Constructo	Itens do instrumento de pesquisa
Influência Social	20- As pessoas com as quais trabalho no SERPRO me incentivam a fazer uso do modelo de desenvolvimento de software UNIFICA.
	21- O gerente do setor no qual trabalho considera que eu devo utilizar o modelo de desenvolvimento de software UNIFICA nos projetos de software que atuo.
	22- Em geral, minha organização tem apoiado o uso do modelo de desenvolvimento de software UNIFICA

Quadro 5 Constructos Condições facilitadoras e itens do questionário

Constructo	Itens do instrumento de pesquisa
Condições Facilitadoras	23- Eu tenho à minha disposição os recursos necessários para utilizar modelo de desenvolvimento de software UNIFICA
	24- Eu tenho os conhecimentos necessários para utilizar o modelo de desenvolvimento de software UNIFICA nos projetos que atuo.
	25-O modelo de desenvolvimento de software UNIFICA é compatível com outras tarefas que eu utilizo no desenvolvimento de software.
	26- Posso obter ajuda de outros colegas quando encontro dificuldades no uso do modelo de desenvolvimento de software UNIFICA.

Quadro 6 Constructos Intenção de uso e itens do questionário

Constructo	Itens do instrumento de pesquisa
Intenção de Uso	27 - Eu pretendo utilizar o modelo de desenvolvimento de software UNIFICA em projetos futuros.
	28 - Eu irei utilizar o modelo de desenvolvimento de software UNIFICA em projetos futuros.
	29 - Embora possa ser favorável, o uso do modelo de desenvolvimento de software UNIFICA certamente não é obrigatório no meu trabalho.
	30 - O uso do modelo de desenvolvimento de software UNIFICA é pouco relevante nos projetos que atuo.