

UNIVERSIDADE DE TAUBATÉ

Gustavo Traversin

DEVOPS:

**habilidades e capacidades necessárias para sua utilização
na área de tecnologia da informação**

Taubaté – SP

2017

Gustavo Traversin

**DEVOPS:
habilidades e capacidades necessárias para sua utilização
na área de tecnologia da informação**

Monografia apresentada para obtenção do Certificado de Especialização no Curso de MBA em Gerencia de Projetos do Departamento de Gestão e Negócios da Universidade de Taubaté.

Orientador: Prof. Dr. Luiz Carlos Fraga

Taubaté – SP

2017

GUSTAVO TRAVERSIN

**DevOps: habilidades e capacidades necessárias para sua utilização na área de
Tecnologia da Informação**

Monografia apresentada para obtenção do
Certificado de Especialização no Curso de
MBA em Gerencia de Projetos do
Departamento de Economia, Contabilidade e
Administração da Universidade de Taubaté.

Data: ____ / ____ / ____

Resultado: _____

BANCA EXAMINADORA

Prof. Dr. Luiz Carlos Fraga

Universidade de Taubaté

Assinatura: _____

Prof. Dr. Edson Aparecida De Araujo Querido Oliveira

Universidade de Taubaté

Assinatura: _____

Prof. Me. Evandro Luiz De Oliveira

Universidade de Taubaté

Assinatura: _____

À minha mãe, ao meu pai, à minha esposa e às
minhas amadas filhas, Maria Eduarda e Maria
Fernanda, por quem tenho muito amor e estima.

AGRADECIMENTOS

Agradeço primeiramente a Deus, que nos deu paciência e sabedoria para que conseguíssemos alcançar o término deste estudo.

Ao estimado Prof. Dr. Luiz Carlos Fraga, pela valiosa orientação, confiança e especialmente pela paciência e tempo dedicado durante todo o período de construção deste trabalho.

À Universidade de Taubaté e a todos os professores, que nos transmitiram seus conhecimentos. Não apenas por isso, mas também por que nos ensinaram lições de vida para que possamos crescer pessoalmente e profissionalmente.

Aos colegas de sala de aula, que sempre ajudaram e hoje fazem parte da história da minha vida.

Só a experiência própria é capaz de tornar sábio o ser humano

Sigmund Freud

RESUMO

Nos últimos anos, os métodos convencionais de gerenciamento de projetos têm sido gradativamente substituídos por métodos ágeis na indústria de tecnologia da informação (TI), mais especificamente em desenvolvimento de softwares. Os conceitos dos métodos ágeis evoluíram para uma nova forma de aproximar as áreas de desenvolvimento e operações, originando o DevOps, a colaboração entre desenvolvimento e operações. O uso do DevOps requer capacidades e habilidades que ainda não são absolutamente conhecidas pelos profissionais de TI. Este trabalho tem como objetivo identificar as capacidades e habilidades necessárias para a prática do DevOps e para isso realizou-se uma survey com 108 profissionais experientes na área de TI na Região Metropolitana do Vale do Paraíba e Litoral Norte (RMVALE). Os dados foram analisados e validados por equações estruturais e os resultados indicam cinco dimensões chave para adoção do DevOps: produtividade, mentalidade ágil, velocidade e automação, qualidade e segurança. Essas dimensões, se forem bem exploradas pelas organizações e seus profissionais, podem garantir a entrega de projetos e serviços de TI de forma ágil e eficaz.

Palavras-chave: *DevOps*. Ágil. Automação. Gerenciamento de Projetos. Tecnologia da Informação. Desenvolvimento. Operações. RMVALE.

ABSTRACT

DevOps: skills and abilities needed for use it in Information Technology area

In recent years, conventional methods of project management have been gradually replaced by agile methods in the information technology (IT) industry, specifically in software development. The concepts of agile methods have evolved into a new way of approaching development and operations, leading to DevOps - collaboration between development and operations. However, to use DevOps is needed skills and abilities that are not yet fully understood by IT professionals. This monography aims to identify the skills and abilities necessary to practice DevOps and for this, a survey was conducted with 108 experienced professionals in the IT area in the metropolitan region of the Paraíba Valley (RMVALE). The survey was analyzed and validated through structural equations and the results indicate five key dimensions for DevOps adoption: Productivity, Agile Mentality, Speed and Automation, Quality, Safety. These dimensions, if well explored by organizations and their professionals, can ensure the delivery of IT projects and services in an agile and efficient way.

Keywords: *DevOps*. Agile. Automation. Project Management. Information Technology. Development. Operations.

LISTA DE FIGURAS

Figura 1 – Desenvolvimento de uma solução de sistema de informação	22
Figura 2 – Esquema convencional de entregas em TI	23
Figura 3 – Modelo <i>waterfall</i>	25
Figura 4 – Modelo <i>waterfall</i> eminente a falhas	26
Figura 5 – Metodologia scrum.....	31
Figura 6 – Componentes de serviço	36
Figura 7 – Ciclo de Vida ITIL.....	40
Figura 8 – Sistema de Gestão de Serviços, na visão da ISO/IEC 20000	44
Figura 9 – Demonstração <i>DevOps</i>	47
Figura 10 – G*Power <i>Software</i> – Cálculo da Amostra Mínima.....	60
Figura 11 – Modelo Estrutural da Pesquisa <i>DevOps</i>	65
Figura 12 – Modelo de mensuração da pesquisa.....	67
Figura 13 – Carga fatorial dos indicadores do modelo inicial	72
Figura 14 – Carga fatorial dos indicadores do modelo ajustado	74
Figura 15– Resultados do procedimento de <i>bootstrapping</i> (Teste t)	80
Figura 16 – Dimensões chaves para adoção ao DevOps	95

LISTA DE QUADROS

Quadro 1 – Características relevantes do modelo tradicional	27
Quadro 2 – Princípios dos métodos ágeis	29
Quadro 3 – Principais características dos modelos <i>waterfall</i> e <i>scrum</i>	34
Quadro 4 – Definições e características dos tipos de serviços de TI.....	37
Quadro 5 – Processos e funções da ITIL	42
Quadro 6 – Estrutura ISO/IEC 20000	43
Quadro 7 – Estrutura da parte 1 da norma ISO/IEC 20000.....	45
Quadro 8 – Esquema de diagrama de testes e seus tipos.....	55
Quadro 9 – Infraestrutura como código: padrões	58
Quadro 10 – Parâmetros e indicadores do modelo	63
Quadro 11 – Parâmetros de referência r^2	70
Quadro 12 – Indicadores de velocidade/automação e suas cargas fatoriais.....	83
Quadro 13 – Indicadores de qualidade e suas cargas fatoriais	85
Quadro 14 – Indicadores de segurança e suas cargas fatoriais	86
Quadro 15 – Indicadores de produtividade e suas cargas fatoriais	89
Quadro 16 – Indicadores de mentalidade ágil e suas cargas fatoriais.....	91
Quadro 17 – Síntese dos resultados da pesquisa.....	93

LISTA DE TABELAS

Tabela 1 – Projetos e seus resultados.....	18
Tabela 2 – Resultados dos projetos por metodologia (ágil x <i>waterfall</i>).....	19
Tabela 3 – Carga fatorial dos coeficientes no modelo inicial.....	73
Tabela 4 – Carga dos coeficientes no modelo ajustado.....	75
Tabela 5 – Confiabilidade composta e Alfa de Cronbach do modelo ajustado.....	76
Tabela 6 – Validade convergente do modelo ajustado.....	76
Tabela 7 – Cargas cruzadas dos indicadores do modelo ajustado.....	77
Tabela 8 – Coeficientes de determinação de Pearson (R^2).....	78
Tabela 9 – Tamanho do efeito (f^2).....	78
Tabela 10 – Indicador de validade preditiva (Q2).....	79
Tabela 11 – Índice de ajuste do modelo (GoF).....	79

LISTA DE SIGLAS

AC	Alfa de Cronbach
AUT	Autorização de Utilização Terapêutica
AVE	Average Variance Extracted
CC	Confiabilidade composta
CCTA	Central Computer Telecommunication Agency
CSV	Comma Separated Values (Valores Separados por Vírgula).
DevOps	Development (Desenvolvimento) Operations (Operações)
f^2	Tamanho do Efeito
GITIM	Government Information Technology Infrastructure Method, renomeado
GoF	Goodness of Fit
HPE	Hewlett Packard Enterprise
IEC	International Electrotechnical Commission
ISO	International Organization of Standardization
ITIL®	The Information Technology Infrastructure Library
ITIMF	IT Information Management Forum
ITSMF	IT Service Management Forum
MA	Mentalidade Ágil
MEEPLS	Modelagem de Equações Estruturais Baseada nos Mínimos Quadrados Parciais
PLS	Partial Least Squares-
PLS	Squares-path Modeling (Modelo de Caminhos)
PMI	Project Management Institute
PROD	Produtividade
Q^2	Stoner-Geisser
QA	Quality Assurance - garantia da qualidade
R^2	Coefficientes de determinação de Pearson
RMVALE	Região Metropolitana do Vale do Paraíba e Litoral Norte
SEG	Segurança
SGS	Sistema de Gestão de Serviços
t de Student	Valores de todos coeficientes de caminho do modelo
TI/IT	Tecnologia da Informação/Information Technology
VC	Validade convergente
VL	Variável latente
VO	Indicador ou variável observada
Γ	Coefficientes de caminho

SUMÁRIO

RESUMO.....	7
ABSTRACT	8
LISTA DE FIGURAS	9
LISTA DE QUADROS	10
LISTA DE TABELAS	11
LISTA DE SIGLAS	12
1 INTRODUÇÃO	16
1.1 Problema	17
1.2 Objetivos do Trabalho	17
1.2.1 Objetivo Geral	17
1.2.2 Objetivos Específicos	17
1.3 Delimitação do Trabalho	18
1.4 Relevância do Trabalho	18
1.5 Organização do Trabalho.....	20
2 REVISÃO DA LITERATURA.....	21
2.1 Tecnologia de Informação.....	21
2.2 Abordagens convencionais de entregas de TI	21
2.3 Projetos	24
2.4 Gerenciamento tradicional – Modelo <i>Waterfall</i>	25
2.5 Gerenciamento ágil – Modelo SCRUM.....	27
2.5.1 Papéis.....	31
2.5.2 Eventos	32
2.5.3 Artefatos	33
2.6 Principais características dos modelos <i>Waterfall</i> e <i>Scrum</i>	34
2.7 Gerenciamento de serviços e operações em TI	36
2.7.1 Modelo ITIL	38
2.7.2 Modelo ISO 20000	42
2.8 <i>DevOps</i> – História	46
2.8.1 Conceitos	47
2.8.1.1 Velocidade e automação.....	48
2.8.1.2 Qualidade	49
2.8.1.3 Produtividade	50

2.8.1.4 Segurança	51
2.8.1.5 Mentalidade ágil	52
2.8.2 Boas práticas	52
2.8.2.1 Integração contínua	52
2.8.2.2 Entrega contínua	53
2.8.2.3 Testes contínuos e automatizados	54
2.8.2.4 Infraestrutura como código	57
3 MÉTODO DE PESQUISA	59
3.1 Natureza da pesquisa	59
3.2 População e amostra	59
3.3 Instrumento e procedimentos de aquisição dos dados	61
3.4 Modelo estrutural.....	64
3.5 Análise e interpretação dos dados.....	65
3.5.1 Avaliação do modelo de mensuração reflexivo	68
3.5.1.1 Confiabilidade dos indicadores	68
3.5.1.2 Consistência interna (confiabilidade composta)	68
3.5.1.3 Validade convergente	68
3.5.1.4 Validade discriminante	69
3.5.2 Avaliação do modelo estrutural.....	69
3.5.2.1 Coeficiente de Pearson (R^2).....	69
3.5.2.2 Tamanho do efeito (f^2).....	70
3.5.2.3 Relevância preditiva (Q^2) ou indicador de Stoner-Geisser	70
3.5.2.4 Goodness of Fit (GoF)	70
3.5.2.5 Análise dos valores e significância dos Coeficientes de Caminho (β).....	71
4 DISCUSSÃO DOS RESULTADOS.....	72
4.1 Apresentação e interpretação dos resultados	72
4.1.1 Avaliação do modelo de mensuração.....	72
4.1.1.1 Avaliação do modelo de mensuração reflexivo	73
4.1.1.2 Avaliação do modelo estrutural.....	77
4.2 Análise e discussão dos resultados.....	80
4.2.1 Comportamento dos parâmetros e seus indicadores	81
4.2.1.1 Dimensão DevOps	81
4.2.1.1.1 Construto Velocidade e Automação (AUT).....	81
4.2.1.1.2 Construto Qualidade (QA).....	84

4.2.1.1.3 Construto Segurança (SEG)	86
4.2.1.1.4 Construto Produtividade (PROD).....	88
4.2.1.1.5 Construto Mentalidade Ágil (MA)	90
4.2.2 Dimensões chave do modelo DevOps.....	92
5 CONSIDERAÇÕES FINAIS	96
REFERÊNCIAS	99
APÊNDICE A - Questionário da pesquisa	105
ANEXO 1- Aprovação do comitê de ética em pesquisa	112

1 INTRODUÇÃO

Com o surgimento do fenômeno da globalização e o aumento do dinamismo do mercado, as organizações foram forçadas a renovar seus processos para enfrentar uma nova realidade, que exige um grande desenvolvimento de estratégias e procedimentos para conquistar uma porção do mercado mundial cada vez mais competitivo, exigindo mudanças mais rápidas nos processos de negócios das organizações que, por sua vez, são e estão cada vez mais dependentes da tecnologia da informação (TI).

As mudanças no contexto do negócio, independentemente de sua natureza, seja ela mercadológica, regulatória ou estratégica, têm implicações nos sistemas de informação, seja demanda por novas aplicações ou por melhorias em aplicações existentes, surgindo então uma grande e crescente diversidade de equipamentos, dispositivos e aplicativos tecnológicos.

Por outro lado, os sistemas estão cada vez mais complexos e integrados a outros serviços, o que exige um alto grau de confiabilidade e disponibilidade (BADRINARAYANAN; *et al*, 2013).

Tradicionalmente, os projetos que envolvem o desenvolvimento de *software* adotam abordagens processuais com fases bem definidas e delimitadas, com evolução linear, ênfase no planejamento e no controle.

Os modelos de desenvolvimento de software tradicionais, também chamados de 'orientadas a documentação', são geralmente baseados em entrega de funcionalidades que são especificadas, desenvolvidas e testadas em estágios sequenciais e lineares (ROYCE, 1970; HUTTERMANN, 2012).

Percorre-se um processo longo entre o desenvolvimento de um sistema e sua efetiva entrada em operação. Embora tenham sido amplamente usadas, as abordagens *Waterfall*, ou cascata têm sido questionadas pelo tempo requerido para entregar sistemas funcionais às organizações.

Diante da demanda por entregas mais rápidas e constantes das tecnologias habilitadoras para o negócio, novos métodos têm sido adotados para simplificar os projetos de sistemas e reduzir o ciclo que vai desde a coleta dos requisitos a serem implantados à sua operacionalização. São os chamados métodos ágeis.

Esses métodos se baseiam na identificação das funcionalidades e necessidades mais importantes para o cliente e no desenvolvimento e entrega de partes menores de um sistema de informação em ciclos mais curtos, o que agiliza a entrega de valor.

Com eles estabelece-se um novo paradigma, baseado na aproximação das atividades de desenvolvimento, testes e liberação de *software* para produção e sustentação, denominado *DevOps*, cuja adoção prática requer habilidade e capacidade por parte da organização de TI.

A interação frequente e melhor entre as equipes de projetos de desenvolvimento de *software* e operações, que os suportam após sua implantação, é necessária para adoção do *DevOps* como alternativa aos modelos tradicionais e para modificar os processos de TI dentro das organizações, de forma a criar um ambiente propício e eficaz para suportar as recorrentes necessidades de mudanças.

1.1 Problema

Este trabalho busca responder à seguinte questão de pesquisa: quais as habilidades e capacidades necessárias para a adoção e prática do *DevOps* no desenvolvimento e na entrega de projetos na área de tecnologia da informação?

1.2 Objetivos do Trabalho

1.2.1 Objetivo Geral

Analisar os modelos *Waterfall* e SCRUM de gerenciamento de projetos e ITIL e ISO 20000 para gerenciamento de operações, com enfoque em *DevOps* para identificar as habilidades e capacidades necessárias à sua utilização. O *DevOps* é uma prática de engenharia de *software* que possui o intuito de unificar o desenvolvimento (Dev) e a operação (Ops) de *softwares*.

1.2.2 Objetivos Específicos

O trabalho tem os seguintes objetivos específicos:

- Identificar as habilidades e capacidades em DevOps necessárias para sua utilização na área de Tecnologia da Informação (TI).
- Na perspectiva de profissionais de TI, avaliar a importância das habilidades e capacidades identificadas para a prática do DevOps.

1.3 Delimitação do Trabalho

O presente trabalho limita-se à análise e identificação dos principais conceitos de gerenciamento de projetos com relação aos métodos tradicional e ágil, já que a crescente demanda para novas formas de gestão de projetos se fazem necessárias.

1.4 Relevância do Trabalho

Com base em um estudo com cerca de 50 mil projetos em todo o mundo, que vão de pequenas melhorias até implementações maciças de reengenharia de sistemas, o relatório chamado CHAOS, do grupo *Standish*, demonstra que ainda há muito trabalho a ser feito em torno da obtenção de resultados bem-sucedidos de projetos de desenvolvimento de *software*.

A Tabela 1, traz os resultados obtidos nos projetos executados e avaliados pela pesquisa entre os anos de 2011 a 2015, usando a definição de fatores de sucesso no prazo, no orçamento e nos resultados satisfatórios (HASTIE; WOJEWODA, 2015).

Tabela 1 – Projetos e seus resultados

	2012	2013	2014	2015
Sucesso	27%	31%	28%	29%
Desafiador	56%	50%	55%	52%
Falha	17%	19%	17%	19%

Fonte: adaptada de Hastie e Wojewoda (2015)

De acordo com o relatório CHAOS, do grupo Standish, os projetos foram classificados em três tipos:

- projetos considerados sucessos: foram concluídos no prazo e no orçamento acordado, com todos os recursos e funções inicialmente especificados;
- projetos desafiadores: foram os concluídos e entregues de forma operacional, mas ultrapassaram no prazo e no orçamento inicialmente acordado. Foram entregues com menos recursos e funções do que originalmente especificado; e
- projetos considerados como falha: foram cancelados em algum ponto durante o ciclo de seu desenvolvimento (HASTIE; WOJEWODA, 2015).

Nos últimos anos, com uma grande aceitação de métodos ágeis em projetos de desenvolvimento de *software*, foi possível comparar os resultados entre projetos que utilizaram a metodologia ágil e os que utilizaram a metodologia tradicional *Waterfall*.

Em todos os tamanhos de projetos avaliados, as abordagens ágeis resultaram em projetos mais bem-sucedidos e com menor índice de falhas, conforme mostra a Tabela 2.

Tabela 2 – Resultados dos projetos por metodologia (ágil x *waterfall*)

Tamanho	Método	Bem Sucedido	Desafiador	Falhou
Todos os tamanhos de projetos	Agil	39%	52%	9%
	Waterfall	11%	60%	29%
Projetos de grande porte	Agil	18%	58%	23%
	Waterfall	3%	55%	42%
Projetos de médio porte	Agil	27%	62%	11%
	Waterfall	7%	68%	25%
Projetos de pequeno porte	Agil	58%	38%	4%
	Waterfall	44%	45%	11%

Fonte: adaptado de Hastie e Wojewoda (2015)

Os resultados do estudo mostram como a mudança nos métodos de gestão de projetos se intensificou nos últimos anos em decorrência dos resultados melhores do que os métodos ágeis proporcionam para as empresas que os adotam. Por isso, os profissionais e acadêmicos de tecnologia da informação têm discutido, experimentado e aderido ao modelo *DevOps*.

Os estudos e pesquisas na área desse modelo ainda estão em fase inicial e tendem a focar na questão cultural dos problemas existentes no modelo tradicional de gerenciamento de projetos de *software*, além do aspecto da redução do tempo necessária para entregar valor.

1.5 Organização do Trabalho

O estudo está organizado em cinco seções que se encontram descritas da seguinte forma: na primeira seção encontram-se a introdução, o problema, o objetivo geral, a delimitação, a relevância e a organização do estudo. A segunda seção aborda a revisão da literatura, com conceitos teóricos relacionados ao tema da pesquisa. A terceira seção descreve a metodologia de avaliação do modelo de pesquisa a ser aplicado para a identificação das habilidades e capacidades dos profissionais de TI em *DevOps*. Na quarta seção se encontram a apresentação dos resultados e a validação do modelo de pesquisa. Na quinta seção estão as considerações finais.

2 REVISÃO DA LITERATURA

2.1 Tecnologia de Informação

A ideia que a área de tecnologia da informação (TI) gera somente despesa para as organizações já está ultrapassada. Atualmente, observando a forma como a tecnologia está nas 'coisas' é possível afirmar que é uma época em que os *softwares* estão na maioria das atividades do dia a dia das pessoas.

A rapidez das mudanças tecnológicas faz com que área de TI fique cada vez mais competitiva e a gestão de projetos é uma forma de realizar essa transformação veloz.

Dessa forma, observa-se uma crescente demanda por gerenciamento de projetos de TI. Nos projetos relacionados à tecnologia da informação, seja no âmbito do desenvolvimento de *software* ou na operação de sustentação da tecnologia já existente e que se renova rapidamente é fundamental um conjunto de procedimentos, regras, políticas, metodologias e boas práticas, que ditam como essas demandas devem ser gerenciadas para que possam conseguir atingir seus objetivos com verdadeira qualidade.

De acordo com Vargas (2016):

Para atender a demanda de maneira eficaz, em um ambiente caracterizado pela velocidade das mudanças, torna-se indispensável um modelo de gerenciamento baseado no foco em prioridades e objetivos. Por essa razão, o gerenciamento de projetos tem crescido de maneira tão acentuada no mundo nos últimos anos (VARGAS, 2016, p. 3).

Nesta seção encontra-se todo embasamento teórico utilizado para fundamentar e compor as teorias usadas no estudo. É necessário o entendimento das metodologias que são utilizadas para gerenciar as áreas de projetos e operações em TI.

2.2 Abordagens convencionais de entregas de TI

A tecnologia da informação (TI) nas empresas vem se tornando uma grande aliada.

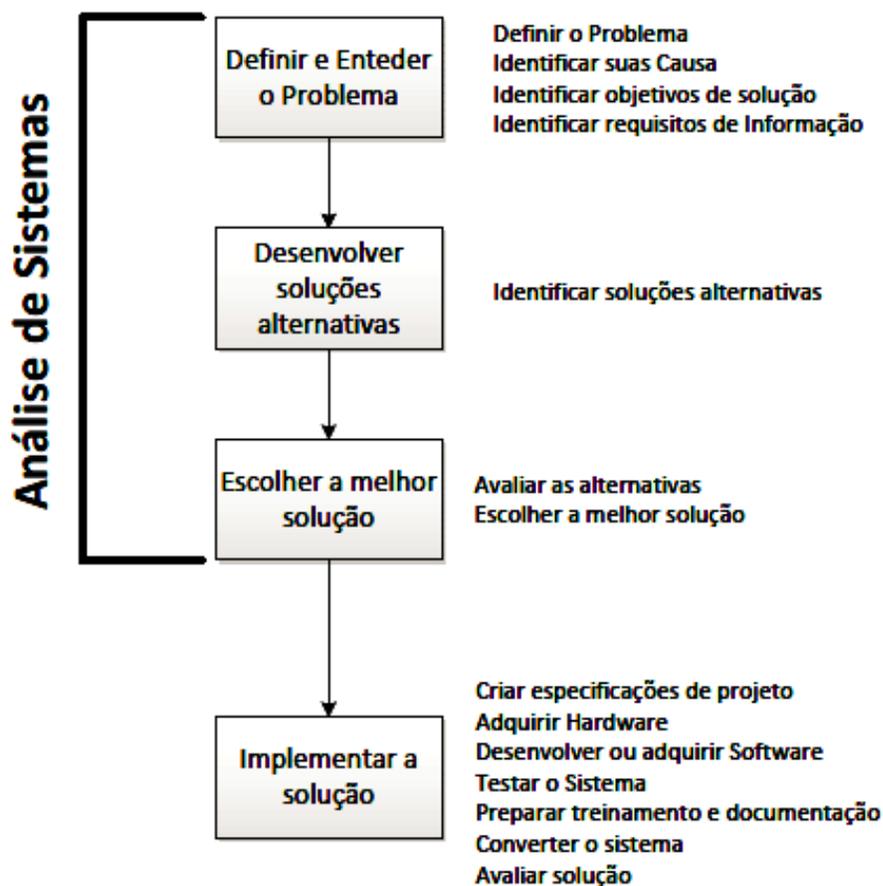
O principal papel dos sistemas de informação consiste em resolver problemas onde geralmente existe a necessidade de desenvolvimento de novas funcionalidades para a solução de um ou mais problemas que a organização reconhece enfrentar, visando obter um desempenho mais eficiente.

Segundo Laudon e Laudon (2014):

Um problema só pode ser solucionado se, antes, for adequadamente definido. Os membros das organizações precisam estar de acordo sobre a existência de algo errado e sobre isso ser grave. O problema precisa ser investigado para ser mais bem compreendido. Em seguida, vem um período em que se delineiam soluções alternativas, e depois outro em que se avalia cada alternativa e se seleciona a melhor. O estágio final é a implementação da solução na qual um projeto detalhado para a solução é especificado, traduzido em um sistema físico testado, introduzido na organização e mais tarde, refinado conforme é usado (LAUDON; LAUDON, 2014, p. 404).

A Figura 1 representa os principais passos necessários e utilizados para que o desenvolvimento de sistemas auxilie e seja eficaz na resolução de problemas.

Figura 1 – Desenvolvimento de uma solução de sistema de informação



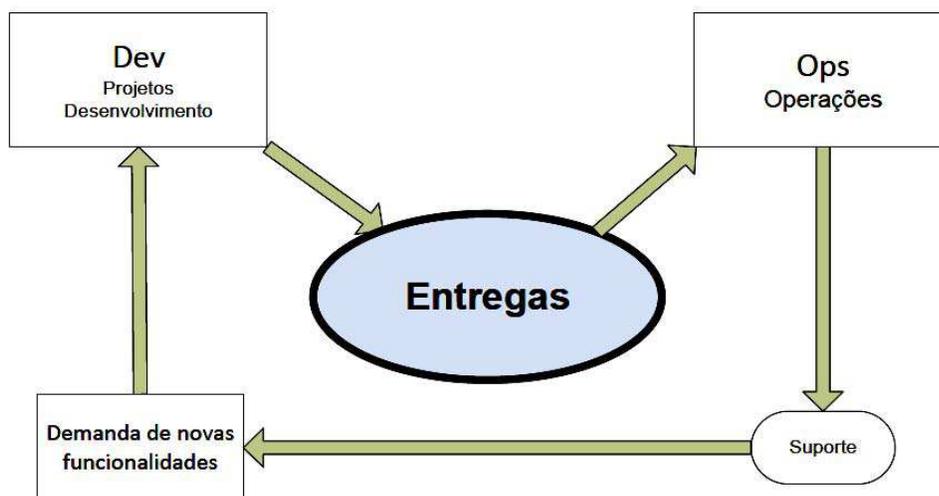
Fonte: adaptada de Laudon e Laudon (2014, p. 404)

A análise de sistemas compreende os três primeiros passos da resolução de problemas: o problema é identificado; logo em seguida informações são coletadas; soluções alternativas são propostas; e depois uma decisão é tomada. O que vem após é denominado projeto, quando a solução propriamente dita é desenvolvida, testada e implementada.

