



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Proposta de metodologia de gestão de riscos para
projetos ágeis de software no Instituto Nacional de
Estudos e Pesquisas Anísio Teixeira (INEP)**

Elizabethte Caldas Ferreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Orientadora
Prof.^a Dr.^a Ana Carla Bittencourt Reis

Brasília
2017

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

C EL43p CALDAS FERREIRA, ELIZABETTE
Proposta de metodologia de gestão de riscos para
projetos ágeis de software no Instituto Nacional de Estudos
e Pesquisas Anísio Teixeira (INEP) / ELIZABETTE CALDAS
FERREIRA; orientador Ana Carla Bittencourt Reis. --
Brasília, 2017.
132 p.

Dissertação (Mestrado - Mestrado Profissional em
Computação Aplicada) -- Universidade de Brasília, 2017.

1. Gestão de Riscos. 2. Metodologias ágeis. 3. Risco. 4.
Desenvolvimento de software. 5. Scrum. I. Bittencourt Reis,
Ana Carla , orient. II. Título.



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

**Proposta de metodologia de gestão de riscos para
projetos ágeis de software no Instituto Nacional de
Estudos e Pesquisas Anísio Teixeira (INEP)**

Elizabethte Caldas Ferreira

Dissertação apresentada como requisito parcial para conclusão do
Mestrado Profissional em Computação Aplicada

Prof.^a Dr.^a Ana Carla Bittencourt Reis (Orientadora)
EPR/CIC/UnB

Prof.^a Dr.^a Simone Borges Simão Monteiro
EPR/CIC/UnB

Prof. Dr. Edilson Feredá
Universidade Católica de Brasília - UCB

Prof. Dr. Marcelo Ladeira
Coordenador do Programa de Pós-graduação em Computação Aplicada

Brasília, 26 de Julho de 2017

Dedicatória

Dedico este mestrado, primeiramente a Deus, por torna-lo possível.

Dedico a meu único filho, razão da minha existência, pelos momentos em que tive que sacrificar nossa convivência. Igualmente o dedico a meu marido por total incentivo, entendimento e apoio desde seu início até hoje.

Dedico também aos meus pais pelo dom da vida.

E ainda, dedico este mestrado a todos os inconformados, assim como eu, para que de forma positiva consigam mudar a própria vida e o meio em que vivem.

Agradecimentos

Agradeço ao INEP, por me ceder as informações necessárias para a elaboração desta dissertação, reconhecendo a importância e atualidade do tema para a melhoria contínua dos processos internos.

Agradeço a minha orientadora que me auxiliou de forma serena na concretização das ideias, pela condução e acompanhamento do processo de elaboração deste trabalho.

Agradeço ao PPCA (Programa de Pós-Graduação em Computação Aplicada), pela qualidade do curso ministrado e relevância do tema na Administração Pública do País.

Resumo

Visando atender as necessidades finalísticas do INEP, a Diretoria de Tecnologia e Disseminação de Informações Educacionais (DTDIE) desenvolve sistemas censitários e avaliativos para levantamento de dados que posteriormente possam gerar indicadores a fim de subsidiar a formulação de políticas públicas educacionais [46].

Este trabalho trata-se de estudo de caso realizado com base nos dados dos sistemas da Coordenação Geral de Sistemas de Informação (CGSI) subordinada a DTDIE. Analisando quantitativamente os dados dos sistemas da CGSI, foi possível diagnosticar que aproximadamente 48% de todas as ordens de serviços (OSs) de desenvolvimento de software foram entregues em atraso. Para entender as possíveis causas destes atrasos, foi realizada análise de causas raízes, por meio da FTA (Análise da Árvore de Falhas). Neste contexto, somando-se às recomendações por órgãos de controle sobre a adoção da gestão de riscos, avistou-se a necessidade de se propor uma metodologia de gestão de riscos com vistas a aumentar a perspectiva de sucesso dos projetos.

A elaboração da metodologia de gestão de riscos propõe uma integração ao ciclo de desenvolvimento ágil de software adotado na coordenação. Para isso, foi desenvolvido o processo da gestão de riscos, alinhado à norma ISO/IEC 31000:2009 em conjunto com as recomendações constantes na instrução normativa INC MP/CGU Nº 01/2016, integrada às etapas de desenvolvimento de software abarcadas no *framework Scrum* [93] [64]. Assim, foram analisados os modelos mais atuais de integração da gestão de riscos e métodos ágeis, por fim, dando origem a um modelo próprio. Esta metodologia também define um fluxograma evidenciando quais ferramentas poderão ser utilizadas em cada atividade da gestão de riscos. Ainda, foi desenvolvido um *checklist* de riscos comuns em projetos de software, para auxiliar na atividade de identificação dos riscos. Também foi elaborada proposta de *template* de Relatório de Gestão e Comunicação dos Riscos, que contemple todo o gerenciamento e monitoramento, podendo ser customizado dentro de software de gestão de riscos.

Palavras-chave: Gestão de Riscos, Metodologias Ágeis, Riscos, Desenvolvimento de Software, Scrum

Abstract

In order to meet the final needs of INEP, the Educational Information Technology and Dissemination Board (DTDIE) develops census and evaluation systems for data collection that can later generate indicators to subsidize the formulation of educational public policies [46].

This work is a case study based on data from the General Information System Coordination (CGSI) systems under DTDIE. This General Coordination is directly responsible for the development, maintenance and support of INEP's final systems. By quantitatively analyzing data from CGSI systems, it was possible to diagnose that approximately 48 % of all software development service orders (OSs) were delivered behind schedule. In order to understand the possible causes of these delays, a root cause analysis was performed through the Fault Tree Analysis (FTA). In this context, risk management is highly recommended because it presents risk mitigation actions as a way to prevent them from materializing and can increase the probability of reaching institutional goals. That is the reason the adoption of risk management has been widely recommended by the government controlling bodies.

The proposed risk management methodology suggests an integration to the cycle of agile software development adopted. For this purpose, the risk management process was developed according to ISO / IEC 31000: 2009 and the recommendations of Normative Instruction INC MP / CGU N° 01/2016 and totally integrated to the Scrum Framework software development stages [93] [64]. Prior to the model definition, most current models of risk management and agile methods integration were analyzed. Proposed Methodology also defines a flow chart showing tools that could be used in each risk management activity. Also, a checklist of common risks in software projects was developed to assist the risk identification activity. Finally, a template for the Risk Management and Communication Report is presented. This document includes all management and monitoring activities and could be used to provide transparency to risks.

Keywords: Risk management, Agile methodology, Risks, Software Development, Scrum

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Problemática	2
1.3	Justificativa	3
1.4	Objetivo	5
1.4.1	Objetivo Geral	5
1.4.2	Objetivos Específicos	5
1.5	Contribuição Esperada	5
1.6	Estruturação dos Capítulos	6
2	Base Conceitual e Revisão da Literatura	7
2.1	Modelos de Processos de Desenvolvimento de Software	7
2.1.1	Modelos de Processos Prescritivos	7
2.1.2	Metodologias Ágeis	13
2.2	Gestão de Projetos	25
2.2.1	PMBOK - Project Management Body of Knowledge (Corpo de Co- nhecimento em Gerência de Projetos)	25
2.3	Riscos	27
2.4	Gestão de Riscos	28
2.4.1	ABNT NBR ISO/IEC 31000	28
2.4.2	ABNT NBR ISO/IEC 31010	31
2.4.3	ABNT NBR ISO/TR 31004	34
2.5	Riscos em Metodologias Ágeis	34
2.6	Gestão de Riscos em Projetos de Desenvolvimento de Software	37
2.7	Técnicas e Ferramentas da Gestão de Riscos Aplicadas a Projetos de Software	39
2.8	Modelos de Gestão de Riscos aplicados às Metodologias Ágeis	41
2.8.1	Modelo de Gestão de Riscos Segundo Andrat e Jaswal (2015)	41
2.8.2	Modelo de Gestão de Riscos Segundo a Extensão de Software do PMI - 5º Edição	43