A entrega de projetos e operações na área de TI é realizada em um modelo convencional, que atualmente é o mais utilizado por pequenas, médias e até grandes empresas para gerir a área de TI.

A Figura 2 representa claramente o ciclo de entrega dos projetos para as respectivas operações.

Figura 2 – Esquema convencional de entregas em TI



Fonte: adaptada de Laudon e Laudon (2014, p. 404)

A demanda de um novo projeto é identificada e a empresa realiza um estudo da viabilidade de um novo sistema ou novas funcionalidades para sistemas existentes. Uma vez que a viabilidade é confirmada, a organização inicia o processo de desenvolvimento.

Depois da entrega do sistema e/ou funcionalidades existe uma curva de aprendizado para que entre em operação de forma eficaz, o que não garante ser realmente necessário o que foi projetado.

Após a curva de aprendizagem, uma determinada equipe assume a manutenção de todo o conjunto de servidores, banco de dados, licença de *softwares*, manutenção em rede, suporte ao usuário, o que se chama de operações.

2.3 Projetos

Para se entender melhor o gerenciamento de projetos, é preciso observar o que realmente é um projeto.

Para Vargas (2016):

Projeto é um empreendimento não repetitivo, caracterizado por uma sequência clara e lógica de eventos, com início, meio e fim, que se destina a atingir um objetivo claro e definido, sendo conduzido por pessoas dentro de parâmetros predefinidos de tempo, custo, recursos envolvidos e qualidade (VARGAS, 2016, p. 7).

Projetos existem desde o princípio da civilização e como exemplos citam-se alguns tipos, como a criação de um novo modelo de um carro; escrever um livro; construção ou reforma de uma casa; criação de um novo software para atender uma determinada demanda da organização; preparação do Brasil para a Copa do Mundo em 2014; e preparação do Rio de Janeiro para as Olimpíadas em 2016.

De acordo com o PMI (2013), o projeto é definido como:

(...) um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo. A natureza temporária dos projetos indica que eles têm um início e um término definidos. O término é alcançado quando os objetivos do projeto são atingidos ou quando o projeto é encerrado porque os seus objetivos não serão ou não podem ser alcançados, ou quando a necessidade de o projeto deixar de existir (PMI, 2013, p. 1).

Diante dessas informações, pode-se observar que no desenvolvimento de um projeto são necessárias ferramentas para auxiliar sua gestão e se possa garantir o resultado esperado.

O gerenciamento de projetos é um conjunto de ferramentas que permite à empresa desenvolver um conjunto de habilidades, incluindo conhecimento e capacidades individuais destinadas ao controle de eventos não repetitivos, únicos, e muitas vezes complexos, dentro de um cenário de tempo, custo e qualidade predeterminado (VARGAS, 2016).

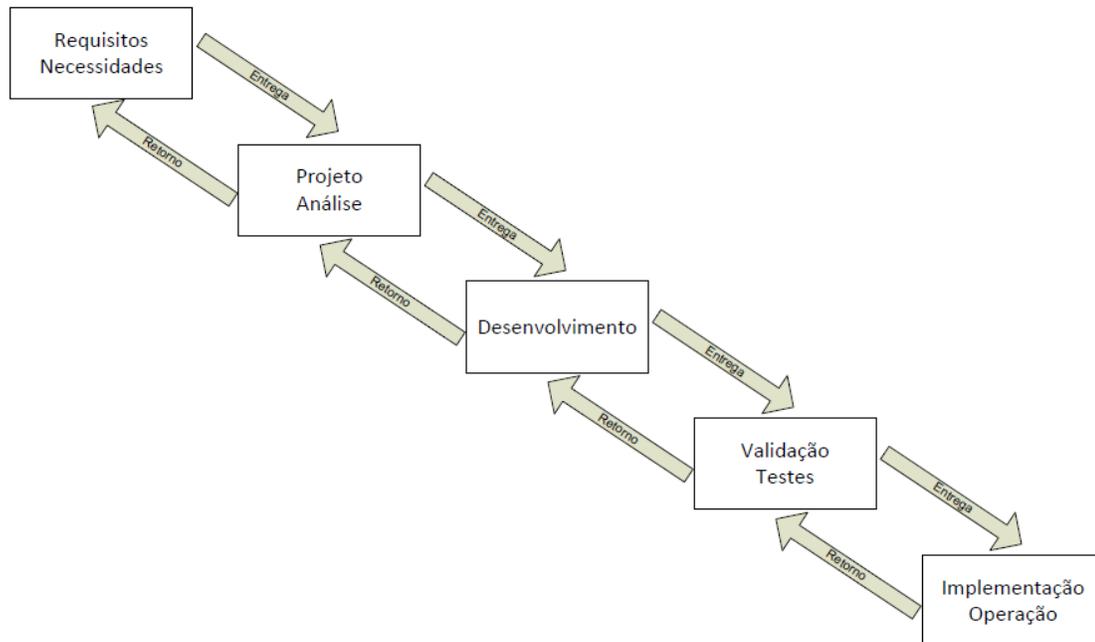
Para gerir projetos de TI existem duas metodologias que são mais utilizadas pelas organizações e serão descritas a seguir: o gerenciamento tradicional, modelo *Waterfall* e o gerenciamento ágil, modelo *SCRUM*.

2.4 Gerenciamento tradicional – Modelo *Waterfall*

Conhecido também como modelo cascata, referindo-se a uma cachoeira em que a água escorre do topo até embaixo, ele surgiu no início da década de 70 quando o seu criador Royce (1970), que trabalhava como engenheiro de *software* para programas aeroespaciais, publicou-o em um artigo.

Ele se baseia na execução de atividades padronizadas em uma determinada sequência, em que o resultado unitário de uma sequência é alimento para a atividade seguinte, podendo retornar para a atividade anterior. Essa interação consolida a qualidade de cada fase do projeto, conforme exemplo na Figura 3.

Figura 3 – Modelo *waterfall*



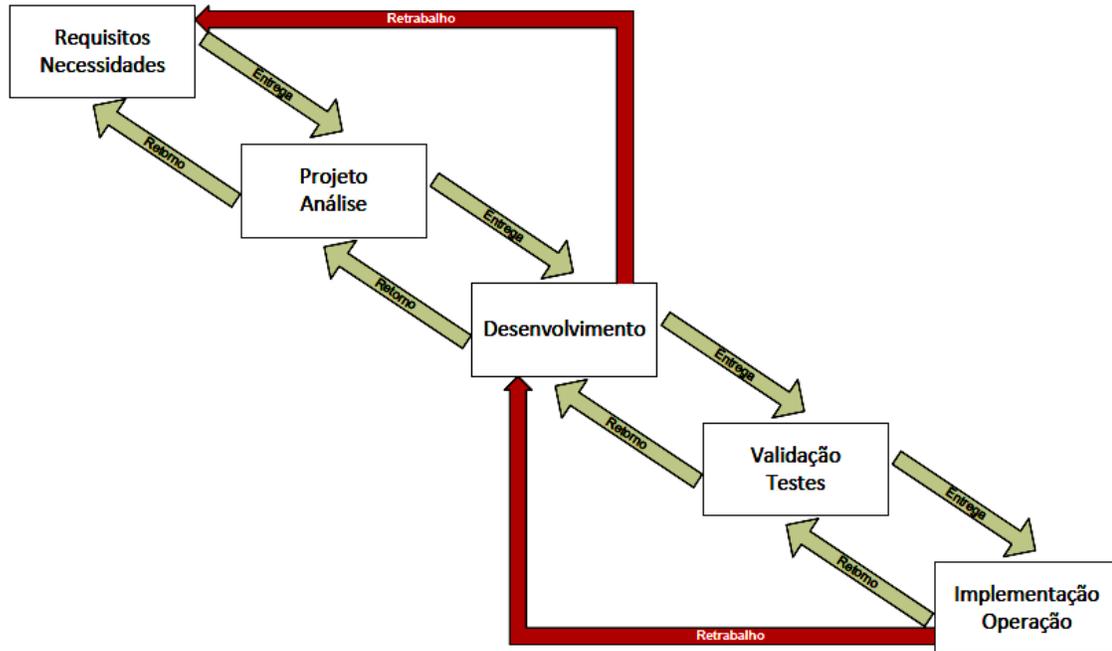
Fonte: adaptada de Royce (1970)

A maioria das equipes de projetos aderiu ao modelo, que é utilizado ainda na atualidade, porém, tolera pouca ou nenhuma mudança entre fases que não são subsequentes.

Assim, um novo requisito durante a fase de testes gera retrabalho nas diversas fases anteriores caso a adaptação não seja compatível com toda a estrutura do projeto.

O próprio autor destaca esse risco quando cita: “Eu acredito neste conceito, mas a implementação descrita acima [*Waterfall*] possui riscos e convida às falhas” (ROYCE; 1970, p. 330, tradução nossa) (Figura 4).

Figura 4 – Modelo *waterfall* eminente a falhas



Fonte: adaptada de Royce (1970)

O modelo tradicional de gerenciamento de projetos exige, portanto, uma atenção especial às atividades de planejamento, documentando de forma massiva os requerimentos, sem muita margem para mudanças fora do momento adequado, engessando o escopo e a condução do projeto como um todo.

Para minimizar esses riscos que decorrem do modelo, Royce (1970, p. 331-338) propõe os seguintes passos:

- criar uma fase de desenho preliminar do projeto antes da fase de análise;
- gerar documentação abrangente do desenho;
- fazer uma simulação completa do processo, a partir do desenho preliminar, flexibilizando os requisitos de documentação para chegar a uma revisão/versão 1, que possa ser utilizada para o processo completo;
- planejar, controlar e monitorar o teste; e
- envolver o cliente.

O modelo *Waterfall* é definido por alguns autores como uma metodologia que é efetiva quando aplicada em projetos cujos requisitos são bem definidos e conhecidos, assumindo que devem ser “totalmente especificáveis, previsíveis e podem ser feitos por meio de um meticuloso e extensivo planejamento” (NERUR; MAHAPATRA; MANGALARAJ, 2005, p. 3).

Esses requisitos relatados pelos autores citados são visivelmente conflitantes com a maioria dos projetos, principalmente quando se trata de desenvolvimento de *software*.

Deemer; *et al* (2010) destacam que:

(...) esta técnica [*Waterfall*] requer que todas as boas ideias apareçam ao começo do ciclo de desenvolvimento, estágio no qual podem ser incorporadas ao plano. Porém, como todos nós sabemos, as boas ideias aparecem espontaneamente ao longo do processo – no começo, na metade e mesmo um dia antes da implantação (...). Com a técnica *Waterfall*, uma boa ideia que apareça tarde no ciclo de desenvolvimento não é um presente, é uma ameaça (DEEMER; *et al*, 2010).

O Quadro 1 demonstra um resumo das características mais relevantes do modelo tradicional de gerenciamento.

Quadro 1 – Características relevantes do modelo tradicional

Tópico	Características
Objetivo principal	Orientado por atividade e centrado em processo
Tipo de Projeto	Estáveis e com baixo nível de mudanças.
Tamanho	Aplicável em projetos de todos os tamanhos. Mais efetivo em projetos de maior duração.
Gerente de Projeto	Controle total do projeto
Equipe do Projeto	Atuação com papéis claros e bem definidos.
Cliente	Participa das fases iniciais de requisitos e das validações dos produtos.
Planejamento	Detalhado e os envolvidos têm o papel de validação, não participam da elaboração do planejamento.
Arquitetura	Definida com foco em todo o projeto e na reusabilidade.
Modelo de Desenvolvimento	Cascata, espiral e iterativo
Comunicação	Formal
Controle de Mudanças	Processo formal de identificação e aprovação entre os envolvidos. Incorporação de novos requisitos pode ser lento e caro.

Fonte: Arakaki e Ribeiro (2006)

2.5 Gerenciamento ágil – Modelo SCRUM

Na década de 1980 existia uma generalização de que a melhor maneira para se concluir um projeto, principalmente de desenvolvimento de *software*, era um processo de planejamento minucioso, com um controle rigoroso e controlado.

Essa visão veio dos engenheiros de *software*, responsáveis pela implantação de projetos grandes e duradouros, como sistemas aeroespaciais e de defesa.

Comumente as equipes trabalhavam em diferentes posições geograficamente e trabalhavam por longos períodos, porém, quando esse modelo é aplicado em organizações de pequeno e médio porte, segundo Sommerville (2011):

(...) se gasta mais tempo em análises de como o sistema deve ser desenvolvido do que no desenvolvimento de programas e testes. Como os requisitos do sistema se alteram, o retrabalho é essencial, e, pelo menos em princípio, a especificação e o projeto devem mudar com o programa (SOMMERVILLE (2011, p. 39-40).

No início dos anos 90, motivados pela insatisfação com as abordagens massivas de metodologia de projetos, desenvolvedores de *software* propuseram novos métodos ágeis de gestão.

De acordo com Sommerville (2011, p. 39-40),

Métodos ágeis, universalmente, baseiam-se em uma abordagem incremental para a especificação, o desenvolvimento e a entrega do *software*. Eles são mais adequados ao desenvolvimento de aplicativos nos quais os requisitos de sistema mudam rapidamente durante o processo de desenvolvimento (SOMMERVILLE, 2011, p. 39-40).

Esse modelo é influenciado pelas premissas de uma prática iniciada na década de 1980, que surgiu em uma linha de produção de automóveis da Toyota, no Japão, conhecida como *Lean* – produção enxuta.

Surgiu então, em fevereiro de 2001, o manifesto ágil, quando alguns metodologistas de projetos e pensadores da área se encontraram em uma reunião para discutir os principais problemas relacionados ao gerenciamento de projetos, mais especificamente para desenvolvimento de *softwares*.

Ainda em forma de manifesto, esse grupo estava determinado a mudar de forma inovadora a maneira como os projetos eram realizados e entregues, conforme observado no AGILE MANIFESTO (2001).

Estamos descobrindo maneiras melhores de desenvolver *software*, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Por meio desse trabalho, passamos a valorizar: Indivíduos e interações mais que processos e ferramentas. *Software* em funcionamento mais que documentação abrangente. Colaboração com o cliente mais que negociação de contratos. Responder a mudanças mais que seguir um plano. Ou seja, mesmo havendo valor nos itens à direita (do termo 'mais que'), valorizamos mais os itens à esquerda (AGILE MANIFESTO, 2001).

O modelo ágil de gerenciamento de projetos atua com pequenas iterações focadas no produto, com uma participação forte do cliente, e comumente as decisões são tomadas de forma conjunta por toda a equipe, facilitando a integração de novas funcionalidades.

Para Pressman (2011, p. 81): “A filosofia defende a satisfação do cliente e a entrega de incremental prévio; equipes de projeto pequenas e altamente motivadas; métodos informais; artefatos mínimos e, acima de tudo, simplicidade no desenvolvimento geral”.

Isso demonstra que esse modelo de gestão de projetos não depende de um planejamento massivo em sua fase inicial, porque a sua forma de atuação já se adapta aos novos requisitos e a novas ideias e necessidades que surgem no decorrer da execução dos projetos.

Sommerville (2011) descreve que:

O desenvolvimento ágil tem de ser gerenciado de modo que se faça o melhor uso com o tempo e os recursos disponíveis para a equipe. Isso requer do gerenciamento de projeto uma abordagem diferente, adaptada para o desenvolvimento incremental e para os pontos fortes dos métodos ágeis (SOMMERVILLE (2011, p. 50).

Conforme Pressman (2011, p. 81): “Os princípios de desenvolvimento priorizam a entrega mais que análise e projeto, (embora essas atividades não sejam desencorajadas); também priorizam a comunicação ativa e contínua entre desenvolvedores e clientes”. Uma demonstração dos princípios dos métodos ágeis encontra-se no Quadro 2.

Quadro 2 – Princípios dos métodos ágeis

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações
Entrega incremental	O software é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar as mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Por isso, projete o sistema de maneira a acomodar essas mudanças.
Manter a simplicidade	Focalize a simplicidade, tanto do software a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Fonte: Sommerville (2011)

O modelo ágil de gerenciamento de projetos é baseado no planejamento das atividades de forma rápida, detalhando a iteração de forma básica e com poucos artefatos como entregas. Por não priorizar um planejamento massivo no início do projeto, isso não quer dizer que o modelo ágil de gestão não documente as entregas, que são realizadas conforme o seu andamento.

Arakaki e Ribeiro (2006, p. 1597) destacam que o modelo:

[...] fundamenta-se pelo planejamento rápido, com reuniões intensivas e com a participação de todos os envolvidos com o objetivo de obter o plano de projeto aprovado e pela participação efetiva do cliente em todas as fases do projeto atuando na definição, validação e aprovação do trabalho a ser realizado em conjunto com a equipe do projeto, pelo ambiente de colaboração entre os membros da equipe e pela rápida incorporação de alterações durante o ciclo de vida do projeto (ARAKAKI; RIBEIRO, 2006, p. 1597).

Entre as pessoas que se reuniram durante a criação do manifesto ágil em fevereiro de 2001 estavam um dos fundadores do modelo *Scrum*, Ken Schwaber. O desenho desse modelo se iniciou em 1990, quando seus criadores Ken Schwaber e Jeff Sutherland (2016) buscavam alternativas ao modelo tradicional para gerenciar projetos de desenvolvimento de *softwares*.

Por definição Schwaber e Sutherland (2016), o *Scrum* é:

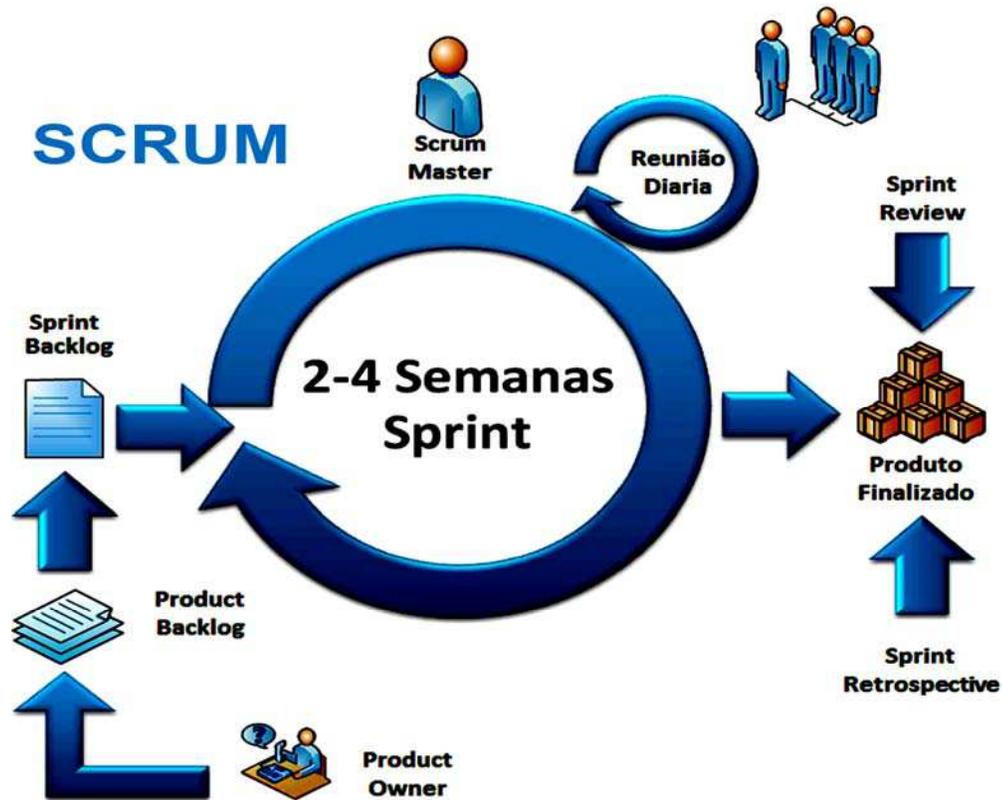
um *framework* dentro do qual as pessoas podem tratar e resolver problemas complexos e adaptativos, enquanto produtiva e criativamente entregam produtos com o mais alto valor possível. *Scrum* é: Leve, Simples de entender e Extremamente difícil de dominar (SCHWABER; SUTHERLAND, 2016, p. 3).

O termo tem como origem uma clássica formação de um jogo de *rugby*, em que um pequeno grupo de jogadores unido harmonicamente elenca os esforços individuais para atingir um objetivo comum.

O *Scrum* focaliza seus esforços nas entregas, que irão gerar maior valor para seu cliente com o menor desperdício possível. Por sua estrutura iterativa é possível garantir entregas periódicas chamadas de *sprints*, de duas a quatro semanas, não mais que isso, incrementando o produto final (SCHWABER; SUTHERLAND, 2016).

A Figura 5 demonstra a metodologia do *Scrum*.

Figura 5 – Metodologia scrum



Fonte: Sutherland e Schwaber (2012)

A Figura 5 demonstra como é interessante a organização do *Scrum*, que pode ser descrito em três papéis, três entregáveis e três eventos, descritos e exemplificados a seguir (SUTHERLAND; SCHWABER, 2012; SCHWABER; SUTHERLAND, 2016).

2.5.1 Papéis

Uma equipe que utiliza o SCRUM como modelo de gestão deve ser composta por apenas três papéis, que são:

- **1º Dono do Produto (Product Owner):** como o próprio nome já diz, dono do produto é origem da necessidade e detentor dos requisitos. É responsável por garantir o maior retorno do investimento possível (ROI), que deve identificar e priorizar as necessidades que mais dão retorno de forma que se tenha a decisão exata do que deve estar no topo da lista para o próximo Sprint, sempre a executar esse ciclo durante o projeto, onde se deve gerar novas atividades no backlog do produto;

- **2º Time:** equipe de desenvolvimento do projeto, que tem a responsabilidade pela produção dos requisitos determinados pelo dono do produto. O time no Scrum é responsável por dizer o que é possível entregar em cada Sprint e se auto-organizar de uma forma geral; e
- **3º Scrum Master:** deve atuar como facilitador do time, manter todos unidos e produtivos, remover os bloqueios que impedem o time de prosseguir com as atividades, cuidar da equipe para que não tenha nenhum tipo de interferência externa e guiá-la para seguir o modelo. Também demonstra de forma bem clara qual é o objetivo de cada item do backlog do produto e auxilia o dono, com técnicas para a modelagem do backlog.

2.5.2 Eventos

O SCRUM também tem eventos que são utilizados para criar uma rotina de trabalho, além de garantir transparência e inspeção criteriosa. Os principais eventos são:

- **Sprint:** determinado em no máximo um mês, quando toda equipe se engaja para entregar o planejado. Uma nova sprint se inicia após o encerramento da anterior e a cada início de uma sprint uma reunião de planejamento é feita (Sprint Planning), que não pode durar mais de oito horas para uma sprint de no máximo quatro semanas, composta por duas partes:
 - 1ª parte: o dono do produto e todo o time, incluindo o scrum master, revisam as atividades e prioridades do backlog do produto: discutem as metas, as necessidades, os motivos das prioridades e esclarecem a principal meta. Definem os requisitos necessários para declarar que as atividades da sprint estão realmente prontas e com a devida qualidade.
 - 2ª parte: definem a forma e como irão atacar os requisitos da Sprint, elencando quais serão as atividades selecionadas para fazerem parte dessa Sprint. Utiliza como referência o backlog do produto já devidamente priorizado.

- **Daily Meeting:** reunião diária que deve durar no máximo 15 minutos, realizada sempre no mesmo horário, de preferência com todos os participantes em pé. Esse é o momento em que todos os membros do time tem a oportunidade de se organizar e administrar os impedimentos que surgiram durante o progresso do Sprint e categoricamente devem explicar os três principais pilares da Daily Meeting:
 - O que executaram desde a última Daily Meeting;
 - O que irão entregar na próxima Daily Meeting; e
 - O que, ou qual foram os impedimentos que estão no caminho.
- **Revisão do Sprint (Sprint Review):** Sprint review é uma reunião de no máximo quatro horas para uma Sprint de quatro semanas, realizada com todos os membros do time, que tem como principal objetivo inspecioná-la e adaptá-la para que a equipe e o dono do produto evoluam baseando-se no feedback, que deve ser feito em ciclos repetitivos.
- **Retrospectiva do Sprint (Sprint Retrospective):** o Sprint retrospective é uma reunião de no máximo três horas para uma Sprint de quatro semanas, realizada com todos os membros do time, que tem como principal objetivo inspecionar e adaptar-se a si próprio em relação às pessoas, relacionamentos, processos e ferramentas utilizados no último Sprint, o que gera oportunidades para evolução pelo feedback aplicado no próximo Sprint que ocorre logo depois da Sprint Review e antes do Sprint Planning da próxima Sprint.

2.5.3 Artefatos

A documentação e outros elementos gráficos utilizados ao longo da utilização do modelo são denominados artefatos. Cada um tem uma função específica e se faz necessário para que ao final do projeto se tenha alguma documentação. Entre os principais artefatos da Scrum destacam-se:

- **Product Backlog:** demanda propriamente dita, ou seja, a necessidade do dono do produto que ainda não foi criada pelo time de desenvolvimento;
- **Sprint Backlog:** descreve os requisitos a serem executados durante cada Sprint; e

- **Burndown Chart:** modelo gráfico que exibe o comportamento do projeto ao longo dos Sprints, que de certa maneira exibe o status do projeto.

2.6 Principais características dos modelos *Waterfall* e *Scrum*

As principais características dos modelos de gestão de projetos abordados neste estudo podem ser observadas no Quadro 3.