2.8.3	Modelo de Gestão de Riscos Segundo Popli e Naresh (2013)	43
2.8.4	Modelo de Gestão de Riscos Segundo Cunha et al. (2013)	44
2.8.5	Modelo de Gestão de Riscos Segundo Khatri et al.(2014)	45
2.9	Legislação Governamental em Gestão de Riscos	49
2.9.1	Avaliação da Governança de TI na Administração Pública Federal-TC 003.732/2014-2	49
2.9.2	Conhecimento acerca da Utilização de Métodos Ágeis nas Contratações para Desenvolvimento de Software pela Administração Pública Federal-TC 010.663/2013-4	50
2.9.3	Instrução Normativa Conjunta MP/CGU Nº 01/2016	52
2.9.4	Metodologia de Gestão de Riscos de Segurança da Informação e Comunicações - SISP	54
3	Metodologia de Pesquisa	57
3.1	Método da Pesquisa	57
3.2	Estruturação da Pesquisa	59
4	Estudo de Caso: Diagnóstico da Situação Atual	62
4.1	Contexto Externo	62
4.1.1	INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira	62
4.2	Contexto Interno	67
4.2.1	Diretoria de Tecnologia e Disseminação de Informações Educacionais (DTDIE)	67
4.2.2	Plano Diretor de Tecnologia da Informação - PDTI	69
4.2.3	Coordenação Geral de Sistemas de Informações (CGSI)	73
4.3	Análise da Situação dos projetos de software da CGSI	75
4.3.1	Análise do Catálogo de Sistemas	75
4.3.2	Análise das Ordens de Serviços dos Projetos de Software	80
4.3.3	Análise de Causa Raiz por meio da Análise de Árvore de Falhas (FTA)	84
4.3.4	Impacto do Retrabalho na Cadeia de Valor e Recomendações Finais	88
5	Proposta de Metodologia de Gestão de Riscos da CGSI	91
5.1	Contextualização	91
5.2	Conceitos	92
5.3	Nível de Atuação da Metodologia	95
5.4	Objetivo	95

5.5	Papéis	96
5.6	Proposta para processo de gestão de riscos integrada ao <i>Scrum</i>	97
5.6.1	Mapeamento da Gestão de Riscos no <i>framework Scrum</i>	99
5.7	Fluxograma das Atividades da Gestão de Riscos com Técnicas e Ferramentas a serem Utilizadas	101
5.7.1	Método de Identificação de Riscos	103
5.7.2	Método de Análise e Avaliação de Riscos	106
5.7.3	Tratamento e Planejamento de Resposta aos Riscos	110
5.7.4	Monitoramento do Risco	112
5.7.5	Comunicação dos Riscos	114
5.7.6	Análise Crítica	117
5.7.7	Melhoria Contínua	118
6	Conclusão	120
	Referências	122
	Apêndice	129
A	Lista/<i>checklist</i> de Riscos - CGSI	130
B	Relatório de Gestão e Comunicação de Riscos	132

Lista de Figuras

2.1	Modelo Cascata. Fonte: [79].	8
2.2	Modelo Incremental. Fonte: [79].	9
2.3	Prototipação. Fonte: [79].	10
2.4	Modelo Espiral. Fonte: [79].	11
2.5	Modelo Unificado. Fonte: [79].	13
2.6	Framework Scrum. Fonte:[81].	17
2.7	Gráfico de métodos ágeis. Fonte:[99].	19
2.8	Extreme Programming-XP (Programação Extrema). Fonte:[79].	20
2.9	Kanban. Fonte:[4].	23
2.10	Processo da gestão de riscos. Fonte: Adaptado da ABNT NBR ISO/IEC 31000:2009.	29
2.11	Pirâmide de Risco. Fonte: [8].	42
2.12	Diagrama de Rede dos Riscos. Fonte: [8].	42
2.13	Gestão de Riscos para Métodos Adaptativos. Fonte: Adaptado de PMI, IEEE Computer Society (2013)	43
2.14	Processo de gestão de risco ágil. Fonte: [28].	44
2.15	Diagrama de fluxo do Processo da gestão de risco ágil. Fonte: [51].	45
2.16	Nível de abstração no processo Scrum para análise e documentação do risco. Fonte: [51].	46
2.17	Processo da MGR-SISP. Fonte: [61]	55
2.18	Critério de classificação para o tratamento e aceitação de riscos da MGR- SISP. Fonte: [61]	55
3.1	Caracterização do método de pesquisa. Fonte: Elaboração própria.	58
3.2	Estrutura da Pesquisa. Fonte: Elaboração própria.	60
4.1	Organograma INEP. Fonte: Adaptado do portal INEP [46].	68
4.2	Percentual dos Módulos de Sistemas por Situação. Fonte: Elaboração pró- pria [31].	76
4.3	Percentual de Módulos por Área de Negócio. Fonte: Elaboração própria [31].	77

4.4	Percentual de Módulos por Tecnologia. Fonte: Elaboração própria [31].	78
4.5	Percentual de Módulos de Sistemas por Diretoria. Fonte: Elaboração própria [31].	78
4.6	Quantidade de Módulos por Sistema. Fonte: Elaboração própria [31].	79
4.7	Percentual do tipo de Ordens de Serviço. Fonte: Elaboração própria.	82
4.8	Situação das OSs entregues. Fonte: Elaboração própria.	83
4.9	Tipos de avaliações a serem submetidas cada tipo de OS. Fonte: Elaboração própria.	85
4.10	Análise da Árvore de Falhas - Analisa as causas dos atrasos das OSs. Fonte: Elaboração própria.	87
4.11	Ciclo da Ordem de Serviço. Fonte: Elaboração própria.	89
5.1	Proposta de processo de gestão de riscos integrada ao <i>Scrum</i> . Fonte: Elaboração Própria.	98
5.2	Fluxograma proposto para a Metodologia de Gestão de Riscos da CGSI. Fonte: Elaboração própria.	102
5.3	Subprocesso Identificar Riscos. Fonte: Elaboração própria.	105
5.4	Subprocesso Analisar e Avaliar Riscos. Fonte: Elaboração própria.	107
5.5	Subprocesso Tratar e Planejar Resposta aos Riscos. Fonte: Elaboração própria.	111
5.6	Subprocesso Monitorar Riscos. Fonte: Elaboração própria.	113
5.7	Subprocesso Comunicar Riscos. Fonte: Elaboração própria.	115
5.8	Quadro <i>Scrum</i> com controle dos Riscos da <i>Sprint</i> . Fonte: Elaboração própria.	116
5.9	Subprocesso Análise Crítica. Fonte: Elaboração própria.	117
5.10	Subprocesso Melhoria Contínua. Fonte: Elaboração própria.	119

Lista de Tabelas

4.1	Classificação de prazo estimado x executado por tipo de OS. Fonte: Elab- oração própria	84
4.2	Cluster de prazo estimado x executado por diretoria. Fonte: Elaboração própria	90

Lista de Quadros

2.1	Checklist de fatores àgeis Scrum e métodos tradicionais. Fonte: [88]	26
2.2	Ferramentas utilizadas para o processo de avaliação de riscos. Fonte: [3]. (continua)	32
2.3	Riscos comuns de Software. Adaptado de Albadarneh et al. (2015) [6]	36
2.4	Técnicas e Ferramentas Utilizadas nos Modelos de Gestão de Riscos inte- grados ao Desenvolvimento de Software. Fonte: Elaboração própria.	39
2.5	Comparação entre métodos ágeis e tradicionais. Fonte: [8]	41
2.6	Consolidação dos Modelos de Gestão de Riscos integrados aos Métodos Ágeis. Fonte: Elaboração própria	48
4.1	Objetivos Estratégicos de TI. Fonte: [30]	70
5.1	Mapeamento da Gestão de Riscos no <i>Scrum</i> . Fonte: Adaptado de Nyfjord, et al. (2008)[69]	100
5.2	Critério de classificação de Nível de riscos. Fonte: Adaptado da MGR-SISP [61]	108