Quadro 3 – Principais características dos modelos *waterfall* e *scrum*

Contexto	Características <i>Waterfall</i>	Características <i>SCRUM</i>
Objetivo principal	Orientado por atividade e centrado em processo	Orientado por produto e centrado em pessoas
Documentação	Iterativa	Linear
Tipo de Projeto	Estáveis e com baixo nível de mudanças.	Projetos com mudanças constantes e que necessitam de respostas rápidas
Tamanho	Aplicável em projetos de todos os tamanhos. Mais efetivo em projetos de maior duração.	Mais efetivo em projetos pequenos (5-10 pessoas).
Gerente de Projeto	Controle total do projeto	Papel de facilitador ou coordenador
Equipe do Projeto	Atuação com papéis claros e bem definidos.	Atuação colaborativa em todas as atividades do projeto
Cliente	Participa das fases iniciais de requisitos e das validações dos produtos.	Essencial. Deve ser parte integrante da equipe do projeto.
Planejamento	Detalhado e os envolvidos tem o papel de validação, não participam da elaboração do planejamento.	Curto e com a participação de todos os envolvidos na elaboração do planejamento.
Arquitetura	Definida com foco em todo o projeto e na reutilização.	Aplicação de design simples. Evolui junto com o projeto e baseia-se na refatoração.
Modelo de Desenvolvimento	Cascata, espiral e iterativo	Iterativo e incremental
Modelo de Gestão	Descentralizado	Gerenciado
Liderança	Liderança colaborativa e servidora	Comando e controle
Comunicação	Formal	Informal
Controle de Mudanças	Processo formal de identificação e aprovação entre os envolvidos. incorporação de novos requisitos pode ser lento e caro.	Dinâmico e com rapidez de incorporarão nas iterações.
Garantia da Qualidade	Centralizado no cliente	Centralizado no processo
Retorno do Investimento	Ao longo do projeto	Final do Projeto
Envolvimento do cliente	Alto e ao longo do projeto	Varia de acordo com a vida do projeto

Fonte: adaptado de Arakaki e Ribeiro (2006) e SCRUMstudy (2014)

O método tradicional é aplicado em qualquer ambiente de projeto, mas nas situações críticas envolvendo prazos restritos apresenta dificuldades em responder com rapidez às mudanças. O gerente do projeto tem atuação forte e centralizadora nas decisões e é o responsável pelo sucesso do projeto.

Nesse método a equipe participa de várias atividades, mas existe uma definição clara de papéis e cada membro tem suas tarefas definidas, o que limita a influência e uma maior colaboração durante a execução do projeto. O cliente tem participação muito importante nas fases iniciais e no levantamento de requisitos, mas à medida que o projeto evolui sua participação diminui, restringindo-se a validações de artefatos e dos produtos gerados pelo projeto.

Esse planejamento é extenso e detalhado e cria diversos artefatos, como cronograma de atividades e pontos de controle e procedimentos, que direcionam a geração dos produtos previstos e a coordenação do processo.

A comunicação dentro da equipe do projeto baseia-se na documentação gerada nas diversas etapas do processo de produção do *software*, como a especificação de requisitos, protótipos, especificação de projeto e outras, e é uma referência para a construção dos produtos previstos (ARAKAKI; RIBEIRO, 2006).

Já a base do gerenciamento ágil está: em responder rapidamente às alterações; na confiança nos membros da equipe; e na realização de entregas frequentes ao cliente, que estabelece um ambiente colaborativo e integrado para a realização do projeto. A flexibilidade do modelo sugere sua aplicação em qualquer tipo de ambiente de projetos.

O planejamento deve ser realizado de forma rápida, com a participação de todos os envolvidos. A equipe trabalha em grupos pequenos. Os requisitos a serem implementados em cada ciclo são avaliados em conjunto e as decisões tomadas de forma colaborativa entre todos os envolvidos.

O cliente deve ser membro efetivo da equipe com conhecimento e domínio do processo de negócio que está sendo construído e com poder de decisão nas questões e alterações que surgirem durante a fase de construção. A comunicação é interpessoal e informal (ARAKAKI; RIBEIRO, 2006).

A gestão de projetos *waterfall* baseia-se em um plano abrangente, que descreve detalhadamente os objetivos do projeto. Muitas vezes esses projetos conseguem cumprir seus objetivos, mas sem satisfazer o cliente.

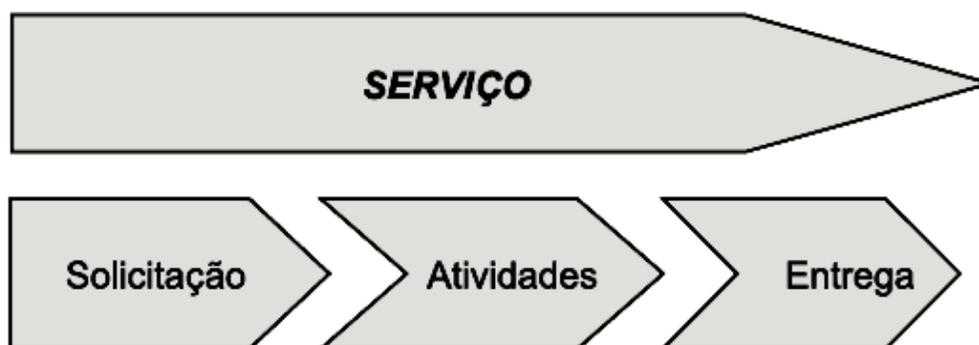
Um dos principais objetivos do modelo *Scrum* é manter o foco na satisfação das necessidades e exigências do cliente e facilitar a colaboração entre todas as partes envolvidas no projeto, incluindo o cliente, a equipe de negócios e as partes interessadas. As entregas potencialmente utilizáveis para os clientes geram retorno do investimento mais rápido do que no modelo *waterfall* (SCRUMSTUDY, 2014).

2.7 Gerenciamento de serviços e operações em TI

Para se entender o gerenciamento de serviços e operações em TI, primeiramente é preciso definir o que é um serviço. Para Freitas (2013, pp. 26-27) um serviço é composto de uma solicitação, um conjunto de atividades a serem realizadas e de um resultado ou uma entrega para quem solicitou o serviço, ou seja, um serviço de TI é um conjunto de atividades técnicas realizadas para entregar os serviços de negócio aos clientes.

A Figura 6 representa de forma simplificada os processos que compõem um serviço.

Figura 6 – Componentes de serviço



Fonte: Freitas (2013)

Pode-se observar, também, que a definição de serviços se baseia nos meios de entregar valor para o cliente, de forma a facilitar os resultados que pretende atingir, sem a propriedade de custos e riscos específicos do fornecimento do serviço.

De acordo com Soula (2013):

Serviços são meios de entregar valor aos clientes de forma a facilitar seus resultados de maneira positiva, visando alcançar os menores custos e riscos. Tal fato se dá a partir de uma visão facilitadora para atingir tais resultados, reforçando capacidades para realizar tarefas associadas aos serviços oferecidos, diminuindo o efeito restritivo causado por alguma razão. O resultado final pode ser percebido no aumento da probabilidade de atingir os resultados desejados (SOULA, 2013, p.35-36),

Segundo o Axelos (2012, p. 106), um serviço é o meio de fornecer valor a clientes e facilitar a obtenção de resultados que eles desejam, sem que tenham que arcar com a propriedade de determinados custos e riscos. O termo 'serviço' é algumas vezes utilizado como sinônimo para serviço principal, serviço de TI ou pacote de serviço.

Os serviços de TI podem ser definidos por qualquer serviço originário de um provedor cuja composição se dê pela combinação de tecnologia da informação, pessoas e processos (SOULA, 2013).

O Quadro 4 demonstra os tipos de serviços de TI, sua definição e características.

Quadro 4 – Definições e características dos tipos de serviços de TI

Tipo de serviço	Definição	Característica(s) / Exemplo(s)
Serviço de suporte (às vezes chamado de serviço de infraestrutura)	Serviço não usado diretamente pelo negócio, mas exigido do provedor de serviços de TI para que outros serviços sejam produzidos e entregues	Habilita processos e serviços, mas não são diretamente visíveis aos clientes. Exemplos: serviços de diretório, DNS ...
Serviço voltado para o cliente interno	Suporta diretamente um processo de negócio gerenciado por uma outra área da organização	Depende de um conjunto integrado de serviços de suporte. Exemplos: ERP, serviço de relatório, ...
Serviço voltado para o cliente externo	Diretamente fornecido pela TI para um cliente externo	Visa atender aos objetivos de negócio definidos na estratégia da organização. Exemplos: <i>supply chain</i> , <i>e-business</i> , ...

Fonte: adaptado de Soula (2013)

Com base nas visões definidas para serviços de TI citadas, faz-se necessário observar agora o processo de gerenciamento de tais serviços. Para que isso possa ser possível é necessário contextualizar o gerenciamento de serviços, que corresponde a um conjunto especializado de habilidades organizacionais para fornecer valor a clientes na forma de serviços (AXELOS, 2012).

Ainda se pode descrever o gerenciamento de serviços como um conjunto de capacidades organizacionais especializadas que permitem entregar valor aos clientes na forma de serviços e devem ser capazes de garantir a qualidade dos serviços prestados ao negócio com maior valor agregado (SOULA, 2013).

Com a finalidade de alcançar esse objetivo, as empresas cada vez mais têm investido na melhoria da gestão de TI, para que possam acompanhar a competitividade do mercado em busca de aperfeiçoar os processos com a principal finalidade de satisfazer as necessidades, expectativas e desejos de seus clientes e usuários.

Para que essas organizações alcancem excelência e integração total do gerenciamento de TI com a estratégia de negócios, é preciso entregar os serviços com qualidade, gerenciar seus processos de forma eficaz e garantir a integração com a organização e suas políticas.

Algumas boas práticas devem ser adotadas e seguidas. Os modelos ITIL e ISO/IEC 20000 são as principais abordagens adotadas e serão descritas nas subseções 2.7.1 e 2.7.2.

2.7.1 Modelo ITIL

Surgiu inicialmente em 1986, no Reino Unido, pelo órgão do governo CCTA (*Central Computer Telecommunication Agency*), que iniciou a criação de um programa para desenvolver um conjunto de boas práticas com o propósito de garantir a eficiência em relação à padronização dos serviços de TI que eram contratados pelos órgãos do governo britânico.

Em 1988, por meio da CCTA desenvolveu o método GITIM (*Government Information Technology Infrastructure Method*) focado na padronização do gerenciamento de serviços e práticas de TI, em busca de melhorar o atendimento, processos, terminologia, desempenho, qualidade, custos, capacidade e disponibilidade.

Sua intenção era atender à crescente dependência do governo em TI, pois percebia-se que cada empresa privada com a qual o governo mantinha contrato utilizava suas próprias práticas de gestão de TI, podendo causar esforços, desentendimentos e custos excessivos por não se adotar o mesmo padrão.

No ano seguinte, em 1989, o GITIM foi renomeado para ITIL (versão 1 ou V1), que significa *Information Technology Infrastructure Library* (Biblioteca de Infraestrutura de Tecnologia da Informação), por conta da crescente aderência ao método, que passou a ser utilizado como modelo de boas práticas, como resultado do crescente interesse nas práticas propostas por empresas externas ao governo britânico, refletindo a visão de que o ITIL não era uma regra.

Ainda em 1989 ocorreu a publicação dos primeiros livros, num total de 31, que abordam os aspectos fundamentais para provisão de serviços de TI, como exemplo: gerenciamento de nível de serviço, *help-desk*, planejamento de contingências e gerenciamento de mudança.

Em 1990 ocorreu a publicação dos livros para gerenciamento de problemas, gerenciamento de configuração e gerenciamento de custos.

No ano seguinte, em 1991, os livros para controle e distribuição de *softwares* foram publicados e também ocorreu a criação do *IT Information Management Forum* (ITIMF), uma organização internacional sem fins lucrativos, responsável por liderar o desenvolvimento do modelo e que se dedica ao desenvolvimento do gerenciamento de serviços de TI.

Logo em seguida, em 1992, foi publicado um livro que trata do gerenciamento de disponibilidade e foi consolidada após cinco anos o ITIMF, que mudou o nome para *IT Service Management Forum* (ITSMF). Na década de 2000 ocorreram diversas alterações no modelo ITIL, que ficou conhecido como ITIL V2 (Versão 2). Segundo Freitas (2013, p. 62), Bom (2012) e Axelos (2012), as principais mudanças no modelo foram:

- Ano de 2000 – publicação do livro *Service Support*;
- Ano de 2001 – publicação do livro *Service Delivery* e o CCTA foi transferido para o Office of Government Commerce (OGC), tornando-se responsável pelo ITIL;
- Ano de 2002 - publicação dos livros *Application Management*, *Implementation Planning for ITSM* e *ICT Infrastructure Management*;
- Ano de 2003 – publicação do livro *Software Asset Management*;
- Ano de 2004 – publicação da ISO/IEC 20000; e
- Ano de 2006 – publicação do Glossário do ITIL V2.

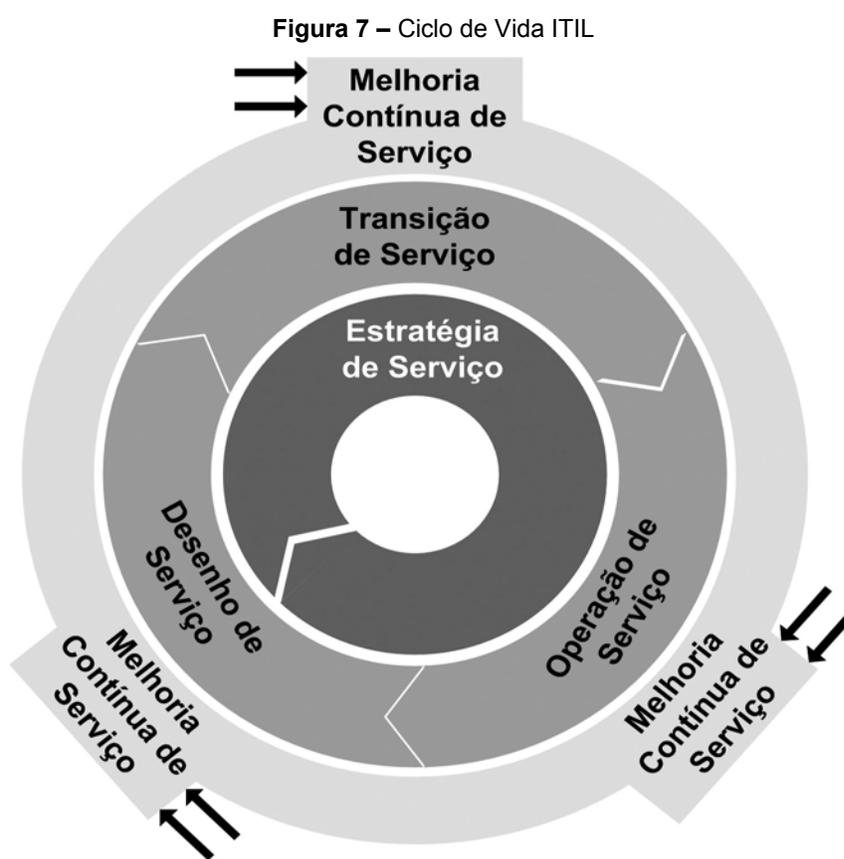
Ainda na década de 2000, mais especificamente no ano de 2007, ocorreu mais uma grande mudança no modelo com a publicação do ITIL V3 (Versão 3), quando cinco principais livros estabeleceram o seu ciclo de vida em uma estrutura que se mantém até os dias de hoje.

Em 2010, o ITIL e alguns outros modelos de boas práticas foram transferidos da OGC para o *Cabinet Office*, órgão do gabinete do primeiro-ministro britânico. Em 2011 ocorreu a publicação do ITIL edição 2011, com a introdução de alguns novos processos e funções, além de melhorias menores para facilitar o entendimento e uso do modelo.

Já em 2013, o *Cabinet Office* e a empresa Capita formaram uma *joint venture* chamada AXELOS, que se tornou a principal responsável pelos modelos de boas práticas criados pelo governo britânico (ITIL, PRINCE2, M_o_).

A estrutura do modelo ITIL atualmente em prática (ITIL edição 2011) é composta por um ciclo de vida que colabora para uma gestão mais abrangente e integrada de serviços, pois não foca os seus cinco grupos de processos de serviços isoladamente, contribuindo de forma objetiva para que se possa medir e gerenciar o valor que os serviços de TI agregam ao negócio (FERNANDES; ABREU, 2014, p. 227).

A Figura 7 traz a representação desse ciclo de vida e seus processos principais correspondentes às publicações podem ser observadas.



Fonte: Fernandes e Abreu (2014)

O ITIL é visto ao redor do mundo como uma biblioteca de boas práticas que podem ser usadas pelas organizações para constituir e melhorar suas habilidades em gerenciamento de serviços.

É formado por suas cinco principais publicações, e cada uma está associada aos estágios do ciclo de vida citado na Figura 7, com as orientações necessárias para uma abordagem integrada de gerenciamento de serviços, descritos e exemplificados a seguir (FERNANDES; ABREU, 2014, p. 228-229):

- **Estratégia de Serviço:** orienta sobre como visualizar o gerenciamento de serviços não somente como uma capacidade organizacional, e sim como um ativo estratégico, e descreve os princípios inerentes à prática desta disciplina que são úteis para criar políticas, diretrizes e processos de gerenciamento de serviços ao longo do ciclo de vida de serviço. Entre os tópicos abordados nesta publicação estão a criação de valor através dos serviços, os ativos de serviço, provedores e tipos de serviços, gerenciamento financeiro, portfólio de serviços, desenvolvimento organizacional, riscos estratégicos, entre outros.
- **Desenho de Serviço:** fornece orientação para o desenho e desenvolvimento dos serviços e das práticas de gerenciamento de serviços, detalhando aspectos do gerenciamento do catálogo de serviços, do nível de serviço, da capacidade, da disponibilidade, da continuidade, da segurança da informação e dos fornecedores, além de mudanças e melhorias necessárias para manter ou agregar valor aos clientes ao longo do ciclo de vida de serviço.
- **Transição de Serviço:** orienta sobre como efetivar a transição de serviços novos e modificados para ambientes operacionais gerenciados, detalhando os processos de planejamento e suporte à transição, gerenciamento de mudanças, gerenciamento da configuração e dos ativos de serviço, gerenciamento da liberação e da distribuição, teste e validação de serviço, avaliação e gerenciamento do conhecimento.
- **Operação de Serviço:** descreve a fase do ciclo de vida do gerenciamento de serviços que é responsável pelas atividades do dia a dia, orientando sobre como garantir a entrega e o suporte a serviços de forma eficiente e eficaz em ambientes operacionais gerenciados, e detalhando os processos de gerenciamento de eventos, incidentes, problemas, acesso e de cumprimento de requisições; e
- **Melhoria Contínua de Serviço:** orienta, através de princípios, práticas e métodos de gerenciamento da qualidade, sobre como fazer sistematicamente melhorias incrementais e de larga escala na qualidade do serviço, nas metas de eficiência operacional, na continuidade do serviço, com base no modelo PDCA (*Plan-Do-Check-Act*).

O Quadro 5 demonstra os processos e funções do modelo ITIL distribuídos entre seus cinco estágios.

Quadro 5 – Processos e funções da ITIL

Etapa do Ciclo de Vida	Processos	Funções
Estratégia de Serviço	<ul style="list-style-type: none"> •Gerenciamento Estratégico para Serviços de TI •Gerenciamento de Portfólio de Serviço •Gerenciamento Financeiro para Serviços de TI •Gerenciamento de Demanda •Gerenciamento do Relacionamento com o Negócio 	
Desenho de Serviço	<ul style="list-style-type: none"> •Coordenação de desenho •Gerenciamento de Catálogo de Serviço •Gerenciamento de Nivel de Serviço •Gerenciamento de Disponibilidade •Gerenciamento de Capacidade •Gerenciamento de Continuidade de Serviço de TI •Gerenciamento de Segurança da Informação •Gerenciamento de Fornecedor 	
Transição de Serviço	<ul style="list-style-type: none"> •Planejamento e Suporte da Transição •Gerenciamento de Mudança •Gerenciamento de Configuração e de Ativo de Serviço •Gerenciamento de Liberação e Implantação •Validação e Teste de Serviço •Avaliação da Mudança •Gerenciamento de Conhecimento 	
Operação de Serviço	<ul style="list-style-type: none"> •Gerenciamento de Evento •Gerenciamento de Incidente •Cumprimento de Requisição •Gerenciamento de Problema •Gerenciamento de Acesso 	<ul style="list-style-type: none"> •Central de Serviços •Gerenciamento Técnico •Gerenciamento de Operações de TI •Gerenciamento de Aplicativo
Melhoria Contínua de Serviço	<ul style="list-style-type: none"> •Processo de Melhoria de Sete Etapas 	

Fonte: adaptado de Fernandes e Abreu (2014).

2.7.2 Modelo ISO 20000

De acordo com Fernandes e Abreu (2014, p. 264), a Norma BS 15000 foi publicada em 2000 pela BSI (*British Standards Institution*). Passou a ser o primeiro padrão mundial voltado ao gerenciamento de serviços de TI, priorizou as disciplinas de suporte e entrega de serviços de TI alinhados às diretrizes do ITIL.

Em 2002 foi republicada e passou a ser estruturada em duas partes: especificação focada em requisitos básicos da norma e código de prática com o foco em diretrizes de suporte.

Em dezembro de 2005, a BS 15000 evoluiu para o padrão internacional ISO/IEC 20000 com a colaboração dos grandes órgãos internacionais ISO (*International Organization for Standardization*) e IEC (*International Electrotechnical Commission*). Tornou-se relevante como padrão internacional para as organizações de TI, passou a ser adotada também nos Estados Unidos e América Latina.

No ano 2010 foi melhorada com o refinamento de algumas definições, bem como o alinhamento com as normas ISO/IEC 9001 e ISO/IEC 27001, com o ITIL V3 e a inclusão do conceito de Sistema de Gestão de Serviços (SGS) (FERNANDES; ABREU, 2014, p. 264).

Ainda segundo Fernandes e Abreu (2014), o principal objetivo desse modelo é:

Regulamentar um padrão para o Gerenciamento de Serviços de TI, através da uniformização dos conceitos e da visão dos processos que o implementam, permitindo assim que os provedores de serviços de TI compreendam os meios através dos quais poderão planejar, executar, verificar e melhorar continuamente a qualidade dos serviços entregues, em conformidade com os requisitos estabelecidos junto ao negócio e a seus clientes (FERNANDES; ABREU, 2014, p. 265).

O Quadro 6 representa as cinco partes nas quais a Norma ISO/IEC 20000 foi estruturada.

Quadro 6 – Estrutura ISO/IEC 20000

Etapa	Processos	Funções
Parte 1	Requisitos do Sistema de Gestão de Serviços	<ul style="list-style-type: none"> • Especificação formal da norma • Define os requisitos para o gerenciamento de serviços, dentro de um nível aceitável de qualidade e em conformidade com os requisitos do negócio • Descreve o que deve ser levado em consideração na implementação do gerenciamento de serviços de TI, visando a certificação dos processos relacionados em relação aos requisitos da norma.
Parte 2	Código de Prática	<ul style="list-style-type: none"> • Conjunto de diretrizes baseadas na experiência do mercado • Orientações para as empresas de serviços a planejar melhorias em seus serviços ou Orientações para as empresas de serviços se prepararem para serem auditadas e certificadas na norma, em relação a cada um dos requisitos presentes na Parte 1.
Parte 3	Diretrizes de Escopo	<ul style="list-style-type: none"> • Orientações para definição do escopo e a aplicabilidade da norma aos diversos tipos de organizações de serviços de TI.
Parte 4	Modelo de Referência de Processos	<ul style="list-style-type: none"> • Orientações para a definição dos processos de Gerenciamento de Serviços de TI em alinhamento com a ISO/IEC 15504-2 – Auditoria de processos de TI.
Parte 5	Exemplo de Plano de Implementação	<ul style="list-style-type: none"> • Apoio à preparação para a implementação de um Sistema de Gerenciamento de Serviços de TI

Fonte: adaptado de Fernandes e Abreu (2014)

Apenas a parte 1 do padrão ISO/IEC 20000 é relevante para a elaboração deste trabalho, pois reflete uma abordagem principal dos requisitos que contêm as informações necessárias para a implantação do gerenciamento de serviços de TI, em que a Norma ISO/IEC 20000 descreve um sistema de gestão de serviços (SGS) que controla as atividades do provedor de serviços.

O SGS contém todas as regras para que o desenho, a transição e a entrega dos serviços e suas melhorias possam ser realizadas conforme a norma rege.

A Figura 8 demonstra a visão do SGS pela perspectiva da norma.

Figura 8 – Sistema de Gestão de Serviços, na visão da ISO/IEC 20000



Fonte: Fernandes e Abreu (2014)

A Parte 1 da norma ISO/IEC 20000 é composta por nove seções, compiladas e explicadas no Quadro 7.

Quadro 7 – Estrutura da parte 1 da norma ISO/IEC 20000

Seção	Funções
Escopo	Nesta seção são descritos os propósitos e cenários para os quais a norma é recomendada, e o Sistema de Gestão de Serviços é apresentado.
Referências Normativas	Nesta seção é referenciada a Parte 2 da norma (Diretrizes para aplicação dos sistemas de gestão de serviços).
Termos e Definições	Nesta seção é estabelecido um glossário contendo a terminologia e as definições aplicáveis aos propósitos da norma. Vários dos termos contidos nesta seção fazem parte do contexto da ITIL.
Requisitos gerais para um SGS	Nesta seção é especificado os requisitos que também são encontrados em outras normas ISO para o foco do Sistema de Gestão de Serviços, dos quais: <ul style="list-style-type: none"> • Responsabilidade da Direção; • Governança de processos operados por outras partes; • Gerenciamento da documentação; • Gerenciamento de recursos; • Estabelecimento e melhoria do SGS;
Desenho e transição de serviços novos ou modificados;	Esta seção visa assegurar que serviços novos e/ou mudanças em serviços possam ser planejados, desenhados, desenvolvidos, entregues e gerenciados, considerando aspectos como escopo, orçamento, alocação de recursos, níveis de serviço e processos já existentes etc. Neste ponto, percebe-se um total alinhamento com as práticas de desenho de serviços e transição de serviços da ITIL 2011.
Processos de entrega de serviço	Esta seção descreve a abordagem da norma para os processos relacionados à entrega de serviços: <ul style="list-style-type: none"> • Gerenciamento do nível de serviço; • Relatos de serviço; • Gerenciamento da continuidade e disponibilidade do serviço; • Orçamento e contabilização para serviços; • Gerenciamento da capacidade; • Gerenciamento da segurança da informação.
Processos de relacionamento	Nesta seção é descrito os aspectos do relacionamento do provedor de serviços dentro da cadeia de valor que engloba a prestação do serviço, ou seja, com os clientes que recebem o serviço e com seus demais fornecedores de produtos, recursos e outros serviços: <ul style="list-style-type: none"> • Gerenciamento de relações de negócio; • Gerenciamento de fornecedores.
Processos de resolução	Englobam os processos de gerenciamento de incidentes, requisições de serviço e de problemas, assim como a sua priorização e o estabelecimento e a manutenção de soluções de contorno.
Processos de controle	Englobam os processos que tratam os componentes dos serviços e das mudanças pelas quais estes passam ao longo do tempo

Fonte: Fernandes e Abreu (2014)

Dessa forma, observa-se que para gerir projetos as organizações utilizam dois modelos mais usuais, *Waterfall* e *Ágil/Scrum*. Após a finalização, os projetos são disponibilizados para uso, entrando em operação. Os modelos mais utilizados de gestão de operações são ITIL e ISO/IEC 20000.

Essas duas áreas - Dev e Ops - precisam ser interligadas para que a eficácia de desenvolvimento de projetos e operações possa entregar maior valor em seu resultado final, eliminando desperdício de tempo e dinheiro.

A subseção seguinte apresenta e analisa essa tendência.

2.8 DevOps – História

Durante a conferência *Agile* no ano de 2008, Andrew Clay Shafer e Patrick Debois discutiram quanto à infraestrutura ágil aplicada em TI, pois era algo que ainda não estava consolidado.

Essa metodologia estava presente somente nos projetos de desenvolvimento de software, gerando alguns conflitos para a área de infraestrutura, que não tinha a mesma rapidez e velocidade para entregar a infraestrutura e as operações pós-implantação do software, como ocorria com as entregas dos desenvolvedores de projetos que já utilizavam ágil como metodologia (KIM *et al.*, 2016; MUELLER, 2016).

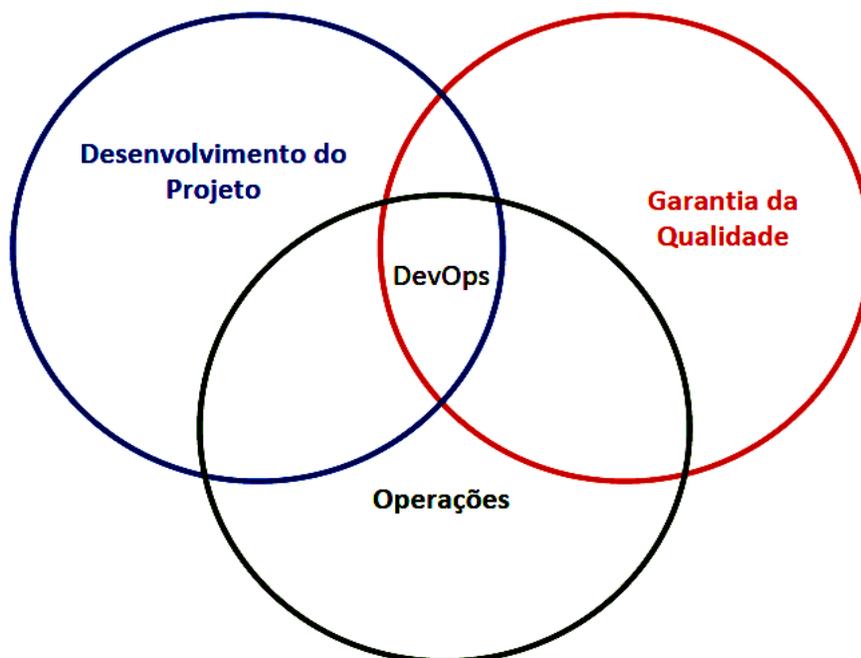
No ano seguinte, 2009, durante a conferência *Velocity* (Velocidade), John Allspaw e Paul Hammond fizeram uma apresentação chamada ‘*10 Deploys per Day: Dev and Ops Cooperation at Flickr*’ - 10 implantações por dia: cooperação entre Dev e Ops no *Flickr* – quando apresentaram a todos como eles criaram objetivos compartilhados entre Dev e Ops, usando práticas de integração contínua para fazer implantação de todo o trabalho de TI. Foi o primeiro relato do uso de um novo conceito que viria a originar o *DevOps* (KIM; *et al.*, 2016; ALLSPAW; HAMMOND, 2009).

Inspirado pela apresentação de Allspaw e Hammond e aliado às dúvidas já discutidas em 2008 por Shafer e Debois com relação aos problemas recorrentes entre as equipes, Patrick Debois organizou outro evento, chamado *DevOps Days*, que ocorreu no ano seguinte, em 2009, em Ghent, Bélgica (KIM; *et al.*, 2016; MUELLER, 2016).

DevOps é uma junção das palavras “**DE**veloper” (Desenvolvedor) e “**OP**eration**S**” (Operações). Nessa primeira conferência foram propostas novas ideias e abordagens para automatizar os processos que ocorrem desde o desenvolvimento de um novo *software* até sua operação/sustentação pós-entrega, propondo uma colaboração mútua entre equipes de desenvolvimento e operações.

A ideia principal é não perder tempo com rotinas que possam ser automatizadas, deixando os profissionais livres para focar suas atividades em soluções inteligentes (MUELLER, 2016).

A Figura 9 demonstra como *DevOps* deve ser, integrando as áreas distintas do desenvolvimento à entrega do projeto de *software*.

Figura 9 – Demonstração *DevOps*

Fonte: adaptada de Pepin (2014)

2.8.1 Conceitos

DevOps é um movimento cultural e profissional que tem como principal objetivo a colaboração, comunicação e integração entre os profissionais envolvidos nas áreas de desenvolvimento de *softwares*, qualidade (testes) e operações, servindo como estrutura e modelo a serem seguidos para que esses profissionais possam trabalhar de forma conjunta e se tornem mais efetivos (KIM; *et al*, 2016; ATLISSIAN, 2017).

Segundo HPE (2016, p. 3), o conceito *DevOps* tem sido amplamente aceito como a forma pela qual as empresas de todos os tipos devem construir, testar e entregar *software* no futuro próximo.

Neste mundo cada vez mais móvel, mutante e sempre conectado, o *DevOps* capacita as empresas a acompanhar o rápido ritmo de entrega que os usuários exigem.

O alinhamento dos processos, ferramentas e responsabilidades entre as equipes de desenvolvimento de *software* e equipes de operações é primordial para que as entregas em produção possam ser mais rápidas e com um alto grau de qualidade.

Enquanto a metodologia ágil aproximou o negócio às equipes de desenvolvimento com a redução da distância entre essas áreas, o *DevOps* busca trazer a agilidade para as entregas com a diminuição da distância entre desenvolvimento e operações.

No desenvolvimento de *software* tradicional, o aumento da velocidade geralmente resulta em uma qualidade menor e maior vulnerabilidade nas alterações executadas, porém as iniciativas do *DevOps* podem gerar melhorias nas quatro principais dimensões: velocidade e automação, qualidade, produtividade e segurança.

Além desses itens, a mentalidade ágil também se faz necessária e precisa ser avaliada, pois *DevOps* aproxima as práticas de desenvolvimento ágil com testes e implantação, fazendo uso amplo de ferramentas de automação (HPE 2016).

Logo a seguir estão descritas as dimensões em que o *DevOps* procura trabalhar para melhorar a integração entre Dev e Ops.

2.8.1.1 Velocidade e automação

O *DevOps* foi criado como uma forma de os desenvolvedores obterem recursos no mercado rapidamente em resposta às demandas dos usuários para um conteúdo cada vez mais rápido e sempre melhor.

A velocidade não só oferece vantagem competitiva para a empresa, mas também melhora a forma como as empresas trabalham com TI. Tecnicamente, o *DevOps* implica automatizar tanto quanto possível para acelerar processos e reduzir erros humanos (HPE, 2016).

Os processos manuais dificultam a entrega mais rápida e também fazem com que as pessoas executem atividades cansativas e pouco criativas. Assim, a automação de processos manuais é fundamental para o *DevOps* e para oferecer um melhor *software* e de forma rápida.

Os processos automatizados e repetitivos são mais fáceis de entender, mais fáceis de auditar, mais fáceis de proteger e mais fáceis de melhorar. A automação permite um aprendizado mais veloz e uma resposta mais rápida às demandas do cliente e do mercado (PUPPET, 2016).

2.8.1.2 Qualidade

As organizações contam com *software* para fornecer valores e experiências diferenciadas para seus clientes.

Atualmente o *software* é utilizado como uma ferramenta chave para os negócios, gerando bons resultados e com isso novas necessidades devem ser entregues rapidamente e com alta qualidade. Muitas organizações, no entanto, acreditam que a entrega com velocidade e qualidade não é fácil de ser alcançada e muitas barreiras devem ser superadas para que isso se torne verdade (FORRESTER, 2016).

Com o *DevOps* não é necessário penalizar a qualidade para se obter entregas com mais velocidade, pois a ideia em torno do aumento de velocidade é reduzir as entregas de *software* em pequenas partes, feitas de forma mais rápida, com melhor qualidade e com os riscos reduzidos e mais controlados, criando qualidade em toda a sua cadeia de entrega (HPE, 2016).

Segundo KIM; *et al.* (2016), obter respostas de forma rápida é fundamental para alcançar qualidade, confiabilidade e segurança no fluxo de valor da tecnologia, em que a resolução dos problemas, à medida que ocorrem, desenvolve novos conhecimentos, eleva a qualidade do *software* e aperfeiçoa continuamente o trabalho. Utilizar reuniões de retrospectiva e/ou lições aprendidas se torna comum e rotineiras para todos os envolvidos, além de aumentar ainda mais a integração da equipe.

Também se nota que há um impacto direto e negativo na qualidade do *software* e na experiência do usuário quando existe certa inconsistência entre as suas necessidades e o que realmente foi desenvolvido e entregue.

Segundo HPE (2016), quando os requisitos de usuário e os testes que os validam não estão vinculados, as mudanças nos requisitos podem levar a testes falhos que não mostrarão quantos requisitos foram atendidos.

Quando os requisitos e os testes estão sincronizados, qualquer alteração em um requisito resulta em visibilidade correspondente na cobertura do caso de teste. A ligação direta e positiva afeta e indica melhor qualidade do *software*, menor conflito entre analistas de negócios, desenvolvedores e equipe de controle de qualidade.

2.8.1.3 Produtividade

Segundo Badrinarayanan; *et al.* (2013, p. 18), a economia em rápida evolução e a expansão dos mercados e a nova concorrência pressionam as empresas para que executem e produzam resultados excepcionais. Algumas dessas pressões são diretamente transferidas às equipes de *software* que buscam melhorar a produtividade para oferecer tecnologia e inovação mais rapidamente a fim de suportar os objetivos de negócios. Isso é alcançado com a utilização de ferramentas, métodos e conceitos que aumentam a produtividade e reduzem o desperdício.

O propósito principal do *DevOps* é auxiliar a TI a criar mais valor para o negócio, porém isso não ocorrerá se os esforços não estiverem adequadamente alinhados com o negócio e a estratégia da organização, não importa o quão rápido se possa produzir *software* (HPE, 2016).

Uma das formas de alcançar esse objetivo é tornar as atividades de ambos os times, desenvolvimento e operações, tão visíveis quanto possível. A utilização de quadros visuais físicos ou eletrônicos é uma das melhores ferramentas para esse objetivo, contendo divisões em forma de colunas, que podem ser: atividades a serem executadas, atividades em execução e atividades finalizadas. Cada atividade é representada por cartões físicos ou eletrônicos.

O fluxo de movimentação de uma atividade no quadro se origina da parte esquerda para a direita, de forma a facilitar a visualização do que está para ser feito: coluna da esquerda do quadro; o que está em execução: coluna do meio do quadro; e o que já foi entregue: coluna da direita. Isso auxilia a toda a equipe com a visualização rápida das atividades e também contribui para a produtividade em geral (KIM; *et al.*, 2016; PEREIRA, 2017).

Quando o *DevOps* é aplicado de forma correta, as organizações têm um aumento da velocidade e aumento da qualidade das entregas feitas por TI, pois alinhada à estratégia de negócio a equipe se concentra em criar recursos, que a organização vai realmente consumir; reduzir o desperdício; e aumentar o valor no negócio com a utilização de tecnologias e processos que melhoram a produtividade do desenvolvedor e do time de operações, com foco em reduzir o tempo de implantação do código e infraestrutura mais confiável (HPE, 2016; PUPPET, 2017).

2.8.1.4 Segurança

A principal característica do *DevOps* é buscar a integração total entre os donos de produtos, equipes de desenvolvimento, garantia da qualidade, operações de TI e segurança da informação para que trabalhem em conjunto, não só ajudando uns aos outros, mas também para garantir que a organização de forma geral seja bem-sucedida.

Ao trabalhar focado em um objetivo comum, eles permitem o fluxo rápido do trabalho planejado em produção, como, por exemplo, realizando dezenas, centenas ou mesmo milhares de implementações de código por dia, ao mesmo tempo em que alcançam a estabilidade, a confiabilidade, a disponibilidade e a segurança da classe mundial (KIM; *et al*, 2016).

A velocidade é extremamente importante no *DevOps*, mas atualmente segurança também é tão importante quanto velocidade. Quando se realiza uma modificação em alguma aplicação ou serviço, as equipes devem garantir que não irão introduzir vulnerabilidades ao executarem tais mudanças

Liberar novos recursos que introduzem novas vulnerabilidades, ou expor as já existentes, é uma forma negativa de impactar as organizações, fazendo com que as equipes se esforcem e levem certo tempo para consertá-las (HPE, 2016).

Segundo KIM *et al.* (2016, p. 483), um dos objetivos do *DevOps* é ter equipes envolvidas com a segurança da informação o mais cedo possível, ao contrário de se envolver no final do projeto.

Uma maneira de fazer isso é convidar a equipe de segurança para as demonstrações de produtos no final de cada intervalo de desenvolvimento, para que se possa entender melhor os objetivos da equipe no contexto dos objetivos organizacionais, observar suas implantações como estão sendo construídas e fornecer orientação e comentários nas primeiras etapas do projeto, quando há maior quantidade de tempo e liberdade para fazer correções.

Visualizar as vulnerabilidades de segurança antecipadamente garante que as compilações sejam estáveis antes de passar para a próxima etapa de lançamento da funcionalidade no ambiente produtivo.

Além disso, medir a segurança pode ajudar a superar a resistência à adoção do *DevOps*, já que à medida que as vulnerabilidades diminuem percebe-se que o aumento da velocidade não é o que degrada a estabilidade do produto (HPE, 2016).

2.8.1.5 Mentalidade ágil

A promessa do *DevOps* é de quebrar as barreiras entre desenvolvimento e operações, mas o sucesso desse modelo dentro das organizações está relacionado à sua adoção por parte das pessoas envolvidas e principalmente a uma mudança da mentalidade.

Se todos trabalham juntos é mais fácil e rápido enfrentar problemas e conseqüentemente haverá mais eficiência, mas essa colaboração imprescindível que pode trazer benefícios tangíveis não acontecerá sozinha. É necessária uma boa relação entre desenvolvimento e operações, melhor e maior integração.

A intenção do *DevOps* é de unir entrega ágil, contínua e automação para ajudar as equipes de desenvolvimento e operação a serem mais eficientes, inovar e atender às demandas mais rapidamente e oferecer maior valor para empresas e clientes.

O *DevOps* é, portanto, uma filosofia a ser adotada e, sem a mentalidade correta, não surtirá efeito algum mesmo com as melhores ferramentas de tecnologia, mais recentes, e processos simplificados (ATLASSIAN, 2017a; ENGATES, 2014).

2.8.2 Boas práticas

Para que a aderência ao *DevOps* se torne realidade nas organizações, uma série de práticas focadas em sustentar e servir como a base para gerar confiança no processo de implantação desse modelo. A seguir, as práticas mais importantes que são observadas em *DevOps*.

2.8.2.1 Integração contínua

É um processo realizado pela equipe de desenvolvimento de *software* que consiste em integrar continuamente, em um único repositório, todo o desenvolvimento que foi realizado em um curto espaço de tempo, de forma a gerar uma fonte única.

Logo após esse processo, um teste integrado em todo o projeto é realizado com o propósito de consolidar todo o processo de integração realizado e entregue para o time responsável pelos testes de aceitação (SHARMA; COYNE, 2015).

Esse processo se tornou popular em conjunto com o movimento ágil, pois é bem diferente da forma tradicional de integração, que consiste em integrar todos os trabalhos desenvolvidos somente no final do projeto, para que o processo de testes seja realizado em todo o *software*. É possível que um problema demore para ser identificado, e mesmo que o retrabalho seja pequeno ele pode interferir em todo o desenvolvimento do projeto.

O principal foco da integração contínua é resolver esse problema de identificar de forma tardia problemas que impactam drasticamente o desenvolvimento do *software*.

Ao se integrar continuamente, há possibilidade de testar e validar se o que foi especificado realmente é o que está sendo entregue e, caso não esteja em conformidade, mudanças ocorrerão e serão mais fáceis de serem administradas, pois o possível problema foi identificado de forma prematura.

Para serem utilizadas, algumas práticas devem ser adotadas para que tragam o benefício desejado, dentre as quais podem-se citar (SHARMA; COYNE, 2015):

- utilizar ferramentas de controle de versão de software;
- utilizar compilação automatizada;
- integrar diariamente novas funcionalidades no repositório principal;
- utilizar um servidor de integração; e
- utilizar ferramentas para testes automatizados.

Conforme observado, a integração contínua é o princípio da implantação do *DevOps* e se tornou pré-requisito para o processo de entrega contínua.

2.8.2.2 Entrega contínua

É um processo que realiza a publicação de um *software* em seu ambiente produtivo com o menor risco possível de falhas e de forma automatizada e confiável. Ao contrário do processo comum de entrega, que consiste em entregar o projeto praticamente finalizado com um grande volume de funcionalidades, em que a experiência com o produto acontece somente no final de seu desenvolvimento e, como consequência, gera necessidade de alterações que podem demorar muito para serem concluídas.

O processo de entrega contínua consiste em dividir todo o projeto em pequenas partes possíveis de serem publicadas no ambiente produtivo, que são entregues em um ciclo de tempo bem menor do que o processo comum de entrega e garante que a experiência com o produto faça parte de seu desenvolvimento.

Essas práticas de entrega contínua reduzem o tempo e o risco associados à entrega de novas versões do *software* para os usuários e permitem, assim, o aumento e a colaboração entre desenvolvedores, testadores e o pessoal de operação responsável pela entrega (HUMBLE; FARLEY, 2010).

Em linhas gerais, entrega contínua é um conjunto de processos e diretrizes que devem ser seguidas para que o projeto de desenvolvimento de *software* seja realizado de forma que possa ser publicado no ambiente produtivo a qualquer momento.

□ Fowler (2013), para que esse processo de entrega contínua aconteça, a equipe de desenvolvimento deve garantir:

- uma estreita e colaborativa relação de trabalho entre todos os envolvidos na entrega;
- automatização extensiva de todas as partes possíveis do processo de entrega;
- garantir que o software possa ser utilizável durante todo seu ciclo de vida, mesmo que em pequenas partes e funcionalidades. A equipe deve manter o software utilizável ao longo do desenvolvimento de novos recursos;
- qualquer pessoa pode obter feedback rápido e automatizado sempre que alguém fizer alguma modificação ou criação de funcionalidades; e
- com o auxílio e utilização de ferramentas de controle de versão de software, é possível obter e implementar qualquer versão em qualquer ambiente.

2.8.2.3 Testes contínuos e automatizados

As práticas adotadas para a utilização do conceito *DevOps* demonstra que ao final de cada desenvolvimento uma entrega e integração devem ser realizadas.

Isso faz com que a cada entrega ocorra uma integração, e um conjunto de testes deve ser executado para garantir que essa integração das novas funcionalidades não afetou o que já estava pronto, testado e às vezes até mesmo sendo utilizado no ambiente produtivo.

Com ciclos de entregas cada vez mais curtos e velozes, é praticamente impossível a execução manual de todos os testes necessários para garantir que as funcionalidades estejam de acordo.

Testar é uma atividade multifuncional, que envolve todo o time e deve ser executada continuamente desde o início do projeto. O objetivo é fornecer uma boa abrangência do comportamento de modo a garantir que as funcionalidades sejam completamente implantadas e de forma correta (HUMBLE; FARLEY, 2010).

No universo *DevOps*, os responsáveis pelos testes contribuem com os desenvolvedores e usuários para que os testes possam ser mapeados, escritos e então automatizados desde o princípio do projeto, quando podem ser descritos em conjunto com os requisitos, antes mesmo de os desenvolvedores iniciarem qualquer tipo de trabalho.

Humble e Farley (2010) destacam os tipos de testes que são amplamente utilizados para garantir a entrega de um *software* ou serviço de alta qualidade, conforme apresentado no Quadro 8.

Quadro 8 – Esquema de diagrama de testes e seus tipos

		Voltado para Negócio			
		Automatizado		Manual	
Suporta o Time		Testes Funcionais de Aceitação Protótipos Simulações		Testes Exploratórios Testes de Usabilidade Testes de Aceitação do Usuário	Critica o Produto
		Testes Unitários Testes Integrados Testes de Sistema		Testes de Aceitação Não-Funcionais (Capacidade, Segurança)	
		Automatizado		Manual/Automatizado	
		Voltado para Tecnologia			

Fonte: adaptada de Humble e Farley (2010)

Nesse diagrama foram classificados os testes de acordo com a sua posição: se eles estão voltados para negócio ou voltados para tecnologia, e se eles suportam o time ou são usados para criticar o produto.

Os testes do quadrante voltados para o negócio e que suportam o time são mais conhecidos como testes funcionais ou de aceitação, garantindo que os critérios de aceitação de uma funcionalidade sejam atendidos.

Os testes de aceitação devem ser escritos e idealmente automatizados antes que o desenvolvimento se inicie. Quando isso ocorre, o time de desenvolvimento pode executar testes automatizados para verificar se um requisito foi finalizado com sucesso.

Nesse caso, não é preciso esperar a execução dos testes pela equipe de qualidade com a finalidade de reduzir a carga de trabalho dos testadores e também liberá-los para se concentrarem em testes exploratórios e atividades de maior valor em vez de tarefas repetitivas (HUMBLE; FARLEY, 2010).

Os testes do quadrante, voltado para tecnologia e que suportam o time, devem ser escritos, mantidos e automatizados exclusivamente pelo time de desenvolvimento. Três tipos de testes que se enquadram nessa categoria são: testes unitários, testes de componentes e testes de implantação.

Às vezes esses tipos de testes exigem recursos consideráveis, como ambientes especiais e conhecimento especializado para configuração e implantação, que podem demorar muito para serem executados, sejam eles automatizados ou não. Mesmo quando eles são totalmente automatizados, tendem a ser executados com menos frequência (HUMBLE; FARLEY, 2010).

Os testes do quadrante voltados para o negócio e que criticam o produto são executados de forma manual e verificam se o *software* ou serviço efetivamente entregará aos usuários o valor que eles esperam, verificando se as especificações escritas em funcionalidades estão corretas e conforme o esperado.

Os testes do quadrante voltados para a tecnologia e que criticam o produto são considerados testes não funcionais: devem ser testadas *performance*, capacidade, disponibilidade e a segurança do sistema.

As ferramentas e as técnicas utilizadas para desenvolver esses tipos de testes estão se tornando mais comuns e a criação de alguns testes básicos desse tipo devem ser mapeados e executados no início de qualquer projeto, por mais simples que ele seja (HUMBLE; FARLEY, 2010).

2.8.2.4 Infraestrutura como código

Para projetos de TI relacionados a *software*, um ambiente é representado por todos os recursos que ele necessita para operar, como, por exemplo, servidores, redes, sistemas operacionais, servidores *web*, servidores de banco de dados (inclui também suas configurações), e assim por diante.

Segundo Humble e Farley (2010)

Infraestrutura representa todos os ambientes da sua organização, juntamente com os serviços que os suportam, como servidores DNS, firewalls, roteadores, repositórios de controle de versão, armazenamento, aplicativos de monitoramento, servidores de e-mail e assim por diante (HUMBLE; FARLEY, 2010, p. 277, tradução nossa).

A crescente aderência dos projetos de TI, envolvidos com *softwares*, aos modelos ágeis de gerenciamento fez com que aumentasse a necessidade de acelerar e dinamizar a configuração de novos ambientes.

No processo de entrega contínua de projetos ágeis de *softwares*, os testes necessitam ser executados em ambientes muito próximos e similares ao ambiente produtivo, para que possíveis falhas possam ser mapeadas o mais cedo possível, reduzindo os impactos negativos no ambiente produtivo.

Com isso, observa-se uma nova forma de se lidar com a infraestrutura, que começa a ser chamada de código, utilizando técnicas, modelos e ferramentas que já são utilizadas para projetos de desenvolvimento de *softwares* como resposta a essa demanda dos métodos ágeis.

Um dos princípios de *DevOps* é investir em automação para permitir a execução de tarefas de forma mais rápida, para minimizar a possibilidade de falha humana e se tornar um processo mais confiável. Com o aumento da cultura *DevOps* e o crescimento da colaboração entre equipes de operações e desenvolvedores, diversas ferramentas surgiram e têm evoluído para garantir a padronização e automação da gestão de infraestrutura.

Essas ferramentas permitem que a infraestrutura seja administrada da mesma maneira que os desenvolvedores de *softwares* tratam o código fonte do programa, pelo controle de versões, realização de testes automatizados, distribuição de módulos comuns e também execução de mudanças de configuração nos servidores (SATO, 2014).

Os padrões que podem ser adotados para se implantar um processo de gerenciamento da infraestrutura como código para manter todo o processo de entrega confiável podem ser visualizados no Quadro 9.

Quadro 9 – Infraestrutura como código: padrões

Infraestrutura como Código: padrões	
Provisionamento Automático	Automação do processo de configuração dos ambientes para incluir redes, serviços externos e infra-estrutura para evitar que as instâncias configuradas sejam obras de arte, fazendo com que os membros da equipe executem etapas parcialmente ou totalmente manuais para provisioná-las.
Monitoramento Comportamental	Automação dos testes para verificar o comportamento da infraestrutura. Execução contínua destes testes podem fornecer alertas em tempo quase real, pois geralmente não existem alertas ou monitoramento em tempo real e a configuração do sistema é escrita sem testes.
Sistemas Imunes	Implantação de uma instância de <i>software</i> por vez ao realizar a Monitoração Orientada por Comportamento. Se for detectado um erro durante a implantação incremental, uma liberação de reversão é iniciada para reverter alterações pois comumente as implantações não são incrementais e sem monitoramento.
Ambientes Bloqueados	Bloqueio dos ambientes compartilhados de uso externo e interno não autorizado, incluindo a equipe de operações. Todas as alterações necessárias devem ser versionadas e aplicadas através de automação, não permitindo que qualquer usuário autorizado possa acessar ambientes compartilhados e aplicar alterações manuais de configuração, colocando o ambiente em um estado desconhecido levando a erros de implantação.
Ambientes Semelhantes ao Produtivo	Ambientes de desenvolvimento e testes devem ser tão semelhantes à produção quanto possível. Os ambientes não devem ser atualizados apenas semanas ou dias antes de um lançamento, e esta atualização não pode ser configurada e controlada manualmente.
Ambientes Transitórios	Os ambientes devem ser capazes de serem criados, gerenciados e finalizados facilmente.

Fonte: adaptado de Cooper (2017)

3 MÉTODO DE PESQUISA

Nesta seção, os procedimentos metodológicos utilizados para a execução da coleta e análise de dados são apresentados.

3.1 Natureza da pesquisa

O método utilizado para a elaboração desta pesquisa é o quantitativo, que faz uso de procedimentos e instrumentos estatísticos para possibilitar análise de dados menos suscetíveis a distorções, levando a interpretações mais sólidas e à aquisição de resultados com maior margem de credibilidade (RICHARDSON; *et al.*, 1999; COOPER; SCHINDLER, 2003).

Esta pesquisa tem caráter exploratório e descritivo. A natureza exploratória pode ser explanada pela busca de conhecimento com maior profundidade em um tema do qual o pouco se sabe. O caráter descritivo destina-se à necessidade de identificar possíveis relações entre variáveis, estudando as características de uma determinada população (SELLTIZ; *et al.*, 1975; HAIR; *et al.*, 2014).