Lista de Siglas

ANASEM - Avaliação Nacional Seriada dos Estudantes de Medicina
APF - Administração Pública Federal
AUP - Processo Unificado Ágil
CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CCP - Controle de Contratos e Projetos
CGSI - Coordenação Geral de Sistemas de Informação
CGU - Corregedoria Geral da União
CRM - Gerenciamento de risco contínuo
DAEB - Diretoria de Avaliação da Educação Básica
DAES - Diretoria de Avaliação da Educação Superior
DEED - Diretoria de Estatísticas Educacionais
DGP - Diretoria de Gestão e Planejamento
DIREC - Diretoria de Estudos Educacionais
DTDIE - Diretoria de Tecnologia e Disseminação de Informações Educacionais
DSDM - Método de Desenvolvimento de Sistemas Dinâmicos
ENCCEJA - Exame Nacional Para Certificação de Competências
ENEM - Exame Nacional do Ensino Médio
FDD - Feature Driven Development (Desenvolvimento guiado por funcionalidade)
FNDE - Fundo Nacional de Desenvolvimento da Educação
FTA - Análise de Árvore de Falhas (FTA)
IEEE - Institute of Electrical and Electronic Engineers
INC - Instrução Normativa Conjunta
INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
MEC - Ministério da Educação
MGDS - Metodologia Geral de Desenvolvimento de Sistemas
MP - Ministério do Planejamento Orçamento e Gestão
MGR - Metodologia de Gestão de Riscos
OMG - Object Management Group
OS - Ordem de Serviço
PHP - Hypertext Preprocessor

PDTI - Plano Diretor de Tecnologia da Informação
PMBOK - Project Management Body of Knowledge (Corpo de conhecimento em gerenciamento de projeto)
PMI - Project Management Institute (Instituto de Gerenciamento de Projeto)
PNE - Plano Nacional de Educação
PF - Pontos de Função
PO - Product Owner (dono do produto)
PROFIP - Programa de Fomento à Integridade Pública
RBEP - Revista Brasileira de Estudos Pedagógicos
SAEB - Sistema Nacional de Avaliação da Educação Básica
SEDIAE - Secretaria de Avaliação e Informação Educacional
SEEC - Serviço de Estatística da Educação e Cultura
SEI - Software Engineering Institute (Instituto de Engenharia de Software)
SIC - Segurança da Informação e Comunicações
SISP - Sistema de Administração de Recursos da Tecnologia da Informação
SWOT - Strengths, Weaknesses, Opportunities, e Threats (Forças, fraquezas, oportunidades e ameaças)
TCU - Tribunal de Contas da União
TIC - Tecnologia da Informação e Comunicação
WIP - Work in Progress (Trabalho em progresso/andamento)
XP - *Extreme Programming (Programação extrema)*

Capítulo 1

Introdução

1.1 Contextualização

O desenvolvimento de software é caracterizado por prazos apertados, contínuas modificações e por uma ênfase na satisfação do cliente-usuário. Em muitos casos o requisito mais importante é o tempo de colocação de um software no mercado. Se este momento oportuno de entrada for perdido, o projeto pode até perder o sentido [79]. Trazendo este conceito para a esfera do governo, inúmeros softwares são criados para atender programas e metas governamentais. Na maioria das vezes, estes sistemas possuem prazos definidos em leis, portarias ou decretos, para estarem disponíveis, sendo este, um requisito não funcional legal e até uma restrição do projeto.

Adicionalmente, existem incertezas, as quais todas as organizações estão sujeitas. Há influências de fatores externos e internos que torna incerto se, quando e em que extensão estas organizações atingirão ou excederão, os seus objetivos [2]. Uma das características de gestão das áreas de Tecnologia da Informação e Comunicação (TIC) é a alta complexidade, oriunda da elevada especialização dos profissionais dessa área e da extrema dependência que a maioria das instituições possuem na continuidade dos serviços disponibilizados para que a área fim alcance seus objetivos. Incidentes de interrupções ou indisponibilidade nos serviços, como: internet, sistemas críticos/finalísticos e servidores de banco de dados, podem provocar fortes reações por parte dos clientes e usuários finais, vindo a acarretar possíveis danos a imagem da instituição [78]. Neste cenário de prazos apertados, ênfase na satisfação do cliente, alta demanda por disponibilidade de serviços, carência de mão de obra especializada, atendimento à prazos legais e principalmente incertezas internas (por exemplo: alta rotatividade de pessoal) e externas (por exemplo: mudanças políticas ou econômicas), a gestão de riscos tem se tornado cada vez mais necessária e começa a fazer parte de várias recomendações de órgãos de controle tais como o Tribunal de Contas da União (TCU) e a Controladoria-Geral da União (CGU). O processo de gestão de riscos