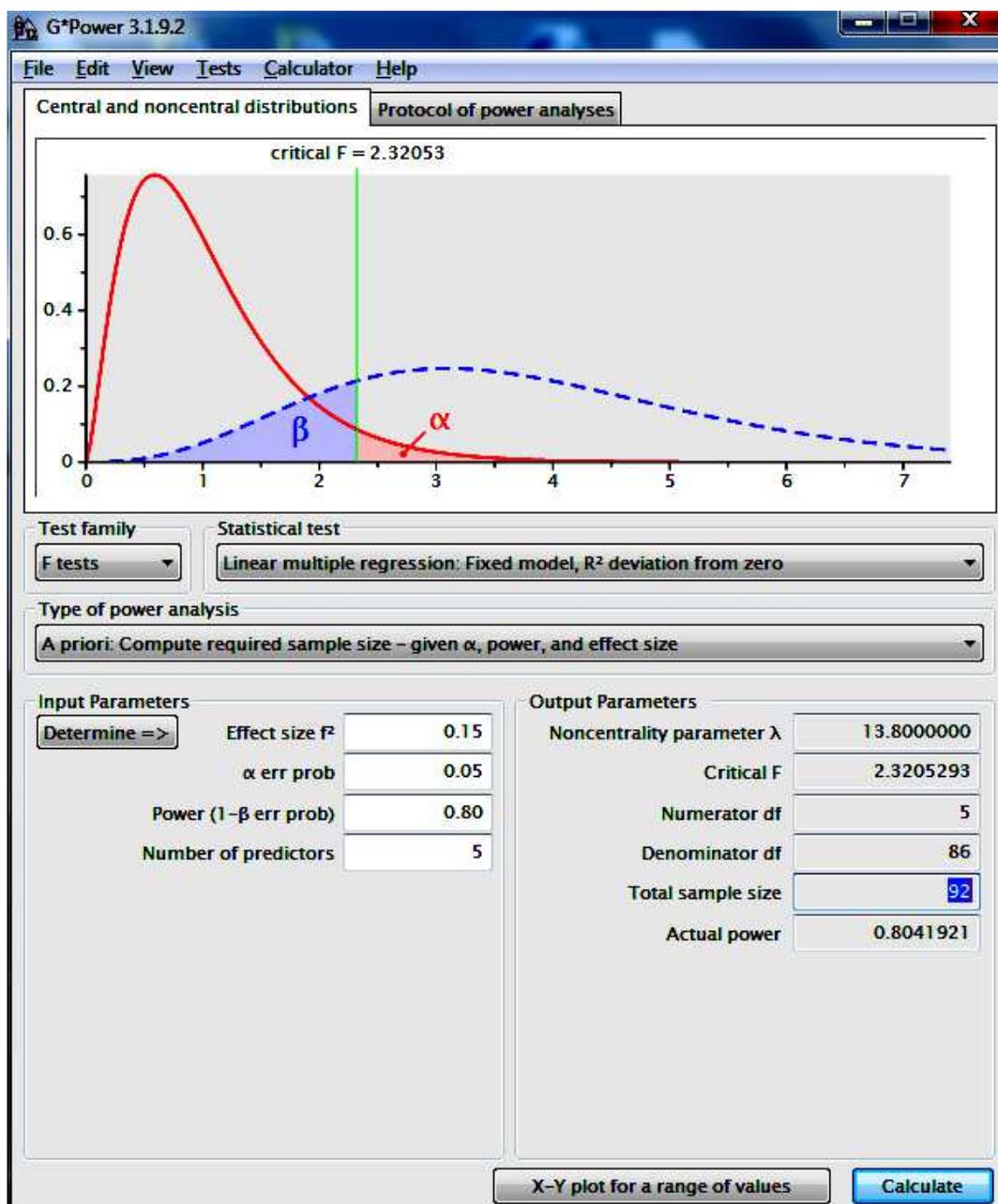
3.2 População e amostra

De acordo com Levin; *et al.* (2012), o grupo de elementos que tem características em comum é denominado universo de dados ou população. Para esta *survey* o universo é composto por cerca de 3.000 profissionais de tecnologia da informação que trabalham em desenvolvimento de *software* e serviços de TI, em um universo de 125 empresas de TI da Região Metropolitana do Vale do Paraíba e Litoral Norte (RMVALE).

Foi seguida a estimativa do poder do teste indicada por Cohen (1988) e utilizado o *software* G*Power versão 3.1.9.2 (<http://www.gpower.hhu.de/en.html>) (FAUL *et al.*, 2009). Para cálculo da amostra mínima utilizou-se um tamanho de efeito de 0,15 (médio) e poder do teste de 0,80.

O número da amostra mínima obtido com o cálculo do *software* foi de 92 questionários, conforme demonstra a Figura 10.

Figura 10 – G*Power Software – Cálculo da Amostra Mínima



Fonte: Software G*Power (2017)

Após a execução da pesquisa foram obtidas 108 respostas ao questionário válidas. Essa amostra de 108 respostas válidas é apropriada à finalidade desta pesquisa.

3.3 Instrumento e procedimentos de aquisição dos dados

O instrumento utilizado para realizar a aquisição dos dados foi um questionário fechado, com 23 questões voltadas à avaliação das habilidades e capacidades necessárias para os profissionais de TI alcançarem sucesso quanto à adoção do modelo *DevOps* nas organizações.

As pesquisas realizadas na revisão de literatura apontaram as cinco principais dimensões que compõem o *DevOps* e que permitiram mensurar as principais habilidades e capacidades responsáveis pelo sucesso na adoção desse modelo.

As cinco principais dimensões do *DevOps* a serem avaliadas e abordadas por esta pesquisa são (HPE, 2016; PUPPET, 2016b; ATLISSIAN, 2017b; FORESTER, 2016; PEREIRA, 2017):

- **Automação/Velocidade:** o *DevOps* foi criado como uma forma dos desenvolvedores responderem rapidamente às demandas de mudanças das organizações garantindo entregas com conteúdo cada vez mais rápido e sempre melhor. A velocidade não só oferece vantagem competitiva, mas também melhora a forma como as organizações trabalham com TI. Tecnicamente, o *DevOps* implica automatizar tanto quanto possível para acelerar processos e reduzir erros humanos.
- **Qualidade:** para aumentar a velocidade, a entrega de software em pequenas unidades, que significam entrega mais rápida, com melhor qualidade e risco reduzido, e cria qualidade em toda a cadeia de entrega de software com uma cultura de excelência.
- **Produtividade:** o principal propósito do *DevOps* é auxiliar a área de TI das organizações para garantir a entrega de maior valor para o negócio, alinhando todos os esforços com a sua necessidade real. Utilizar ferramentas e quadros visuais para o controle e gerenciamento das atividades é uma forma de manter o foco no valor sem desperdício, pois garante que as atividades a serem entregues estão alinhadas com a estratégia de negócio da organização, gerando maior valor agregado e melhoria da produtividade.

- **Segurança:** as equipes devem garantir que não se introduzam vulnerabilidades ao fazer alterações nos softwares e serviços. Devem realizar os testes de segurança no início do ciclo de desenvolvimento para enxergar vulnerabilidades antecipadamente; e
- **Mentalidade ágil:** o DevOps surgiu para auxiliar as equipes de desenvolvimento e operações a serem mais eficientes, inovar e atender às demandas mais rapidamente e oferecer maior valor para as organizações. O DevOps enfatiza a responsabilidade compartilhada e busca a qualidade dos desenvolvedores, testadores, operacionais e membros da equipe de segurança.

Com base no contexto apresentado criou-se uma estrutura de questões e parâmetros (variáveis latentes) e respectivos indicadores (variáveis mensuráveis), com a devida categorização dentro de sua dimensão. As respostas devem variar entre zero e dez, da seguinte forma: 0 = Discordo totalmente da afirmativa e 10 = Concordo totalmente com a afirmativa.

Os pontos importantes que os participantes deveriam considerar ao responder eram três:

- Considerar a experiência acumulada ao longo da carreira ao responder cada pergunta. NÃO considerar apenas sua posição ou emprego atual;
- Considerar o que realmente vivenciou nas organizações onde trabalhou. NÃO responder com a pontuação que considera ideal; e
- NÃO considerar sua melhor ou pior experiência para responder uma questão, mas a experiência geral.

A primeira versão do questionário foi enviada para um profissional sênior de TI, para que seu formato fosse pré-validado e então verificado se houve o entendimento das questões.

Após isso, os ajustes apontados como necessários foram executados de forma a melhorar o instrumento para que então fosse possível iniciar a coleta dos dados. O instrumento final usado neste trabalho foi estruturado conforme demonstrado no Quadro 10.

Quadro 10 – Parâmetros e indicadores do modelo

Dimensão	Capacidades e Habilidades	Código	Indicador	Autores
Automação/Velocidade (Aut)	Processo de implantação automatizado	Aut_1	Frequência de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação.	PUPPET (2016) ATLASSIAN (2017)
		Aut_2	Tempo médio de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação	PUPPET (2016)
	Automação de processos/eficácia no controle	Aut_3	Utilização de processos de rollback automatizados quando problemas/bugs são encontrados nas atualizações executadas em produção.	FORRESTER (2016)
		Aut_4	Utilização de repositórios controlados para auxiliar o controle de versão dos códigos fontes/configuração de ambientes/infra.	PUPPET (2016) PEREIRA (2017)
	Automação de testes/garantia da qualidade	Aut_5	Utilização de ferramentas para testes automatizados durante a validação da maioria das alterações nas aplicações/serviços.	FORRESTER (2016)
		Aut_6	Tempo médio de validação de uma alteração de código ou aplicação de uma correção em um serviço.	HPE (2016)
		Aut_7	Ambiente de teste/validação atualizado em relação ao produtivo	PUPPET (2016) PEREIRA (2017) ATLASSIAN (2017)
Qualidade (QA)	Confiabilidade no processos de entrega	QA_1	Adoção de processos de liberação em produção executado de forma regular e com controle dos riscos.	FORRESTER (2016)
	Grande cobertura para levantamento de requisitos e cenários de testes	QA_2	Baixo número de incidentes e defeitos relatados em um produto ou serviço após implantação para o ambiente produtivo	PUPPET (2016)
		QA_3	Prática de reuniões de retrospectiva pós implantação/lições aprendidas para assegurar qualidade do processo de entrega.	PUPPET (2016) PEREIRA (2017)
	Engajamento dos stakeholders	QA_4	Envolvimento e participação adequada dos <i>Stakeholders</i> dentro do ciclo de vida do projeto.	FORRESTER (2016)
Produtividade (Prod)	Tempo de restauração de um serviço/aplicativo quando ocorre uma falha	Prod_1	Correção dos erros encontrados em ambiente produtivo após implantação de alguma nova aplicação ou funcionalidade dentro do tempo acordado.	PUPPET (2016)
	Alinhamento com o Negócio	Prod_2	As metas da equipe são claras e estão relacionadas com os objetivos de negócios da organização.	PUPPET (2016) PEREIRA (2017)
	Medição de produtividade de forma visual	Prod_3	O uso de painéis visuais com estatísticas de desempenho (entregas e atividades realizadas, à realizar, etc..) disponível para todos os integrantes da equipe favorece a gestão do trabalho e melhora a produtividade.	PEREIRA (2017)
	Foco no valor e alinhamento dos requisitos	Prod_4	Usualmente as equipes entregam correções e novas funcionalidades nos serviços/aplicações que realmente são importantes para o negócio.	HPE (2016)
Segurança (Seg)	Desenvolver de forma correta, sem vulnerabilidades	Seg_1	Realização de testes de segurança e vulnerabilidade antes da promoção para a produção.	HPE (2016)
	Envolvimento do time de segurança	Seg_2	Envolvimento do time de segurança dentro do ciclo de vida do projeto/serviço.	PUPPET (2016)
	Política de segurança/compliance	Seg_3	Políticas e processos de segurança e conformidade para desenvolvimento e implantações de projetos e/ou serviços conhecidas e expostas de forma clara e coesa.	PUPPET (2016)
Mentalidade Ágil (MA)	Utilização da metodologia ágil	MA_1	Adoção do uso de métodos ágeis no ciclo de vida do projeto/serviço.	PEREIRA (2017)
	Liberdade para agregar valor;	MA_2	As pessoas são incentivadas a questionar umas as outras para encontrar as melhores soluções possíveis.	PEREIRA (2017)
	Habilidade de se atualizar de forma contínua	MA_3	Aprendizagem contínua.	PEREIRA (2017)
	Habilidade em ser flexível as mudanças	MA_4	As mudanças internas ou externas que impactam no negócio são analisadas e geram respostas rápidas por parte da equipe.	HPE (2016)
	Proatividade	MA_5	Iniciativa para realizar as tarefas ou as assumem ao invés de esperar pela atribuição das tarefas por um líder.	PEREIRA (2017)

O Quadro 10 apresenta, respectivamente, as dimensões AUT, QA, PROD, SEG e MA, capacidades e habilidades (variáveis latentes) com seus respectivos indicadores que foram usados para compor este modelo de pesquisa. O instrumento final utilizado e aplicado está exemplificado no Apêndice A deste estudo.

Esta pesquisa foi submetida à aprovação do Comitê de Ética em Pesquisa, pela Plataforma Brasil (<http://plataformabrasil.saude.gov.br/login.jsf>). O parecer com a aprovação e liberação do uso da pesquisa pode ser verificado no Anexo A deste trabalho.

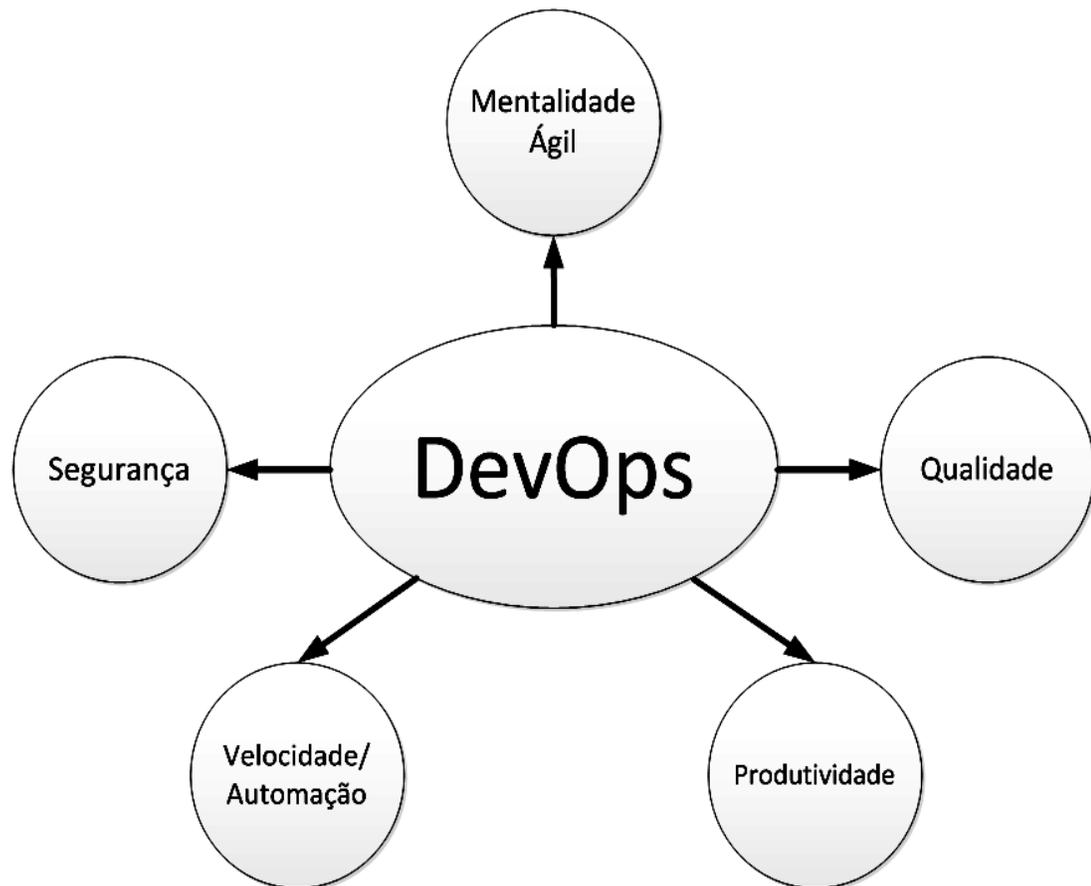
Os resultados para validação deste modelo foram adquiridos a partir de uma pesquisa direcionada aos profissionais de TI. O questionário foi divulgado entre os meses de agosto e setembro de 2017, com o uso do *Google Forms* como ferramenta para a coleta das respostas. Os questionários foram enviados por *e-mail*, utilizando o endereço do *Google Forms* criado para este propósito.

Na primeira semana de divulgação do questionário foram obtidas em torno de 60 respostas válidas. Após a quarta semana a marca de 108 respostas válidas foi atingida e a disponibilidade do endereço de resposta ao questionário *on-line* foi desativada, pois ultrapassou a quantidade mínima de 92 respostas estimadas, conforme já citado na subseção população desta mesma seção.

O crescimento da internet tem permitido a utilização de *e-mails* e ferramentas como esta disponibilizada gratuitamente pelo *Google*, para divulgação da pesquisa e maior velocidade para coleta de um grande volume de dados, reduzindo o tempo de análise das respostas válidas, codificação, armazenamento e exportação dos dados para avaliação das respostas (FLEMING; BOWDEN, 2009; OLSEN, 2009).

3.4 Modelo estrutural

O *DevOps* é composto por cinco dimensões que compõem sua estrutura. A partir da revisão de literatura elaborou-se um modelo estrutural de pesquisa para este trabalho, mostrado na Figura 11.

Figura 11 – Modelo Estrutural da Pesquisa *DevOps*

Com este modelo buscam-se relacionar e analisar as principais habilidades e competências necessárias para desenvolver e aplicar os principais conceitos do modelo *DevOps* para que o profissional de TI possa adotar essa nova tendência.

3.5 Análise e interpretação dos dados

Métodos de análise multivariadas foram utilizados para a produção deste trabalho, por serem consideradas mais adequadas para a caracterização das complexas aqui relações exploradas. A modelagem de equações estruturais (MEE) baseada em *partial least squares* (mínimos quadrados parciais – (PLS) foi identificada como a técnica mais correta a ser usada (HAIR; *et al.*, 2014; RECHZIEGEL, 2014; RINGLE; SILVA; BIDO, 2014).

O uso da MEEPLS para pesquisas na área de sistemas de informação é recomendado por Zwicker, Souza e Bido (2008), pois essa técnica se revela como mais adequada para trabalhos com foco exploratório, por abordar temas sobre o qual ainda há pouco conhecimento.

O modelo MEEPLS ainda é indicado para pesquisas em que os dados não apresentam distribuição normal e são usadas escalas ordinais. É recomendada para modelos com suporte teórico pouco explorado, como ocorre com esta pesquisa (HAIR; *et al.*, 2014; RINGLE, SILVA e BIDO, 2014).

De acordo com as diretrizes de Ringle, Silva e Bido (2014), os indicadores foram denominados de acordo com o nome de seus construtos e utilizaram-se os códigos mostrados no Quadro 11.

Não foi excluído nenhum questionário respondido da amostra, pois todos apresentavam os parâmetros necessários para a análise. Os resultados foram organizados em uma planilha *Excel* e armazenados no formato CSV (*Comma Separated Value*), para que fosse possível executar sua importação no *software* SmartPLS 2.0 M3.

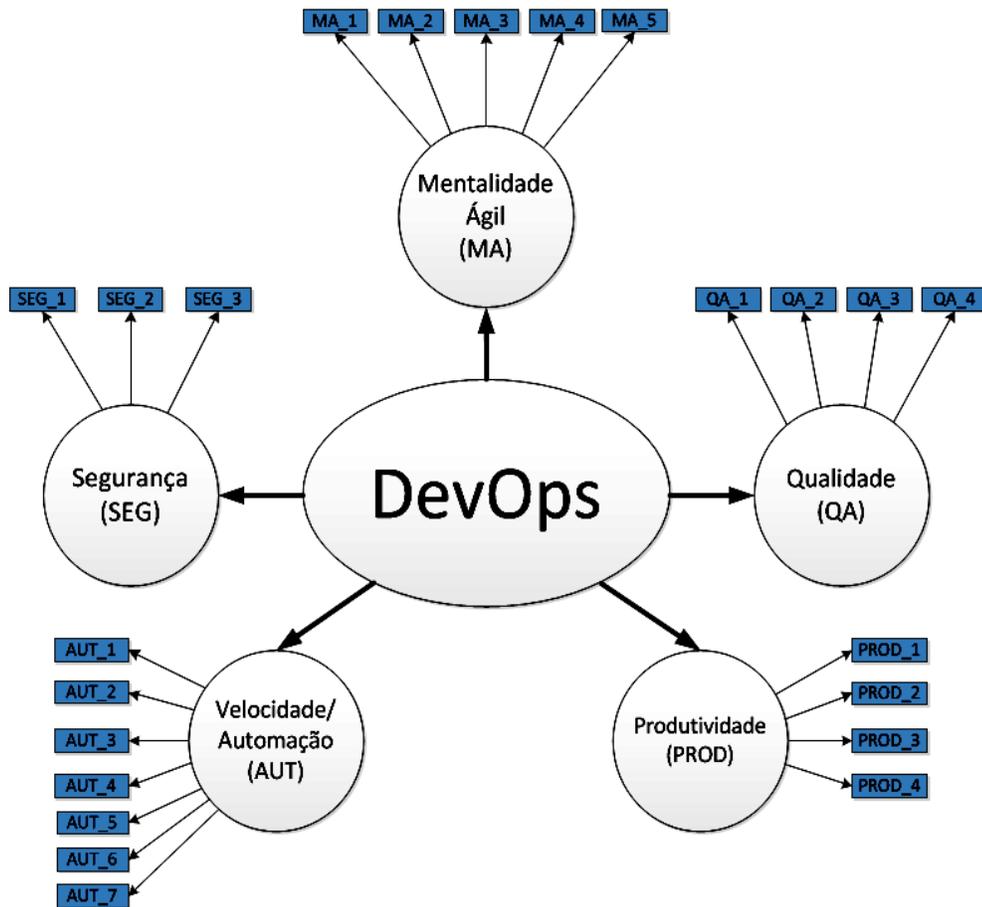
Após os dados importados e validados no SmartPLS foi construído, então, o modelo de mensuração e se utilizou a notação padronizada para a MEE e a relação entre os indicadores e seus respectivos construtos.

Segundo Byrne (2010), o modelo de mensuração define as relações existentes entre variáveis observadas e variáveis não observadas pela relação entre a pontuação do instrumento de medição e os construtos que a pesquisa pretende medir.

A relação entre um conjunto de variáveis com uma variável latente esclarece se o modelo de mensuração é formativo ou reflexivo. Nos modelos reflexivos, as variáveis latentes refletem-se nos indicadores e em termos de representação visual. A direção das setas é da variável latente para as variáveis mensuráveis (MARÔCO, 2010; 2012; HAIR *et al.*, 2014).

O modelo de mensuração desta pesquisa está na Figura 12.

Figura 8 – Modelo de mensuração da pesquisa



No modelo estrutural, os círculos e/ou elipses equivalem aos construtos ou variáveis latentes, enquanto que os retângulos representam os indicadores ou variáveis mensuráveis (TABACHNICK; FIDELL, 2007; BYRNE, 2010).

O modelo reflexivo é notado nas relações entre o construto *DevOps* e seus parâmetros mentalidade ágil, segurança, velocidade/automação, produtividade e qualidade e entre os parâmetros e seus respectivos indicadores.

O objetivo da etapa de avaliação do modelo é validar o quanto o modelo teórico consegue reproduzir a estrutura de correlação dos indicadores observados na amostra da pesquisa (MARÔCO, 2010).

Os procedimentos utilizados para validar de forma correta o modelo reflexivo e estrutural que foram construídos nesta pesquisa serão descritos a seguir e a análise e interpretação dos dados seguirão os procedimentos indicados por Ringle, Silva e Bido (2014) e Hair; *et. al* (2014).

3.5.1 Avaliação do modelo de mensuração reflexivo

3.5.1.1 Confiabilidade dos indicadores

A confiabilidade de um indicador aponta o quanto de sua variância pode ser explanada pela sua variável latente. Um critério usado é que mais de 50% da variância de um indicador seja explicado pelo seu construto.

Nessa avaliação, Hair; *et al.* (2011) e Henseler; *et al.* (2009) recomendam que os valores das cargas dos indicadores de 0,6 a 0,7 são satisfatórios nas pesquisas exploratórias, enquanto pesquisas em estágios mais avançados devem ter valores de 0,7 a 0,9.

Os indicadores com valores entre 0,4 e 0,7 devem ser removidos se aumentam a confiabilidade composta e não afetam os critérios de validade discriminante e de validade convergente do modelo (GOTZ; LIEHR-GOBBERS; KRAFFT, 2010; HAIR; *et al.*, 2014).

3.5.1.2 Consistência interna (confiabilidade composta)

A consistência interna deve variar entre zero e um e a recomendação verificada na literatura sugere resultados das cargas dos indicadores entre 0,6 a 0,7 como apropriadas para pesquisas exploratórias, enquanto pesquisas em fases mais avançadas necessitam manifestar resultados entre 0,7 e 0,9 (HENSELER *et al.*, 2009; HAIR; *et al.*, 2014).

3.5.1.3 Validade convergente

A validade convergente (VC) revela a tendência de um conjunto de indicadores representar o mesmo construto subjacente. Uma forma comum de averiguar a validade convergente de um construto é a *Average Variance Extracted* (AVE), ou variância média extraída, que estabelece uma medida da quantidade de variância nos indicadores que é explanada pelo construto.

O valor mínimo recomendado pelos autores é de 0,5 para que a AVE assegure a presença de validade convergente. Isso quer dizer que uma variável não mensurável consegue explicar mais de 50% da variação dos seus indicadores (HENSELER; *et al.*,2009; HAIR; *et al.*, 2014).

3.5.1.4 Validade discriminante

Segundo Gotz, Liehr-Gobbers e Krafft (2010), a validade discriminante corresponde à dissimilaridade na medição de um instrumento de diferentes construtos, ou seja, cada constructo representa uma dimensão separada das outras. Dois critérios devem ser observados para garantir a validade discriminante do modelo de pesquisa deste trabalho (CHIN, 2010; HAIR, *et al.*, 2014):

- **Cross loadings ou cargas cruzadas:** a carga de um indicador deve ser maior para seu respectivo construto que a carga relacionada a qualquer outro construto do modelo; e
- **Fornel-Lacker:** critério pelo qual a raiz quadrada dos valores das AVEs é comparada com correlações das variáveis latentes. A raiz quadrada do AVE de cada construto deve ser superior à sua maior correlação com qualquer outro construto. Cada variável latente tem mais variância com seu próprio bloco de indicadores do que com outra variável latente que representa um bloco diferente de indicadores.

3.5.2 Avaliação do modelo estrutural

3.5.2.1 Coeficiente de Pearson (R^2)

O coeficiente de determinação de Pearson (R^2) é uma medida indicadora da qualidade do modelo ajustado, mostra a porcentagem de variância da variável dependente que é explicada pela variável independente (latente).

O R^2 mede a amostra da variância de um construto interno que é explicada pelo modelo estrutural (COHEN, 1988; CHIN, 2010; HAIR; *et al.*, 2014). Foram identificados na literatura diferentes valores de referência para o coeficiente de determinação, conforme mostra o Quadro 11.

Quadro 11 – Parâmetros de referência r^2

Autores	Valor do R^2			Área de aplicação
	Fraco	Moderado	Substancial	
Cohen (1998)	2%	13%	26%	Ciências sociais e do comportamento
Henseler <i>et al.</i> (2009)	19%	33%	67%	Não especificada
Hair <i>et al.</i> (2014)	25%	50%	75%	Não especificada

Fonte: Fraga (2015)

Por tratar-se de um trabalho voltado para a área de ciências sociais aplicadas, os valores de referência citados por Cohen (1998) serão utilizados para validar esta pesquisa, critério também adotado por Ringle, Silva e Bido (2014).

3.5.2.2 Tamanho do efeito (f^2)

O tamanho do efeito é obtido pela inclusão e exclusão de construtos do modelo. A alteração do coeficiente de determinação da variável dependente é calculada pela estimação dupla do modelo estrutural, ou seja, uma vez com a variável latente independente e uma vez sem essa variável.

Os valores de 0,02, 0,15, ou 0,35 apontam fraca, moderada ou substancial efeitos da variável latente externa em determinada variável latente interna (COHEN, 1988; HAIR; *et al.*, 2014).

3.5.2.3 Relevância preditiva (Q^2) ou indicador de Stoner-Geisser

De acordo com Hair; *et al.* (2014), o indicador de *Stoner-Geisser* (Q^2) possibilita qualificar a exatidão do modelo ajustado. Ringle, Silva e Bido (2014) confirmam que o resultado de Q^2 igual a um mostra que o modelo demonstra a realidade sem erros. Para validação desta pesquisa, o resultado de Q^2 deve ser maior que zero.

3.5.2.4 Goodness of Fit (GoF)

Goodness of Fit (GoF), indica a adequação ou ajuste geral do modelo calculado pela média geométrica entre o R^2 médio (adequação do modelo estrutural) e a média ponderada das AVEs (adequação do modelo de mensuração) (TENENHAUS; *et al.*, 2005).

Ao propor esse índice, Tenenhaus *et al.* (2005) não recomendaram um valor de corte, porém, Wetzels *et al.* (2009) indicam o valor de 0,36 como apropriado para as áreas de ciências sociais e do comportamento, e é adotado neste trabalho.

3.5.2.5 Análise dos valores e significância dos Coeficientes de Caminho (Γ)

Os coeficientes de caminho apontam o relacionamento entre um construto com o outro construto, com valores entre -1 e +1, em que valores próximos de +1 indicam uma forte relação positiva entre dois construtos, enquanto que valores próximos a -1 indicam o contrário. Valores próximos de zero demonstram relações fracas (RINGLE, SILVA, BIDO, 2014; HAIR *et al.*, 2014).

Para que um Γ seja válido utiliza-se o teste t de Student (estatística t). Para validar se a relação causal entre dois constructos é significativa ou não verificam-se os valores do teste utilizados para determinar o nível de significância dos coeficientes de caminho.

Para identificar a existência da relação entre os constructos, os resultados dos coeficientes devem ser acima de 1,96 para significância de 5% ou 0,05, conforme utilizado neste trabalho (HAIR *et al.*, 2011).

Para calcular os valores de t e validar a significância dos coeficientes de caminho foi executado o algoritmo de *bootstrapping* do *software SmartPLS*, que é um processo de reamostragem que cria um grande número de amostras aleatórias, geradas a partir da amostra original e visa examinar a estabilidade das estimativas. Para esta pesquisa foi utilizado o parâmetro de 1.000 reamostragens, conforme procedimento recomendado por Ringle, Silva e Bido (2014).

Na próxima seção são apresentados e discutidos os resultados da validação do modelo de pesquisa.

4 DISCUSSÃO DOS RESULTADOS

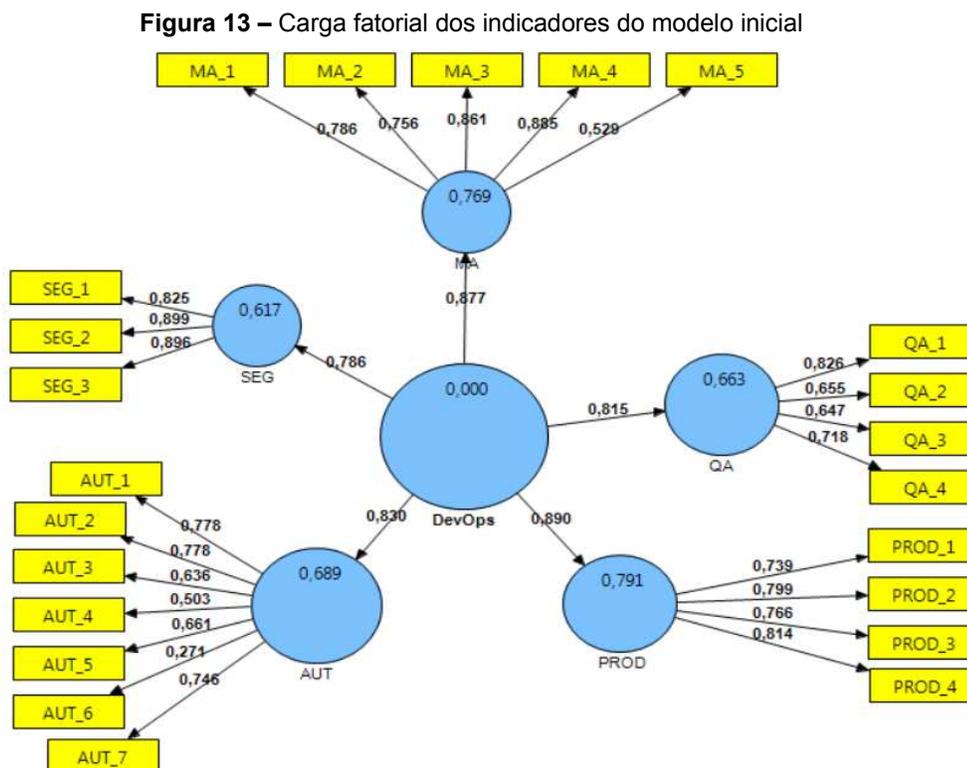
Nesta seção, os resultados da avaliação do modelo de pesquisa são apresentados e analisados de acordo com os procedimentos já citados na seção três deste trabalho. Em seguida, eles são discutidos com base na literatura.

4.1 Apresentação e interpretação dos resultados

4.1.1 Avaliação do modelo de mensuração

O modelo proposto para a pesquisa foi construído no *software SmartPLS 2.0 M3*, com criação dos construtos ou variáveis latentes (VLs) representadas por círculos azuis. Em seguida, os indicadores ou variáveis observadas (VOs), representadas por retângulos amarelos foram associadas a cada construto.

A Figura 13 mostra os coeficientes obtidos na primeira aplicação da MEEPLS sobre o modelo.



Fonte: SmartPLS 2.0 M3 Software

4.1.1.1 Avaliação do modelo de mensuração reflexivo

A primeira etapa para validação do modelo reflexivo consiste em executar a apuração do valor individual das cargas padronizadas dos indicadores existentes no modelo, conforme demonstra a Tabela 3.

Tabela 3 – Carga fatorial dos coeficientes no modelo inicial

Indicadores	Construtos				
	AUT	MA	PROD	QA	SEG
AUT_1	0,77801				
AUT_2	0,77819				
AUT_3	0,63567				
AUT_4	0,50284				
AUT_5	0,66103				
AUT_6	0,27068				
AUT_7	0,74642				
MA_1		0,78628			
MA_2		0,7562			
MA_3		0,86117			
MA_4		0,88511			
MA_5		0,52936			
PROD_1			0,7389		
PROD_2			0,7989		
PROD_3			0,7657		
PROD_4			0,814		
QA_1				0,8263	
QA_2				0,6548	
QA_3				0,6468	
QA_4				0,7181	
SEG_1					0,8248
SEG_2					0,8994
SEG_3					0,8962

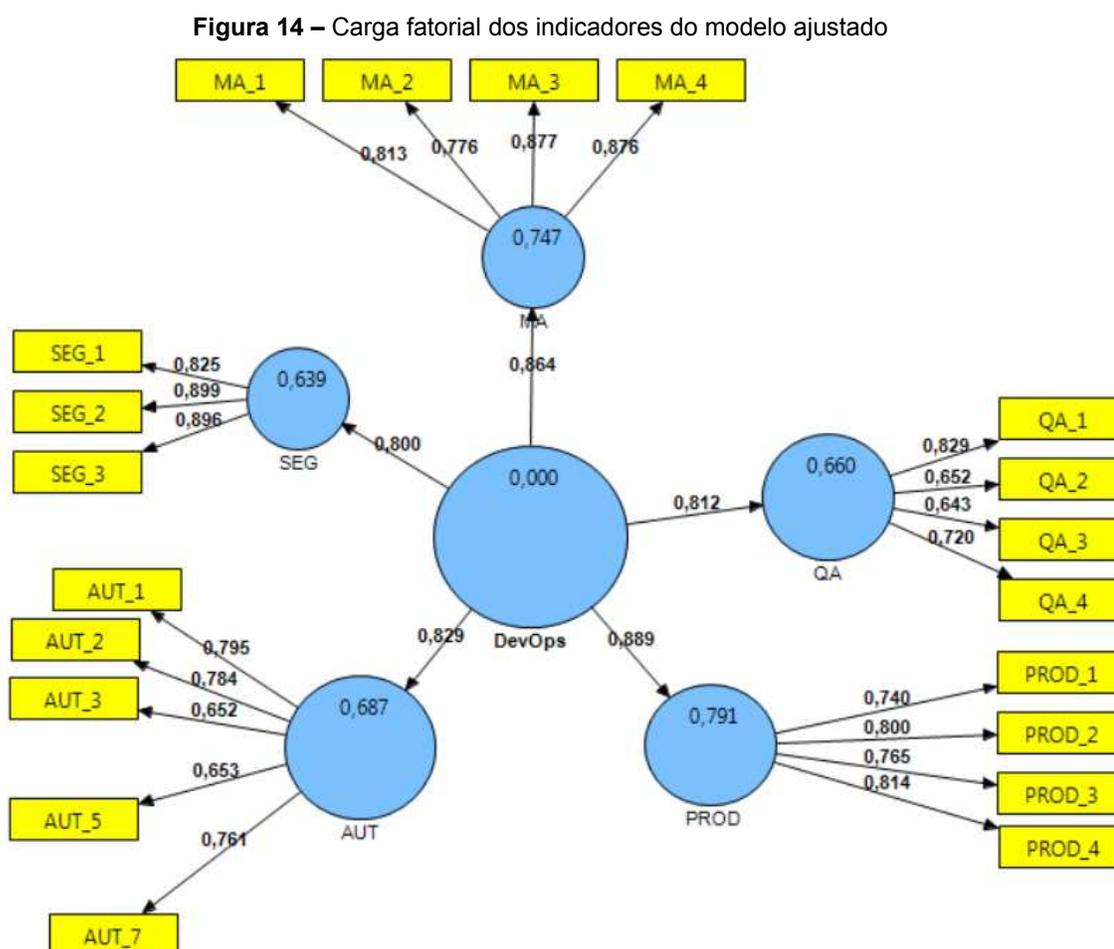
Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

O valor da carga padronizada de cada indicador deve ser superior ou igual a 0,708, para satisfazer o critério de confiabilidade. Observa-se que alguns indicadores têm carga inferior a 0,600, que não satisfazem o critério de confiabilidade, logo, comprometem a consistência interna.

Os indicadores (AUT_4, AUT_6 e MA_5), com as cargas abaixo do especificado, foram removidos. Depois o modelo foi novamente estimado para verificar seu ajuste e aumentou a sua confiabilidade composta.

Nos modelos reflexivos os indicadores dividem temas comuns relativos à variável latente. Por isso, a inclusão ou eliminação de indicadores não compromete a validade do constructo, sendo inclusive necessária para o ajuste do modelo (HAIR *et al.*, 2011;).

Procedeu-se novamente a execução do algoritmo PLS e, em seguida, iniciou-se a avaliação do segundo modelo. A Figura 14 mostra os coeficientes obtidos na aplicação da MEEPLS no modelo já ajustado.



Fonte: SmartPLS 2.0 M3 Software

Após o ajuste do modelo, todos os indicadores atenderam ao critério de confiabilidade, apresentando uma carga padronizada entre 0,6 a 0,7 e até mesmo cargas superiores a 0,7, em que o ajuste realizado se demonstrou satisfatório.

Os valores individuais das cargas padronizadas de todos os indicadores após ajuste são exibidos na Tabela 4.

Tabela 4 – Carga dos coeficientes no modelo ajustado

Indicadores	Construtos				
	AUT	MA	PROD	QA	SEG
AUT_1	0,79496				
AUT_2	0,784023				
AUT_3	0,652337				
AUT_5	0,653474				
AUT_7	0,761309				
MA_1		0,8128			
MA_2		0,7758			
MA_3		0,877			
MA_4		0,8756			
PROD_1			0,7395		
PROD_2			0,7996		
PROD_3			0,7647		
PROD_4			0,8138		
QA_1				0,8294	
QA_2				0,6521	
QA_3				0,6426	
QA_4				0,7202	
SEG_1					0,8254
SEG_2					0,8991
SEG_3					0,8959

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

A segunda etapa para validação do modelo é a conferência de sua confiabilidade composta (CC), um dos critérios e qualidade do modelo mostrados na Tabela 5, juntamente com o Alfa de Cronbach (AC).

Tabela 5 – Confiabilidade composta e Alfa de Cronbach do modelo ajustado

Construto	Confiabilidade composta (CC)	Alfa de Cronbach (AC)
AUT	0,851351	0,781975
MA	0,90282	0,856583
PROD	0,861178	0,785542
QA	0,805354	0,68328
SEG	0,906578	0,845587

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

Ainda que a CC seja mais adequada ao PLS-PM, pode-se usar tanto o AC como a CC para verificar se não há viés na amostra (RINGLE; SILVA; BIDO, 2014). Os valores considerados satisfatórios estão entre 0,70 e 0,90, o que garante ao modelo a confiabilidade composta.

A terceira etapa para validação do modelo é a averiguação de sua validade convergente, que pode ser medida pelas AVEs de seus construtos e se considera valor de referência superior a 0,50 (HENSELER *et al.*, 2009). Conforme mostra a Tabela 6, os valores das AVEs dos construtos do modelo variaram entre 0,53576 e 0,76413. Dessa forma, constata-se que o modelo apresenta validade convergente.

Tabela 6 – Validade convergente do modelo ajustado

Construto	AVE
AUT	0,53576
MA	0,69958
PROD	0,60831
QA	0,51119
SEG	0,76413

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

A quarta etapa para validação do modelo é a sua validade discriminante, que deve atender a dois critérios. O primeiro é o das cargas cruzadas, em que as cargas padronizadas de um indicador nos demais indicadores existentes em seu construto devem ser maiores que as suas cargas nos outros construtos (HAIR; *et al.*, 2014).

A Tabela 7 exhibe as cargas cruzadas dos indicadores nos construtos. É possível observar que todos os indicadores atendem ao critério das cargas cruzadas e nenhum tem valor maior fora de seu construto.

Tabela 7 – Cargas cruzadas dos indicadores do modelo ajustado

Indicador	Construto				
	AUT	MA	PROD	QA	SEG
AUT_1	0,795	0,5079	0,5407	0,4065	0,4712
AUT_2	0,784	0,5304	0,5902	0,5998	0,4635
AUT_3	0,6523	0,4014	0,4043	0,3986	0,4266
AUT_5	0,6535	0,3011	0,3102	0,3459	0,4513
AUT_7	0,7613	0,3811	0,5452	0,4162	0,3937
MA_1	0,4342	0,8128	0,635	0,4528	0,4906
MA_2	0,3649	0,7758	0,5031	0,4545	0,4141
MA_3	0,5402	0,877	0,6609	0,5378	0,5101
MA_4	0,6034	0,8756	0,7542	0,5746	0,5691
PROD_1	0,3885	0,5087	0,7395	0,4862	0,3614
PROD_2	0,6639	0,7192	0,7996	0,5457	0,4748
PROD_3	0,4573	0,5797	0,7647	0,5943	0,534
PROD_4	0,538	0,5763	0,8138	0,4537	0,4807
QA_1	0,6496	0,5682	0,611	0,8294	0,6325
QA_2	0,4287	0,3285	0,5121	0,6521	0,3804
QA_3	0,2889	0,3911	0,3693	0,6426	0,2609
QA_4	0,2452	0,41	0,3641	0,7202	0,3874
SEG_1	0,5011	0,4441	0,4515	0,392	0,8254
SEG_2	0,5016	0,5307	0,5037	0,5942	0,8991
SEG_3	0,5697	0,5812	0,5997	0,5989	0,8959

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

A Tabela 7 confirma a existência da validade discriminante, pois o valor de cada indicador é maior dentro do constructo do qual faz parte, comparado aos valores dos demais constructos.

A seguir os detalhes da avaliação do modelo de mensuração estrutural.

4.1.1.2 Avaliação do modelo estrutural

A primeira etapa para validação do modelo estrutural é a averiguação dos coeficientes de determinação de Pearson (R^2), que medem a capacidade preditiva do modelo. Os valores investigados se encontram na Tabela 8.

Tabela 8 – Coeficientes de determinação de Pearson (R^2)

Construto	R^2
AUT	0,68661
MA	0,74732
PROD	0,7911
QA	0,65983
SEG	0,63921

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

Todos os construtos obtiveram valores maiores que 0,26 (26%), o que indica um poder preditivo considerável para o modelo. A segunda etapa para validação do modelo estrutural é a análise do tamanho do efeito (f^2), para avaliar quanto cada construto é significativo para o ajuste do modelo. Ele mede o impacto de um construto antecessor sobre um construto interno.

Os valores de referência para f^2 são: 0,02 (pequeno), 0,15 (médio) e 0,35 (grande) segundo a literatura (COHEN, 1988; CHIN, 2010; HENSELER, 2010; HAIR *et al.*, 2014). O resultado de f^2 obtido é demonstrado na Tabela 9.

Tabela 9 – Tamanho do efeito (f^2)

Construto	f^2
AUT	0,30068
MA	0,48899
PROD	0,34534
QA	0,19394
SEG	0,50309

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

Dessa forma, os resultados de (f^2) apresentados pela Tabela 9 demonstram que todos os construtos AUT, PROD e QA apresentaram um médio efeito em nível estrutural, enquanto os construtos MA e SEG apresentaram um grande efeito em nível estrutural.

A terceira etapa para validação do modelo estrutural corresponde à avaliação do indicador de *Stoner-Geisser* (Q^2), que permite verificar a precisão do modelo ajustado. Os resultados obtidos para Q^2 são mostrados na Tabela 10.

Tabela 10 – Indicador de validade preditiva (Q²)

Construto	Q ²
AUT	0,3486
MA	0,5161
PROD	0,4774
QA	0,3091
SEG	0,4839

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

Como referência o valor de Q² deve ser maior que zero e quanto mais próximo de um, mais preciso é o modelo. Todos os construtos apresentam valor maior que zero.

A quarta etapa para validação do modelo estrutural é a conferência do *Goodness of Fit* (GoF), usado para medir a adequação ou ajuste geral do modelo, tanto do modelo estrutural quanto do modelo de mensuração. O valor de referência para o GoF é 0,36, considerado adequado para as ciências sociais e do comportamento (WETZELS; ODEKERKEN-SCHRODER; VAN OPPEN, 2009).

O cálculo do GoF é realizado pela média geométrica entre o R² médio – que mede a adequação do modelo estrutural - e a média ponderada das AVEs – que mede a adequação do modelo de mensuração. O resultado do cálculo é apresentado na Tabela 11.

Tabela 11 – Índice de ajuste do modelo (GoF)

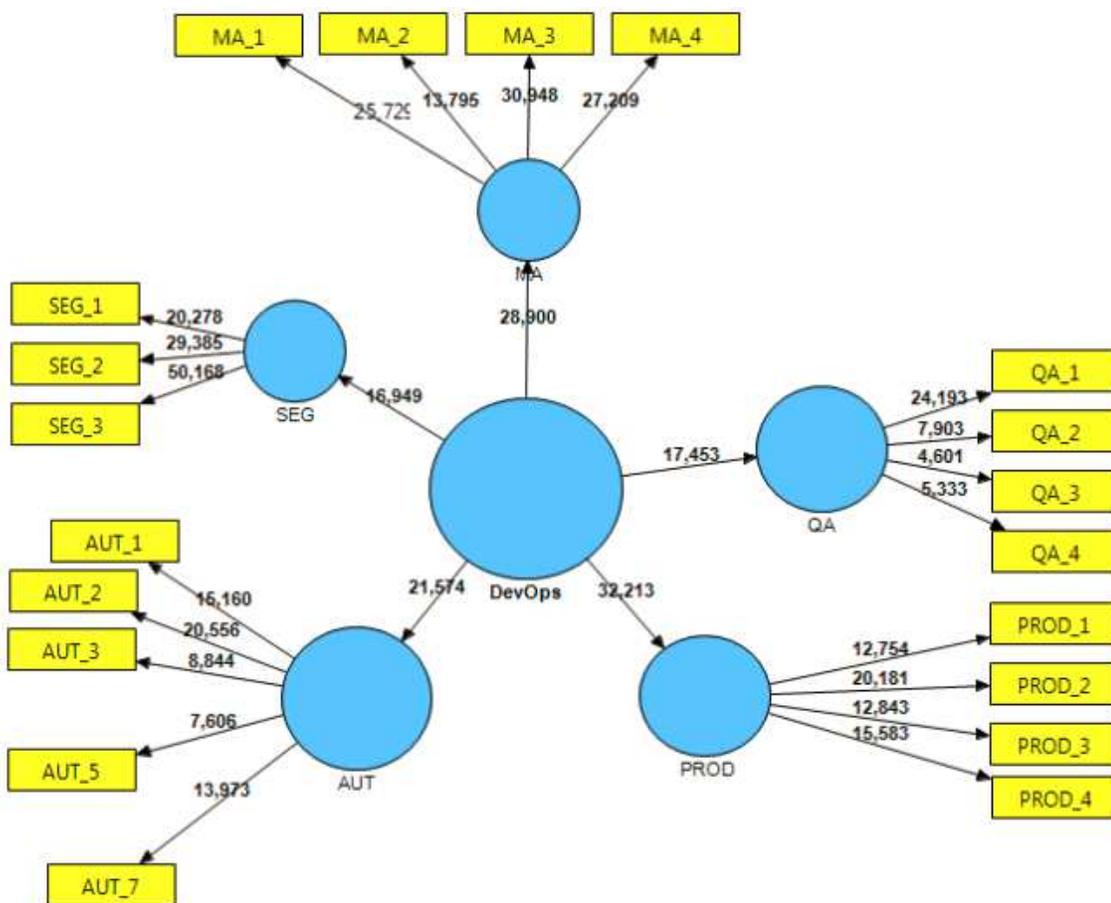
Construto	AVE	R ²
AUT	0,5358	0,68661
MA	0,6996	0,74732
PROD	0,6083	0,7911
QA	0,5112	0,65983
SEG	0,7641	0,63921
Média	0,6238	0,70481
GoF	0,6643043	

Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

O valor obtido pelo cálculo do GoF é 0,664, o que significa um bom ajuste do modelo.

A última etapa da avaliação do modelo estrutural é a análise dos coeficientes de caminho (Γ), que deve ser obtida com a execução do procedimento de *bootstrapping* para estimar os valores de t (*Student*) de todos coeficientes de caminho do modelo. Os resultados são exibidos na Figura 15.

Figura 15– Resultados do procedimento de *bootstrapping* (Teste t)



Fonte: adaptada dos resultados do *Software SmartPLS 2.0 M3*

Os valores dos resultados para todos os coeficientes de caminho obtidos foram maiores que 1,96 para o valor de t , confirmando que há uma relação expressiva entre cada indicador e seu respectivo construto.

4.2 Análise e discussão dos resultados

O modelo criado para esta pesquisa foi construído com base nas principais fontes de literatura especializada sobre o tema.

Buscou-se garantir uma maior coerência com a realidade e o modelo proposto e seus indicadores foram previamente validados por especialistas em tecnologia da informação e *DevOps* em um teste piloto.

A análise dos dados foi orientada pela técnica de modelagem de equações estruturais baseada em mínimos quadrados parciais (*partial least squares*) ou MEEPLS.

Os ajustes que foram executados no modelo, como a dos indicadores AUT_4, AUT_6 do construto, correspondentes à automação, e o indicador MA_5 do construto, correspondente à mentalidade ágil, não comprometeram a validade do modelo, ao contrário, possibilitaram seu refinamento e aumento de sua confiabilidade, conforme já discutidos na seção metodologia (HAIR *et al.*, 2013).

Na próxima seção será discutido o resultado do modelo com suas dimensões e seus respectivos parâmetros e indicadores.

4.2.1 Comportamento dos parâmetros e seus indicadores

Os resultados iniciais coletados no *software* SmartPLS mostraram a necessidade de ajustes no modelo de pesquisa por falta de alinhamento estatístico dos indicadores aos parâmetros. O comportamento dos parâmetros e seus respectivos indicadores são discutidos individualmente por parâmetro.

4.2.1.1 Dimensão DevOps

4.2.1.1.1 Construto Velocidade e Automação (AUT)

No modelo inicial o construto foi formado por sete indicadores. Na validação inicial do modelo dois foram excluídos, AUT_4 e AUT_6, pois tiveram cargas fatoriais abaixo do recomendado pela literatura explanada na metodologia (abaixo de 0,6).

O indicador AUT_4 corresponde à utilização de repositórios controlados para auxílio no controle de versão dos códigos fontes e/ou configuração de ambientes/infra. Isso pode demonstrar uma certa resistência ao uso desse tipo de ferramenta, visto que existe uma mudança no processo a ser executada para o controle de versionamento de *softwares* e infraestrutura.

Isso é parte fundamental da automação do processo de implantação das entregas de TI adotadas pelo conceito de DevOps, que proporcionam maior velocidade e robustez, além de também ser possível controlar qual versão é implantada nos diversos ambientes, possibilidade de visualizar o repositório de controle de revisão do projeto e até mesmo executar um único comando para criar e implantar o aplicativo/ambiente em qualquer dispositivo acessível (HUMBLE; FARLEY, 2010; SHARMA; COYNE, 2015).

O indicador AUT_6 é correspondente ao tempo médio necessário para a validação/testes de uma alteração/correção em *software*/serviço e deve ser menor do que o tempo total de seu desenvolvimento.