tem o objetivo de controlar de forma contínua, os riscos que surgem em todas as fases dos projetos, sendo considerado determinante no sucesso destes. O processo de gestão de riscos é definido através de modelos que especificam as atividades a realizar durante o projeto, com o objetivo de eliminar ou minimizar o impacto dos riscos [28]. Há uma preocupação governamental para que controles internos sejam criados a fim de mitigar os principais riscos capazes de fazerem os órgãos não alcançarem seus objetivos. Além disto, há no setor público, um incentivo para à implantação do Programa de Fomento à Integridade Pública (PROFIP). Este programa se preocupa basicamente com riscos ligados à integridade organizacional das instituições públicas, como: atos de corrupção, abuso de poder, conflitos de interesses, nepotismo e desvios de recursos. Todos estes problemas podem resultar em prejuízos financeiros e danos à imagem dessas instituições. A partir deste programa, é feita a adequação das políticas e diretrizes de integridade já existentes no órgão em questão e são levantados riscos específicos da organização, por meio da construção de planos de integridade. Para isso, o processo de gestão de riscos, objeto deste trabalho, é a base desta construção e deverá estar alinhado com a missão e os objetivos estratégicos do órgão [64].

Percebe-se então que o tema gestão de riscos é bastante atual e tende a se difundir nos órgãos da Administração Pública Federal (APF), seja por recomendações de órgãos auditores, seja por normativos que obriguem a sua adoção ou por meio de fomento à participação em programas de integridade. O fato é que a gestão de riscos se tornou uma necessidade e portanto, precisa ser elaborada de forma customizada para que consiga alcançar a efetividade nas organizações.

1.2 Problemática

No Instituto Nacional de Estudos e Pesquisas Anísio Teixeira (INEP), autarquia vinculada ao Ministério da Educação (MEC), o nível de incerteza devido a mudanças no contexto externo, pode ser impactado por alterações no cenário político e econômico, bem como alterações no Plano Nacional da Educação (PNE). Estas alterações, podem fazer com que o grau de incerteza do órgão, em relação ao atingimento de suas finalidades, se eleve.

No contexto interno existem outras incertezas, que neste trabalho são abordadas especificamente no âmbito da CGSI, que possui papel preponderante para que o INEP atinja suas finalidades. A CGSI é responsável por criar e manter sistemas de informação e documentação que abranjam avaliações, exames e censos, gerando estatísticas para subsidiar as políticas educacionais [45]. Para isso, suas atribuições devem ser efetivamente alcançadas, dentre elas: a realização de exames, avaliações e coleta de dados educacionais da

educação básica e superior tanto das instituições na esfera pública quanto privada de todo o país. Tal levantamento subsidia outras diretorias deste órgão na realização de formulação dos indicadores educacionais úteis para traçar um panorama nacional da educação e assim identificar pontos a serem melhorados, investidos e acompanhados, por meio de formulação de políticas públicas pelo governo, assim como execução de programas na área de educação e transferência de recursos para merenda, transporte escolar, livros, dentre outros [46].

Atualmente, a CGSI possui fábrica de *software* terceirizada alocada em sua coordenação. Solicitações à fábrica são realizadas por meio de emissão de Ordens de Serviços (OSs). Após a entrega de uma unidade funcional de *software*, o gerente de projetos abre demanda de avaliação, para aferir a qualidade do que foi entregue. A CGSI adota uma metodologia própria intitulada de Metodologia Geral de Desenvolvimento de Sistemas (MGDS), baseada nas metodologias ágeis: *Scrum*, XP e *Kanban* [47].