Isso pode demonstrar uma certa resistência, também relacionada ao indicador AUT_4, pois também se faz necessária uma mudança no processo de escrita dos códigos de *software* e na maneira com que se trabalha com a infraestrutura.

De acordo com os conceitos de *DevOps*, o tempo para se testar uma funcionalidade deve ser mais rápido do que o seu desenvolvimento, pelo fato de se utilizar ferramentas para testes automatizados. Com ciclos de entregas cada vez mais curtos e velozes, é praticamente impossível a execução manual de todos os testes necessários para garantir que as funcionalidades estejam de acordo.

É necessária, então, a utilização de ferramentas para execução de testes automatizados, pois facilitam a detecção dos defeitos o quanto antes possível para que não interfiram no processo de entrega (HUMBLE; FARLEY, 2010).

Os indicadores foram removidos por não ter relevância estatística, ou seja, não fazia sentido para os respondentes, pois o modelo é validado com base nas suas experiências profissional.

Há uma certa resistência dos profissionais de TI em utilizar as ferramentas de controle de versão e ferramentas de testes automatizados, como identificado nos trabalhos de Patwardhan *et al* (2016) e Ghantous e Gill (2017) relacionados à resistência a adoção dos conceitos dos modelos ágil e DevOps. Embora conceitualmente seja importante, na prática dos respondentes isso não se confirma e explica a carga baixa desses indicadores.

O construto velocidade e automação ficou configurado com cinco indicadores, mostrados no Quadro 12 com suas respectivas cargas fatoriais.

Quadro 12 – Indicadores de velocidade/automação e suas cargas fatoriais

Indicadores	Descrição	Cargas Fatoriais
AUT_1	Frequência de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação.	0,778012
AUT_2	Tempo médio de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação	0,778188
AUT_3	Utilização de processos de rollback automatizados quando problemas/bugs são encontrados nas atualizações executadas em produção.	0,635666
AUT_5	Utilização de ferramentas para testes automatizados durante a validação da maioria das alterações nas aplicações/serviços.	0,661028
AUT_7	Ambiente de teste/validação atualizado em relação ao produtivo	0,746421

Fonte: adaptado dos resultados do *Software SmartPLS 2.0 M3*

A remoção dos dois indicadores desse construto o ajustou e aumentou a carga fatorial dos demais indicadores para valores acima de 0,60, conforme proposto por Hair *et al.* (2014) como indicado.

Verifica-se que o grupo de indicadores representa suficientemente a velocidade e automação como capacidades e habilidades necessárias para a adoção ao DevOps para os profissionais de TI respondentes.

Dois indicadores, AUT_1 e AUT2, apresentaram carga fatorial superior a 0,77, exercendo maior influência do que os demais indicadores na velocidade/automação AUT_1 e AUT_2, que correspondem respectivamente à frequência e tempo médio de implantação de novas funcionalidades no ambiente produtivo.

Isso indica que implantações frequentes demonstram melhorias contínuas aos aplicativos e serviços, o que os mantém competitivos com novos recursos. As atualizações frequentes demonstram que a TI responde satisfatoriamente às necessidades das organizações.

Tradicionalmente, os sistemas utilizados nas organizações foram desenvolvidos totalmente interligados. A cada nova funcionalidade a ser implantada o sistema ou serviço deveria ficar indisponível e seriam necessárias grandes janelas de indisponibilidade para sua implantação.

Há uma década era aceitável agendar janelas de inatividade de cinco horas para implantações no meio da noite, sem consequências negativas para as organizações e seus clientes. Atualmente, a maioria das organizações tem uma base de clientes que está sempre ativa em algum lugar do mundo e são necessárias implantações mais velozes em janelas mais curtas de indisponibilidade.

O resultado da AVE para o construto AUT foi de 0,53, que assegura correlação positiva entre os indicadores e o construto, pois Henseler *et al.* (2009) propõem que a AVE deve ser maior que 0,50 para que isso ocorra.

A partir desses resultados pode-se concluir que o grupo de indicadores para esse construto (AUT) representa satisfatoriamente a velocidade e automação como capacidades e habilidades necessárias para a adoção do *DevOps*.

É importante salientar que a literatura especializada destaca a seriedade dos tópicos mensurados pelos indicadores usados nesse construto, pois a velocidade oferece vantagem competitiva para a empresa, melhora a forma como as empresas trabalham com a TI.

O conceito de *DevOps* implica automatizar tanto quanto possível para acelerar processos e promover a redução de erros humanos (HPE, 2016; PUPPET, 2016; PEREIRA, 2017).

A automação é a chave para uma colaboração eficiente e uma estreita integração e sinergia entre desenvolvimento e operações, gerando velocidade para ambos os times em suas entregas. A comunidade *DevOps* está em constante busca por novas abordagens e ferramentas para implantar processos automatizados (WETTINGER; *et al.*, 2016).

Confirmam ainda mais a necessidade de automação Aiello e Sachs (2016), que citam que uma das principais lições chave do *DevOps* é o poder e controle obtidos ao automatizar processos. Sem automação, o *DevOps* e o processo de entrega e de implantação contínua simplesmente não seriam possíveis.

Assim, o construto AUT justificou-se estatisticamente no modelo de pesquisa.

4.2.1.1.2 Construto Qualidade (QA)

No modelo inicial, este construto foi formado por quatro indicadores e na validação inicial do modelo não foi necessário remover indicador algum, pois todos obtiveram cargas fatoriais acima do recomendado pela literatura explanada na metodologia (acima de 0,6), conforme demonstrado no Quadro 13.

Quadro 13 – Indicadores de qualidade e suas cargas fatoriais

Indicadores	Descrição	Cargas Fatoriais
QA_1	Adoção de processos de liberação em produção executado de forma regular e com controle dos riscos.	0,826332
QA_2	Baixo número de incidentes e defeitos relatados em um produto ou serviço após implantação para o ambiente produtivo	0,654785
QA_3	Prática de reuniões de retrospectiva pós implantação/lições aprendidas para assegurar qualidade do processo de entrega.	0,646779
QA_4	Envolvimento e participação adequada dos <i>Stakeholders</i> dentro do ciclo de vida do projeto.	0,718089

Fonte: adaptado dos resultados do *Software SmartPLS 2.0 M3*

Observa-se que o grupo de indicadores representa suficientemente a qualidade como capacidade e habilidade necessária para a adoção do DevOps para os profissionais de TI respondentes.

Dois indicadores, QA_1 e QA_2, apresentaram carga fatorial superior a 0,71, portanto exercem maior influência que os demais no construto qualidade:

- QA_1 corresponde à adoção de processos regulares e com controle dos riscos em implantações no ambiente produtivo; e
- QA_4 corresponde ao envolvimento de forma adequada dos *stakeholders* no ciclo de vida do projeto. O envolvimento de equipes de qualidade no processo de desenvolvimento deve ocorrer o quanto antes e quantas vezes forem necessárias.

A adoção de processos regulares e com o controle dos riscos nas implantações de novas funcionalidades no ambiente produtivo é previsto pelos modelos de gestão de serviços ITIL e ISO20000 dentro do ciclo de vida de transição de serviços.

Esses serviços direcionam como se deve efetivar a transição de alterações para ambientes operacionais gerenciados e detalham, principalmente, os processos de gerenciamento de mudanças, gerenciamento da liberação e implantação de serviços de TI (FERNANDES; ABREU; 2014).

O envolvimento de forma adequada dos *stakeholders* no ciclo de vida do projeto é previsto pelo *SCRUM*.

O cliente deve estar intimamente envolvido no processo de desenvolvimento do projeto e sua principal função é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações dentro do *Sprint* do *SCRUM* (SOMMERVILLE, 2011).

O resultado da AVE para o construto QA foi de 0,51, que assegura correlação positiva entre os indicadores e o construto. Henseler *et al.* (2009) propõem que a AVE deve ser maior que 0,50 para que isso ocorra.

A partir desses resultados, pode-se concluir que o grupo de indicadores para esse construto (QA) representa satisfatoriamente a qualidade como uma capacidade e habilidade necessárias para a adoção do *DevOps*.

É importante salientar que a literatura especializada destaca a importância dos tópicos mensurados pelos indicadores usados nesse construto, pois as organizações contam com *software* para fornecer valores e experiências diferenciadas para seus clientes e atualmente esse é utilizado como uma ferramenta chave para os negócios, gerando bons resultados e com isso novas necessidades devem ser entregues rapidamente e com alta qualidade (FORRESTER, 2016; HPE, 2016; KIM; *et al.* 2016; PEREIRA, 2017).

Construir *softwares* e serviços com qualidade exige que seja feito o que é correto desde o início do ciclo de vida do projeto. A abordagem *DevOps* não somente deve envolver o time de operações, mas também todos os envolvidos no projeto, incluindo seus patrocinadores, o que melhora significativamente a qualidade do projeto no geral (AIELLO; SACHS, 2016).

Dessa forma, o construto QA justificou-se estatisticamente no modelo de pesquisa.

4.2.1.1.3 Construto Segurança (SEG)

No modelo inicial, esse construto foi formado por três indicadores e na validação inicial desse modelo não foi necessário remover nenhum, pois todos obtiveram cargas fatoriais acima do recomendado pela literatura explanada na metodologia (acima de 0,6), conforme demonstrado pelo Quadro 14.

Quadro 14 – Indicadores de segurança e suas cargas fatoriais

Indicadores	Descrição	Cargas Fatoriais
SEG_1	Realização de testes de segurança e vulnerabilidade antes da promoção para a produção.	0,824788
SEG_2	Envolvimento do time de segurança dentro do ciclo de vida do projeto/serviço.	0,899368
SEG_3	Políticas e processos de segurança e conformidade para desenvolvimento e implantações de projetos e/ou serviços conhecidas e expostas de forma clara e coesa.	0,896192

Fonte: adaptado dos resultados do *Software SmartPLS 2.0 M3*

Nota-se que o grupo de indicadores representa suficientemente a segurança como capacidade e habilidade necessária para a adoção do DevOps pelos profissionais de TI respondentes. Dois indicadores, entretanto, SEG_2 e SEG_3, apresentaram carga fatorial superior a 0,89 e exercem maior influência que o indicador SEG_1 no construto segurança.

SEG_2 corresponde ao envolvimento do time de segurança dentro do ciclo de vida do projeto/serviço e SEG_3 corresponde à utilização de políticas e processos de segurança e conformidade para desenvolvimento e implantações de projetos e/ou serviços que são conhecidos e expostos de forma clara e coesa por toda a equipe.

Um dos objetivos do *DevOps* é ter equipes de desenvolvimento envolvidas com o time de segurança da informação o mais cedo possível, ao contrário de se envolver principalmente no final do projeto.

Visualizar as vulnerabilidades de segurança antecipadamente garante que as compilações sejam estáveis antes de passar para a próxima etapa de lançamento da funcionalidade no ambiente produtivo.

Ao invés de executar os testes de segurança no final do processo, pode-se obter melhores resultados tornando a segurança uma parte do trabalho diário de todos. Isso significa integrar testes de segurança e controles no trabalho diário de desenvolvimento, controle de qualidade e operações.

Para isso, políticas e processos de segurança devem ser criados e seguidos por todos os envolvidos, e resultam em evidências para auditores, avaliadores ou qualquer outra necessidade (KIM *et al*, 2016; HPE, 2016; PUPPET, 2016).

O resultado da AVE para o construto SEG foi de 0,76, que assegura uma forte correlação positiva entre os indicadores e o construto, pois Henseler *et al.* (2009) propõem que a AVE deve ser maior que 0,50 para que isso ocorra.

A partir desses resultados pode-se concluir que o grupo de indicadores para esse construto (SEG) representa satisfatoriamente a segurança como capacidade e habilidade necessárias para a adoção do *DevOps*.

É importante destacar que a literatura especializada demonstra a importância dos tópicos mensurados pelos indicadores usados nesse construto, pois a principal característica do *DevOps* é buscar a integração total entre os proprietários de produtos, desenvolvimento, QA, operações de TI e segurança da informação para que trabalhem em conjunto não só para ajudar uns aos outros, mas também para garantir que a organização de forma geral seja bem-sucedida.

Uma das formas de se obter essa integração necessária e esperada é convidar a equipe de segurança da informação para as demonstrações dos produtos no final de cada intervalo de desenvolvimento, para que possam entender melhor os objetivos da equipe e das necessidades da organização, a fim de fornecer orientações e comentários nas primeiras etapas do projeto ou serviço, quando ainda há maior quantidade de tempo e liberdade para fazer correções (KIM *et al*, 2016).

Por não estar envolvido, o grupo de segurança da informação infelizmente pode estar atrasado para manter seguros os ativos corporativos de possíveis indisponibilidades, pois em um ambiente DevOps se faz necessário buscar uma sinergia entre a segurança da informação e o time de desenvolvimento, com o compartilhamento de conhecimentos técnicos e os benefícios da aprendizagem de melhores práticas relacionadas à segurança (AIELLO; SACHS, 2016; PUPPET 2016).

Visualizar as vulnerabilidades de segurança antecipadamente garante que as compilações sejam estáveis antes de serem disponibilizadas no ambiente produtivo. Além disso, integrar a segurança pode ajudar a superar a resistência à adoção do DevOps, pois à medida que as vulnerabilidades diminuem percebe-se que o aumento da velocidade não degrada a estabilidade do produto (HPE, 2016).

Dessa forma, o construto SEG justificou-se estatisticamente no modelo de pesquisa.

4.2.1.1.4 Construto Produtividade (PROD)

No modelo inicial esse construto foi formado por quatro indicadores e na validação inicial não foi necessário remover nenhum, pois todos obtiveram cargas fatoriais acima do recomendado na literatura explanada na metodologia (acima de 0,6), conforme demonstrado pelo Quadro 15.

Quadro 15 – Indicadores de produtividade e suas cargas fatoriais

Indicadores	Descrição	Cargas Fatoriais
PROD_1	Correção dos erros encontrados em ambiente produtivo após implantação de alguma nova aplicação ou funcionalidade dentro do tempo acordado.	0,738877
PROD_2	As metas da equipe são claras e estão relacionadas com os objetivos de negócios da organização.	0,798914
PROD_3	O uso de painéis visuais com estatísticas de desempenho (entregas e atividades realizadas, à realizar, etc..) disponível para todos os integrantes da equipe favorece a gestão do trabalho e melhora a produtividade.	0,765736
PROD_4	Usualmente as equipes entregam correções e novas funcionalidades nos serviços/aplicações que realmente são importantes para o negócio.	0,814045

Fonte: adaptado dos resultados do *Software SmartPLS 2.0 M3*

Observa-se que o grupo de indicadores representa de forma satisfatória a produtividade como capacidade e habilidade necessária para a adoção do DevOps pelos profissionais de TI respondentes.

Dois indicadores, contudo, PROD_4 e PROD_2, apresentaram carga fatorial superior a 0,79 exercendo maior influência que os demais no construto produtividade. PROD_4 corresponde a entregas de correções e novas funcionalidades nos serviços/aplicações pelas equipes, que realmente são importantes para o negócio, e PROD_2 corresponde às metas da equipe, que devem ser claras e estarem relacionadas com os objetivos de negócios da organização.

Os dois indicadores estão relacionados entre si, demonstrando que os esforços das equipes de TI estão focados na entrega de itens de maior valor para o negócio e buscam melhorar a produtividade para oferecer tecnologia e inovação para suportar os objetivos de negócios.

Isso é alcançado com a utilização de ferramentas, métodos e conceitos que aumentam a produtividade e reduzem o desperdício, e o propósito principal do *DevOps* é auxiliar a TI a criar mais valor para o negócio, o que não ocorrerá se os esforços não estiverem adequadamente alinhados com a estratégia de negócio da organização, não importa o quão rápido se possa desenvolver e entregar as atividades.

Uma das formas de se obter esse alinhamento é a utilização de painéis visuais com todas as atividades a serem executadas e em execução pelo time com suas metas e objetivos bem aclarados. A equipe pode ser direcionada com o foco na estratégia da organização e resulta em maior retorno e valor para o negócio (HPE, 2016; PEREIRA, 2016).

O resultado da AVE para o construto PROD foi de 0,60, que assegura uma correlação positiva entre os indicadores e o construto, já que Henseler; *et al.* (2009) propõem que a AVE deve ser maior que 0,50 para que isso ocorra.

A partir desses resultados pode-se concluir que o grupo de indicadores para esse construto (PROD) representa satisfatoriamente a produtividade como capacidade e habilidade necessárias para a adoção do *DevOps*.

Dessa forma, o construto PROD justificou-se estatisticamente no modelo de pesquisa.

4.2.1.1.5 Construto Mentalidade Ágil (MA)

No modelo inicial esse construto foi formado por cinco indicadores. Na validação inicial um indicador foi excluído, o MA_5, pois obteve carga fatorial abaixo do recomendado pela literatura explanada na metodologia, que é abaixo de 0,6.

O indicador MA_5 corresponde à iniciativa de as pessoas realizarem tarefas ou assumi-las, ao invés de esperar pela atribuição das atividades por um líder. Isso sugere que na experiência dos respondentes, a proatividade não apareceu como um fator relevante por não ocorrer de uma forma geral.

Isso pode ser sustentado pela pesquisa realizada por Araújo e Gava (2011) em 257 organizações durante cinco anos, em que se verificou que o nível de proatividade ficou entre baixo e médio em cerca de 90% das empresas consultadas, o que corrobora com o resultado desse indicador.

De acordo com a literatura utilizada neste estudo, a proatividade se torna necessária para a utilização do modelo DevOps, principalmente para assumir novas atividades a serem feitas, como, por exemplo, em um caso de falha de *software* ou serviço o time precisa estar preparado para reagir de forma proativa e resolver o problema rapidamente (SATO, 2014).

O construto mentalidade ágil ficou configurado com quatro indicadores, mostrados na Quadro 16 com suas respectivas cargas fatoriais.

Quadro 16 – Indicadores de mentalidade ágil e suas cargas fatoriais

Indicadores	Descrição	Cargas Fatoriais
MA_1	Adoção do uso de métodos ágeis no ciclo de vida do projeto/serviço.	0,826332
MA_2	As pessoas são incentivadas a questionar umas as outras para encontrar as melhores soluções possíveis.	0,654785
MA_3	Aprendizagem contínua.	0,646779
MA_4	As mudanças internas ou externas que impactam no negócio são analisadas e geram respostas rápidas por parte da equipe.	0,718089

Fonte: adaptado dos resultados do *Software SmartPLS 2.0 M3*

A remoção desse indicador do construto ajustou-o, aumentou a carga fatorial dos demais indicadores para valores acima de 0,6, conforme proposto por Hair; *et al.* (2014) como recomendado. Observa-se que o grupo de indicadores representa de forma satisfatória a produtividade como capacidade e habilidade necessária para a adoção do DevOps pelos profissionais de TI respondentes.

Dois indicadores, contudo, QA_1 e QA_4, apresentaram carga fatorial superior a 0,71 e exercem maior influência que os demais indicadores no construto mentalidade ágil, em que QA_1 corresponde à adoção de métodos ágeis no ciclo de vida do projeto/serviço e QA_4 corresponde a responder rapidamente às mudanças externas e internas que impactam o negócio.

A adoção de métodos ágeis no ciclo de vida de projetos e/ou serviços é vital para a utilização do *DevOps*, visto que teve início com a aplicação e utilização de princípios ágeis no setor de operações, pois a metodologia ágil estava presente somente nos projetos de desenvolvimento de *software*.

Isso gerou alguns conflitos para a área de infraestrutura, que não tinha a mesma rapidez e velocidade para entregar a infraestrutura e as operações pós-implantação do *software*, como ocorria com as entregas dos desenvolvedores de projetos que já utilizavam ágil como metodologia (KIM *et al.*, 2016; MUELLER, 2016).

Responder rapidamente às mudanças está relacionado a um dos princípios da metodologia ágil de desenvolvimento de sistemas. As alterações e retrabalho são essenciais e as especificações do projeto mudam com o seu progresso, pois são realizadas entregas incrementais de pequenas partes viáveis de serem usadas, o mínimo produto viável, que amadureceu de acordo com as necessidades das organizações e mudanças de mercado (Sommerville, 2011; KIM *et al.*, 2016; MUELLER, 2016).

O resultado da AVE para o construto MA foi de 0,69, que assegura a correlação positiva entre os indicadores e o construto, já que Henseler *et al.* (2009) propõem que a AVE deve ser maior que 0,50 para que isso ocorra. A partir desses resultados pode-se concluir que o grupo de indicadores para esse construto (MA) representa satisfatoriamente a mentalidade ágil como capacidade e habilidade necessárias para a adoção do *DevOps*.

É importante salientar que a literatura especializada destaca a importância dos tópicos mensurados pelos indicadores usados nesse construto, na medida em que um dos princípios do *DevOps* é de quebrar as barreiras entre desenvolvimento e operações e o sucesso do modelo está ligado principalmente à mudança da mentalidade.

A intenção do *DevOps* é de unir entrega ágil, contínua, automação e muito mais para ajudar as equipes de desenvolvimento e operações a serem mais eficientes, inovar e atender às demandas mais rapidamente e oferecer maior valor para empresas e clientes (ATLASSIAN, 2017; ENGATES, 2014).

Dessa forma, o construto MA justificou-se estatisticamente no modelo de pesquisa.

4.2.2 Dimensões chave do modelo DevOps

Após a análise dos resultados obtidos com a validação do modelo de mensuração deste trabalho, foi possível observar e organizar uma síntese com as principais dimensões do modelo *DevOps*.

O Quadro 17 demonstra os construtos com suas cargas totais, bem como os seus indicadores e as cargas de cada um deles.

Quadro 17 – Síntese dos resultados da pesquisa

Dimensão	Carga Total do Construído	Capacidades e Habilidades	Código	Indicador	Carga Individual por Indicador	Observações
Produtividade (Prod)	0,889	Tempo de restauração de um serviço/aplicativo quando ocorre uma falha	Prod_1	Correção dos erros encontrados em ambiente produtivo após implantação de alguma nova aplicação ou funcionalidade dentro do tempo acordado.	0,739516	- Esforços alinhados com o negócio e estratégia da organização; - Foco em gerar valor para o negócio; e - Visibilidade plena de todas as atividades por todos da equipe.
		Alinhamento com o Negócio	Prod_2	As metas da equipe são claras e estão relacionadas com os objetivos de negócios da organização.	0,799584	
		Medição de produtividade de forma visual	Prod_3	O uso de painéis visuais com estatísticas de desempenho (entregas e atividades realizadas, à realizar, etc.) disponível para todos os integrantes da equipe favorece a gestão do trabalho e melhora a produtividade.	0,764735	
		Foco no valor e alinhamento dos requisitos	Prod_4	Usualmente as equipes entregam correções e novas funcionalidades nos serviços/aplicações que realmente são importantes para o negócio.	0,813756	
Mentalidade Ágil (MA)	0,864	Utilização da metodologia ágil	MA_1	Adoção do uso de métodos ágeis no ciclo de vida do projeto/serviço.	0,812808	- Ambiente favorável a comunicação entre todos da equipe; - Estimular a aprendizagem; - Resposta rápida para as mudanças que ocorrerem;
		Liberdade para agregar valor	MA_2	As pessoas são incentivadas a questionar umas as outras para encontrar as melhores soluções possíveis.	0,775797	
		Habilidade de se atualizar de forma contínua	MA_3	Aprendizagem contínua.	0,87696	
		Habilidade em ser flexível as mudanças	MA_4	As mudanças internas ou externas que impactam no negócio são analisadas e geram respostas rápidas por parte da equipe.	0,875648	
Automação Velocidade (Aut)	0,829	Processo de implantação automatizado	Aut_1	Frequência de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação.	0,79496	- Automação de processos mecânicos para redução de falha humana; - Velocidade para criar e suportar ambientes; e - Integração e Controle de versões de software. - Automação de testes;
			Aut_2	Tempo médio de implantação de novas versões no ambiente produtivo, considerando todo o processo de validação	0,784023	
		Automação de processos/eficácia no controle	Aut_3	Utilização de processos de rollback automatizados quando problemas/bugs são encontrados nas atualizações executadas em produção.	0,652337	
		Automação de testes/Garantia da Qualidade	Aut_5	Utilização de ferramentas para testes automatizados durante a validação da maioria das alterações nas aplicações/serviços.	0,653474	
			Aut_7	Ambiente de teste/validação atualizado em relação ao produtivo	0,761309	
Qualidade (QA)	0,812	Confiabilidade no processos de entrega	QA_1	Adoção de processos de liberação em produção executado de forma regular e com controle dos riscos.	0,829402	- Procedimentos para controle de versões de software; - Aprendizagem com histórico de implantações;
		Grande cobertura para levantamento de Requisitos e cenários de testes	QA_2	Baixo número de incidentes e defeitos relatados em um produto ou serviço após implantação para o ambiente produtivo	0,652119	
			QA_3	Prática de reuniões de retrospectiva pós implantação/lições aprendidas para assegurar qualidade do processo de entrega.	0,642577	
		Engajamento dos stakeholders	QA_4	Envolvimento e participação adequada dos Stakeholders dentro do ciclo de vida do projeto.	0,720185	
Segurança (Seg)	0,800	Desenvolver de forma correta, sem vulnerabilidades	Seg_1	Realização de testes de segurança e vulnerabilidade antes da promoção para a produção.	0,825377	- Prever envolvimento do time de segurança; - Seguir políticas e procedimentos de conformidade; - Envolvimento da equipe de segurança;
		Envolvimento do time de segurança	Seg_2	Envolvimento do time de segurança dentro do ciclo de vida do projeto/serviço.	0,89913	
		Política de segurança/compliance	Seg_3	Políticas e processos de segurança e conformidade para desenvolvimento e implantações de projetos e/ou serviços conhecidas e expostas de forma clara e coesa.	0,89594	

Os resultados da pesquisa mostram que as habilidades e capacidades necessárias para a adoção ao DevOps se baseiam em cinco principais pilares em sequência: produtividade; mentalidade ágil; velocidade e automação; e qualidade e segurança.

- **produtividade:** se deve à necessidade de as empresas se manterem competitivas perante a evolução do mercado;
- **mentalidade ágil:** para que os profissionais de TI tenham em mente algumas das premissas do modelo SCRUM;
- **velocidade e automação:** para que realmente possam aumentar a produtividade com o auxílio de ferramentas de automação de processos repetitivos;
- **qualidade:** com o ganho de testes automatizados; e
- **segurança:** ser envolvida desde o princípio do ciclo de vida da alteração/demanda.

De fato, isso demonstra como os profissionais de TI consideram seus esforços em relação às necessidades das organizações em aumentar a sua competitividade no mercado, conforme observado na revisão de literatura no que diz respeito ao surgimento do *DevOps*, demonstrado também pelo construto mentalidade ágil.

Com isso, velocidade e automação demonstram que para atender à agilidade dos desenvolvedores se faz necessário tornar velozes processos que podem ser automatizados com o surgimento de diversas ferramentas para auxílio dessa atividade. Assim, a eliminação ou redução de atividades manuais mais suscetíveis a falhas podem trazer mais qualidade final aos processos que resultarão na entrega de *software*.

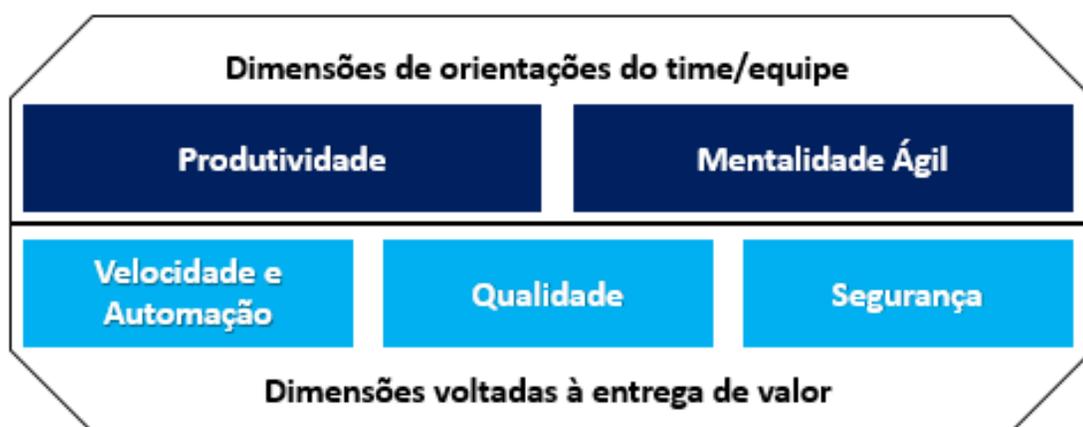
Um ponto a ser observado é com relação à segurança, pois o envolvimento desse tema entre desenvolvedores e equipes de operações mostra uma possível ineficácia no que diz respeito aos processos de política de segurança e implantação de funcionalidades sem os devidos testes de vulnerabilidade.

De uma forma geral, o modelo conceitual validado estatisticamente confirma os aspectos importantes encontrados na literatura, que o *DevOps* surgiu para suprir uma necessidade observada pela área de operações em TI, visando resolver uma alta demanda de serviços e *softwares* de TI em um tempo cada vez menor.

Por essa razão é necessário que se busquem produtividade e mentalidade adequada ao novo esquema ágil, de forma a alcançar os resultados com o uso de ferramentas que asseguram automação e velocidade para gerar maior qualidade alinhada à segurança, sem ter produtividade e mentalidade ágil não se chega ao resultado esperado.

Os resultados desta pesquisa permitiram que o *DevOps* se dividisse em duas dimensões de direcionamento, conforme demonstrado na Figura 16.

Figura 16 – Dimensões chaves para adoção ao DevOps



As duas dimensões de orientação do time/equipe demonstram que em um ambiente que pretende adotar o modelo *DevOps*, os profissionais envolvidos necessitam preocupar-se com a produtividade e ter mentalidade ágil, incorporando em seu comportamento os conceitos *DevOps*.

Já as dimensões de entrega de valor devem ser utilizadas como orientação para foco dos esforços de trabalho dos profissionais em ferramentas que auxiliam a melhoria e aumento da velocidade e automação e qualidade e segurança, para garantirem os resultados que o modelo *DevOps* se propõe a entregar, quando bem utilizado.

5 CONSIDERAÇÕES FINAIS

Esta pesquisa se dedicou ao estudo das habilidades e capacidades necessárias para a utilização das boas práticas do *DevOps*, que atualmente é uma tendência crescente nas empresas que trabalham e utilizam a tecnologia da informação, pela agilidade, eficiência e eficácia que pode proporcionar às entregas da área de TI.

As mudanças que ocorrem no mercado precisam ser absorvidas, desenvolvidas e entregues de forma mais rápida, gerando respostas também de forma rápida. A literatura especializada mostra que a gestão de projetos baseada no modelo tradicional tende a ser alterada para a metodologia ágil, pela sua boa aderência a mudanças e suas entregas e resultados serem obtidos de forma mais efetiva.

Há também a tendência em aplicar a metodologia ágil em operações pós-projetos, pois essa área precisa se adequar às entregas mais rápidas, obtidas pelos modelos de gerenciamento ágeis, que já estão tomando conta dos escritórios de projetos.

Dentro desse cenário explorou-se a validação estatística de um modelo teórico das principais capacidades e habilidades necessárias para a adoção ao emergente modelo *DevOps*, para que as entregas ocorram de forma mais rápida e possam ser garantidas com o aumento da relação de trabalho colaborativo entre as áreas de desenvolvimento e operações TI.

Com base na literatura especializada foram definidos conjuntos de parâmetros e indicadores de avaliação que permitissem perceber as capacidades e habilidades dos profissionais de TI com relação aos princípios que devem ser praticados para a adoção ao modelo *DevOps*.

A análise realizada na literatura mostrou uma equivalência de vários elementos comuns entre as principais dimensões em que o *DevOps* precisa ser estruturado, cuja definição se baseia nos seguintes atributos: velocidade/automação; qualidade; mentalidade ágil; produtividade e segurança.

Dessa forma, um modelo conceitual com as principais habilidades e capacidades em *DevOps* foi gerado e testado na pesquisa realizada com os profissionais de TI da Região Metropolitana do Vale do Paraíba e Litoral Norte (RMVALE), com o intuito de validá-lo com base nas experiências desses profissionais.

A metodologia empregada na pesquisa é de natureza quantitativa. O levantamento dos dados foi realizado a partir da aplicação de um questionário a uma amostra de 3.000 profissionais de TI da RMVALE, que resultou em um retorno de 108 respostas.

A análise dos dados foi feita pela técnica estatística de modelagem de equações estruturais com mínimos quadrados parciais e modelo de caminhos (MEEPLS). Para a análise dos dados utilizou-se o *software* SmartPLS 2.0M3 e planilhas do Excel.

Com relação aos 23 indicadores definidos inicialmente neste modelo de pesquisa, observou-se que se mostraram coesos com as habilidades e capacidades necessárias para a utilização do *DevOps*. A exclusão de alguns indicadores do modelo de seleção inicial durante a sua primeira validação não ocorreu pelo fato de os indicadores não serem importantes para o modelo testado e sim pelo rigor do modelo estatístico utilizado.

O estudo permitiu aprofundamento dos modelos *Waterfall* e ágil para gestão de projetos e dos modelos ITIL e ISO 20000 para operações e serviços, e a partir disso pode-se entender que o *DevOps* é uma realidade para muitas organizações. Ainda há pouca literatura especializada sobre o tema se comparado aos modelos mais tradicionais de desenvolvimento de *software* e gestão de operações e serviços, muito embora tem sido amplamente discutido e adotado por profissionais de TI e organizações.

Este estudo demonstra, a partir de um modelo estatisticamente validado, que as áreas-chaves para adoção do *DevOps* são: produtividade, mentalidade ágil; velocidade e automação; qualidade e segurança. Em geral, as organizações e especificamente os profissionais de TI devem dedicar atenção a essas áreas para fazer com que o *DevOps* traga todos os seus benefícios potenciais e o valor dele esperado.

Por ainda ser relativamente novo no Brasil, o *DevOps* se mostra como uma área muito promissora para fins de pesquisas e publicações, além de se mostrar muito eficiente aos resultados que propõe. Por ser uma área também relativamente nova na indústria e na academia, não existe ainda um padrão de práticas que as organizações utilizem em seu dia a dia.

Como o movimento de *DevOps* é relativamente novo, novas pesquisas devem ser realizadas para aprofundar e validar esses resultados, utilizar critérios mais diretos, como avaliar a tendência de adoção ao modelo e também se basear em uma maior compreensão das pessoas, das iterações entre elas e das tarefas que envolvem projetos de desenvolvimento de *softwares*, serviços e operações.

REFERÊNCIAS

AIELLO, B.; SACHS, L. **Agile application lifecycle management: using DevOps to Drive Process Improvement**. Estados Unidos: Pearson Higher, 2016.

AGILE MANIFESTO. **Agile manifesto**. 2001. Disponível em: <<http://agilemanifesto.org/>>. Acesso em 07-01-2017 as 19:45.

ALLSPAW, J.; HAMMOND, P. **10 + Deploys Per Day: dev and ops cooperation at flickr presentation**. San Francisco-California: Data Center Knowledge, 2009. Disponível em: <http://assets.en.oreilly.com/1/event/29/10+%20Deploys%20Per%20Day_%20Dev%20and%20Ops%20Cooperation%20at%20Flickr%20Presentation.pdf>. Acesso em 23-05-2017.

ARAKAKI, R.; RIBEIRO, A. L. D. **Gerenciamento de projetos tradicional x gerenciamento de projetos ágil: uma análise comparativa**. In: Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação - CONTECSI. 3., 2006, São Paulo. Anais... São Paulo: TECSI - Laboratório de Tecnologia e Sistemas de Informação FEA-USP, 2006. p. 1594- 1604. Disponível em: <<http://www.contecsi.fea.usp.br/envio/index.php/contecsi/3contecsi/paper/viewFile/2014/1128>>. Acesso em 07-01-2017.

ARAÚJO, L.; GAVA, R. **Empresas proativas: como antecipar mudanças no mercado**. Rio de Janeiro: Elsevier, 2011.

ATLASSIAN. **DevOps maturity model**. Austrália: Atlassian, 2017a.

ATLASSIAN. **DevOps: breaking the development-operations barrier**. Austrália: Atlassian, 2017b.

AXELOS Limited. **Glossary of terms english**. Brazilian Portuguese. v.1.0. United Kingdom, 2012.

BADRINARAYANAN, K.; *et al.* **It Ops & Devops productivity report 2013: tools, methodologies and people**. Krish Badrinarayanan, Jevgeni Kabanov, Ryan St. James e Oliver White. US: RebelLabs – ZeroTurnaround, 2013.

BON, J. **Guia de referência ITIL**. Rio de Janeiro: Elsevier, 2012.

BYRNE, B. M. **Structural equation modeling with AMOS: basic concepts, applications, and programming**. 2.ed. New York: Taylor & Francis Group, 2010.

CHIN, W.W. **Bootstrap cross-validation indices for PLS path model assessment**. In: ESPOSITO VINZI, V., CHIN, W.W., HENSELER, J. and WANG, H. (Eds), *Handbook of Partial Least Squares: Concepts, Methods and Applications*, Springer, Heidelberg, Dordrecht, London and New York, NY, 2010. p. 83-97.

- COHEN, J. **Statistical power analysis for the behavioral sciences**. 2. NJ: Erlbaum, 1988.
- COOPER, D. R.; SCHINDLER, P. S. **Métodos de pesquisa em administração**. 7. ed. Porto Alegre: Bookman, 2003. 640 p.
- DEEMER, P.; BENEFIELD, G.; LARMAN, C.; VODDE, B.. **The scrum primer**. v. 1.2, 2010. Disponível em: <<http://goodagile.com/scrumpriemer/scrumpriemer.pdf>>. Acesso em: 07-01-2017.
- DUVALL, P. M. **Continuous delivery patterns and antipatterns in the software lifecycle**. 2017. Disponível em: <<https://dzone.com/storage/assets/3989241-dzonerefcadz145-cdpatternsantipatterns.pdf>>. Acesso em 26-02-2017.
- ENGATES, J. **Building the DevOps mindset: agile collaboration at work**. **Rackspace blog**. August 25, 2014. Disponível em: <<https://blog.rackspace.com/building-the-devops-mindset-agile-collaboration-at-work>>. Acesso em 23-05-2017.
- FAUL, F.; *et al.* **Statistical power analyses using G*Power 3.1: tests for correlation and regression analyses**./ faul, f., erdfelder, e., buchner, a. e lang, A. G. Behavior Research Methods, v. 41, n. 4, 1149-1160, 2009.
- FERNANDES, A. A.; ABREU, F. A. I. **Governança de TI: da estratégia à gestão de processos e serviços**. 4. ed. Rio de Janeiro: Brasport, 2014.
- FLEMING, C. M.; BOWDEN, M. **Web-based surveys as an alternative to traditional mail methods**. Journal of Environmental Management, v.90, n.1, p.284-292, 2009.
- FORRESTER, A. **Accelerate your devops to awesome: learn how to take your DevOps to the next level**. Reino Unido: Forrester, 2016.
- _____. **Application delivery speed drives success: how mastering DevOps enables speed with quality and low cost**. Reino Unido: A Forrester Consulting Thought Leadership Paper Commissioned By HP, october 2015. Disponível em: <https://xact.spiceworks.com/u/gen/Jan2016/HPE_DevOps_Forrester_T2_Applicatn_Delivery_Speed_Drives_Success_FINAL_cea2dc931adfff32af47c43955e948cPDF>. Acesso em 07-01-2017.
- FOWLER, M. **Continuous delivery: reliable software releases through build, test, and deployment automation**. 2013. Disponível em: <<https://www.martinfowler.com/bliki/ContinuousDelivery.html#footnote-when>>. Acesso em Acesso em 07-01-2017.
- FRAGA, L. C. **Papel da tecnologia da informação e maturidade da gestão de TI**. 2015, 143f. Tese (Doutor em Administração) –Programa De Pós-Graduação em Administração - Universidade Nove de Julho (Uninove). São Paulo: UNINOVE, 2015. [Autor: Luiz Carlos Fraga e Silva Júnior].

FREITAS, M. A. S. **Fundamentos do gerenciamento de serviços de TI**. 2. ed. Rio de Janeiro: Brasport, 2013.

GHANTOUS, G. B.; GILL, A. **DevOps: concepts, practices, tools, benefits and challenges**. *In*: PACIFIC-ASIA CONFERENCE ON INFORMATION SYSTEMS - PACIS. 21. 2017, Langkawi-Malasia. Proceedings... Langkawi: Malaysia Chapter of the Association for Information Systems (MyAIS) and Department of Information Systems, School of Computing (SOC), Universiti Utara Malaysia (UUM). Disponível em: <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1191&context=pacis2017>. Acesso em 07-01-2017.

GÖTZ, O.; LIEHR-GOBBER, K.; KRAFFT, M. **Evaluation of structural equation models using the partial least squares (PLS) approach**. *In*: ESPOSITO VINZI, V., CHIN, W., HENSELER, J., WANG, H. (eds) Handbook of Partial Least Squares. Springer Handbooks of Computational Statistics. Springer, Berlin, Heidelberg, 2010. pp. 691-711.

HAIR, J. F.; HULT, T. M.; RINGLE, C. M.; SARSTEDT, M. **A primer on partial least squares structural equation modeling (PLS-SEM)**. Los Angeles: SAGE, 2014.
HASTIE, Shane; WOJEWODA, Stéphane. "Standish Group 2015 Chaos Report - Q&A" with Jennifer Lynch. InfoQ, on Oct 04, 2015. Disponível em: <<https://www.infoq.com/articles/standish-chaos-2015>>. Acesso em 23-05-2017.

HENSELER, J. *et al.* **The use of partial least squares path modeling in international marketing**./Jörg Henseler, Christian M. Ringle, Rudolf R. Sinkovics. *In*: SINKOVICS, Rudolf R.; GHOURI, Pervez N. (ed.) New Challenges to International Marketing. UK: Emerald Group Publishing Limited, 2009, p.v277 – 319. (Advances in International Marketing, Volume 20)

HENSELER, J. **On the convergence of the partial least squares path modeling algorithm**. Computational Statistics, v. 25 n. 1, pp. 107-120, 2010.

HPE - HEWLETT PACKARD ENTERPRISE. **Measuring DevOps success: how do you know devops is working? watch these KPIs**, rev. 2, August 2016. (Business white paper, 4AA6-3036ENN). Disponível em: <http://files.asset.microfocus.com/4aa6-3036/en/4aa6-3036.pdf>. Acesso em 23-05-2017.

HUMBLE, Jez; FARLEY, David. **Continuous delivery: reliable software releases through build, test, and deployment automation**. UK: Pearson Education, Jul 2010. (Addison-Wesley Signature Series)

HUTTERMANN, M. DevOps for Developers: integrate development and operations, The Agile Way. NY: Apress, 2012.

KIM, G. *et al.* **The DevOps handbook: how to create world-class agility, reliability, and security in technology organization**. Jez Humble, Patrick Debois, and John Willis. US: IT Revolution Press, 2016. 480 p.

LAUDON, K. C.; L., Jane P. **Management Information Systems: managing the digital firm**, 13th Edition. England: Pearson, 2014.

LEVIN, J.; FOX, J.A.; FORDE, D. R. **Estatística para ciências humanas**. 11. ed. São Paulo: Pearson Education do Brasil, 2012.

MARÔCO, J. **Análise de equações estruturais: fundamentos teóricos, software & aplicações**. Pêro Pinheiro (PT): Report Number; 2010.

MUELLER, Ernest. **What is DevOps? The agile admin**. Dezembro, 2016. Disponível em: <http://theagileadmin.com/what-is-DevOps/>. Acesso em: 22/01/2017.

NERUR, S.; MAHAPATRA, R.; MANGALARAJ, G. **Challenges of migrating to agile methodologies: organizations must carefully assess their readiness before treading the path of agility**. ACM, v. 48 n. 5, p. 72-78. May/2005.

OLSEN, S. B. **Choosing between internet and mail survey modes for choice experiment surveys considering non-market goods**. Environmental and Resource Economics, v. 44, issue 4, p.591-610, Dec. 2009.

PATWARDHAN, A. S. *et al.* **Embracing Agile methodology during DevOps Developer Internship Program**. NY: Cornell University Library. 2016. Disponível em: [ehttps://arxiv.org/ftp/arxiv/papers/1607/1607.01893.pdf](https://arxiv.org/ftp/arxiv/papers/1607/1607.01893.pdf). Acesso em 23-05-2017. PEREIRA, Steve. The DevOps CheckList, 2017. Disponível em: <http://DevOpschecklist.com/>. Acesso em 23-05-2017. Acesso em: 13/01/2017.

PEPIN, C. **An introduction to DevOps**. Chris Pepin dot com [bolg], January 06, 2014. Disponível em: <http://www.chrispepin.com/2014/01/an-introduction-to-devops.html>. Acesso em 23-05-2017.

PMI. **Um guia do conhecimento em gerenciamento de projetos: guia PMBOK® 2013**. Quinta Edição. USA: Project Management Institute - PMI®, 2013.

PRESSMAN, Roger S. **Engenharia de software [recurso eletrônico]: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011.

PUPPET. **Getting Started with DevOps**. Puppet. Inc, 2016a. Disponível em: <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>. Acesso em 23-05-2017.

_____. **State of DevOps Report**. Puppet Inc, 2016. Disponível em: <https://puppet.com/resources/whitepaper/2016-state-of-devops-report>. Acesso em 23-05-2017.

_____. **State of DevOps report**. Puppet Inc, 2017. Disponível em: <https://puppet.com/blog/2017-state-devops-report-here>. Acesso em 23-05-2017.

RECHZIEGEL, Waldir. **Gestão da inovação aberta: Modelo de seleção de inovações tecnológicas**. 2014. 120 f. Tese (Doutorado em Administração) - Universidade Nove de Julho, São Paulo, 2014.

RICHARDSON, R. J.; PERES, J. A. S.; WANDERLEY, J.C.V.; CORREIA, L. M.; PERES, M.H. **Pesquisa social: métodos e técnicas**. 3ed. São Paulo: Atlas, 1999.

RINGLE, C. M.; SILVA, D.; BIDO, Diógenes. **Modelagem de equações estruturais com utilização do SmartPLS. Remark.** Revista Brasileira de Marketing, São Paulo, v. 13, n. 2, p.56, maio 2014. Disponível em: <http://www.revistabrasileiramarketing.org/ojs-2.2.4/index.php/remark/article/view/2717/pdf_167>. Acesso em 23-05-2017.

ROYCE, W. W. **Managing the development of large software systems: concepts and techniques.** Proc. IEEE WESTCON, Los Angeles, August 1970. Disponível em: <<http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/Waterfall.pdf>>. Acesso em 07-01-2017.

SATO D. **DevOps na prática: entrega de software confiável e automatizada.** S/l. Casa do Código, 2014.

SCHWABER, K.; SUTHERLAND, J. **The definitive guide to scrum: the rules of the game.** Scrum guides [site], July 2016. Disponível em: <<http://www.scrumguides.org/history.html>>. Acesso em: 23-05-2017.

SCRUMstudy. Scrum vs. **Traditional Project Management** [site], July 2014. Disponível em: <<https://medium.com/@SCRUMstudy/scrum-vs-traditional-project-management-60f15f1e63f>>. Acesso em 12-11-2017.

SELLTIZ, C.; WRIGHTSMAN, L.; COOK, S.; KIDDER, L. **Métodos de pesquisa nas relações sociais.** São Paulo: EDUSP, 1975.

SHARMA, S; COYNE, B. **Devops for dummies, 2nd.** IBM Limited Edition. USA: John Wiley & Sons, 2015.

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

SOULA, Jose Maria Fiorino. **ISO/IEC 20000: gerenciamento de serviços de tecnologia da informação.** São Paulo: Brasport, 2013. 432p.

SUTHERLAND, Jeff; SCHWABER, Ken. **The Scrum Papers: nut, bolts, and origins of an agile framework.** Versão 1.1. Cambridge: Scrum Inc, 2012. <http://jeffsutherland.org/scrum/ScrumPapers.pdf>. Acesso em 07-01-2017.

TABACHNICK, B. G.; FIDELL, L. **Using multivariate statistics,** 3rd ed. Boston: Pearson, 2007.

TENENHAUS, M.; VINZI, V. E.; CHATELIN, Y. M.; LAURO, C. **PLS path modeling. computational statistics & data analysis,** v.48, n.1, p. 159-205, 2005.

VARGAS, R. V. **Gerenciamento de projetos: estabelecendo diferenciais competitivos.** 8. ed. Rio de Janeiro: Brasport, 2016.

WETZELS, M.; ODEKERKEN-SCHRODER, G.; VAN OPPEN, C. **Using pls path modeling for assessing hierarchical construct models: Guidelines and Empirical Illustration.** MIS Quarterly, v.33, n.1, p.177-195. 2009.

ZWICKER, R.; SOUZA, C.A.; BIDO, D. S. **Uma revisão do modelo do grau de informatização de empresas:** novas propostas de estimação e modelagem usando PLS (Partial Least Squares). In: ENCONTRO DA ASSOCIAÇÃO NACIONAL DE PÓS-GRADUAÇÃO E PESQUISA EM ADMINISTRAÇÃO, 32, 2008, Rio de Janeiro. Anais... Rio de Janeiro: Anpad, 2008.

APÊNDICE A - Questionário da pesquisa

Questionário para profissionais de Tecnologia da Informação sobre as principais Habilidades/Capacidades necessárias para a prática do <i>DevOps</i>
<p>O <i>DevOps</i> é um conjunto emergente de princípios, métodos e práticas, baseado em muitos dos valores fundamentais do Agile. Ele visa proporcionar maior agilidade nas respostas às necessidades empresariais dinâmicas, ao encolhimento dos orçamentos de TI e às expectativas de qualidade, segurança e satisfação do cliente. Esta pesquisa visa identificar como as capacidades necessárias para a prática do <i>DevOps</i> são percebidas pelos profissionais de TI.</p>
<p>Como se deve responder ao questionário?</p>
<p>As respostas devem variar entre zero e dez, da seguinte forma: 0 = Discordo totalmente da afirmativa 10 = Concordo totalmente com a afirmativa.</p> <p>Pontos importantes para as respostas:</p> <ul style="list-style-type: none"> - Considere a experiência que você acumulou ao longo de sua carreira ao responder cada pergunta. NÃO considere apenas sua posição ou emprego atual. - Considere o que você realmente vivenciou nas organizações onde trabalhou. NÃO responda com a pontuação que você considera ideal. - NÃO considere sua melhor ou pior experiência para responder uma questão, mas sua experiência geral.

ANEXO 1- Aprovação do comitê de ética em pesquisa



PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: DevOps: Habilidades e Capacidades Necessárias para sua utilização na área de Tecnologia da Informação.

Pesquisador: GUSTAVO TRAVERSIN

Área Temática:

Versão: 1

CAAE: 71676917.8.0000.5501

Instituição Proponente: Universidade de Taubaté

Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer: 2.191.804

Apresentação do Projeto:

Este trabalho busca responder a seguinte questão de pesquisa: Quais as habilidades e competências necessárias para a adoção e prática de DevOps no desenvolvimento e entrega de projetos na área de tecnologia da informação?

Objetivo da Pesquisa:

Analisar os principais modelos de gerenciamento de projetos e operações, com enfoque em DevOps para identificar as habilidades e capacidades necessárias para sua utilização.

Avaliação dos Riscos e Benefícios:

Segundo a Resolução 510/16 toda pesquisa oferece risco ao participante de pesquisa, mesmo que seja mínimo portanto atualizar a informação no campo do formulário básico da pesquisa.

Comentários e Considerações sobre a Pesquisa:

Contribuir com o estudo para a identificação e análise das principais habilidades e capacidades necessárias para que o modelo DevOps.

Considerações sobre os Termos de apresentação obrigatória:

Adequado.

Recomendações:

O Comitê de Ética em Pesquisa da Universidade de Taubaté recomenda a entrega do relatório final

Endereço: Rua Visconde do Rio Branco, 210
Bairro: Centro **CEP:** 12.020-040
UF: SP **Município:** TAUBATE
Telefone: (12)3635-1233 **Fax:** (12)3635-1233 **E-mail:** cepunitau@unitau.br



Continuação do Parecer: 2.191.804

ao término da pesquisa.

Conclusões ou Pendências e Lista de Inadequações:

Proposta de pesquisa adequada.

Considerações Finais a critério do CEP:

O Comitê de Ética em Pesquisa da Universidade de Taubaté, em reunião realizada no dia 28/07/2017, e no uso das competências definidas na Resolução CNS/MS 510/16, considerou o Projeto de Pesquisa: APROVADO.

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BASICAS_DO_PROJETO_964350.pdf	24/07/2017 08:46:36		Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	TCLE.pdf	24/07/2017 08:44:34	GUSTAVO TRAVERSIN	Aceito
Folha de Rosto	FolhaDeRosto.pdf	23/07/2017 21:50:09	GUSTAVO TRAVERSIN	Aceito
Projeto Detalhado / Brochura Investigador	Projeto_Monografia_DevOps_Gustavo_Traversin_R7.pdf	23/07/2017 21:41:58	GUSTAVO TRAVERSIN	Aceito
Brochura Pesquisa	Questionario_Habilidades_DevOps.pdf	23/07/2017 21:40:59	GUSTAVO TRAVERSIN	Aceito

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:

Não

TAUBATE, 28 de Julho de 2017

Assinado por:
José Roberto Cortelli
(Coordenador)

Endereço: Rua Visconde do Rio Branco, 210
Bairro: Centro **CEP:** 12.020-040
UF: SP **Município:** TAUBATE
Telefone: (12)3635-1233 **Fax:** (12)3635-1233 **E-mail:** cepunitau@unitau.br

Autorizo cópia total ou parcial desta obra, apenas para fins de estudo e pesquisa, sendo expressamente vedado qualquer tipo de reprodução para fins comerciais sem prévia autorização específica do autor.

Gustavo Traversin

Taubaté, 23 de outubro de 2017.