Apesar das metodologias ágeis, serem em princípio dirigidas a riscos, por sua natureza iterativa e incremental, possibilitando testes e integrações constantes, os modelos de desenvolvimento ágeis ainda carecem de práticas efetivas e formais para o gerenciamento de riscos. Por conseguinte, há uma lacuna a ser preenchida, levando em conta o fato de que o gerenciamento de riscos é considerado uma boa prática na engenharia de *software* contemporânea e que não vem sendo realizado ou tem sido gerenciado de forma insatisfatória na maioria das organizações da APF, conforme relatório de levantamento da governança de TI realizado em 2014 [68] [1] [93].

De modo geral, os principais fatores que contribuem para a problemática atual são:

- a) O fato dos atrasos nos projetos de software serem pouco tolerados, uma vez que os projetos estruturantes (censos, avaliações e exames) possuem prazos legais que já estão pré definidos em leis, decretos ou portarias;
- b) Alta volatilidade dos requisitos, uma vez que, anualmente várias alterações negociais são realizadas nos projetos a pedido da área demandante;
- c) Eventuais instabilidades nas aplicações, devido as manutenções constantes, aumentando a probabilidade de inserção de *bugs*;
- d) Potencial dano à imagem do órgão.

1.3 Justificativa

Ainda que sejam adotados padrões, metodologias e *frameworks* de gerenciamento de projetos, os projetos, em sua maioria, ainda são mais propensos a falharem do que serem

bem sucedidos [11]. Conseqüentemente, a gestão de riscos tem tomado espaço no cenário de aumento das taxas de sucesso dos projetos [11].

Segundo a ISO 31004 (2015), a gestão de riscos é parte integrante de todos os processos organizacionais, ajuda os tomadores de decisões a fazerem escolhas informadas, priorizarem ações e distinguirem entre cursos alternativos de ação. Assim, a adoção da gestão de riscos integrada ao processo ágil de desenvolvimento de sistemas da CGSI se justifica pelo fato de aumentar a perspectiva de sucesso dos projetos por meio do aumento da probabilidade de entregas no prazo e também evita desgastes desnecessários à imagem do órgão, diminuindo a incerteza sobre o alcance dos objetivos finalísticos do instituto.

Concomitante a isto, os projetos estruturantes do órgão, que hoje são demandados para a CGSI já com restrições de prazo, teriam seus riscos geridos e monitorados, uma vez que, anualmente, estes projetos sofrem inúmeras alterações negociais, o que aumenta a volatilidade dos requisitos e também a probabilidade de inserção de novos *bugs*, instabilidades na aplicação e não atendimento à conformidade legal, relacionada ao prazo da disponibilização do sistema na web.

Visto isto, este trabalho se justifica pela importância das ações a seguir:

- a) A importância dos projetos de software estruturantes (avaliações, exames e censos) para o alcance direto da finalidade do INEP;
- b) Adoção da gestão de riscos nos projetos da CGSI, com foco no aumento da taxa de sucesso destes projetos;
- c) Melhoria gradual nos processos internos da CGSI, visto que a adoção da gestão de riscos, acarreta um refinamento natural e adaptativo dos processos [2];
- d) Transparência às partes interessadas, referente aos riscos inerentes dos projetos;
- e) Atendimento à recomendação do TCU em seu relatório sobre "Avaliação da Governança de Tecnologia da Informação na administração pública federal, TC n. 003.732/2014-2", que discorre sobre a necessidade da adoção da gestão de riscos de forma institucionalizada [93];
- f) Alinhamento com a Instrução Normativa Conjunta MP/CGU Nº 01/2016, que dispõe sobre a elaboração de controles internos, gestão de riscos e governança no âmbito do Poder Executivo Federal [64].

1.4 Objetivo

1.4.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de uma metodologia de gestão de riscos integrada ao processo de desenvolvimento ágil de software que esteja baseada na norma ISO/IEC 31000 e nos princípios e valores do manifesto ágil, abrangendo os riscos inerentes ao ciclo de vida dos projetos de *software* ágeis.

1.4.2 Objetivos Específicos

Para que o objetivo geral seja alcançado, há necessidade de se alcançar os objetivos específicos a seguir:

1. Diagnosticar a situação atual dos projetos;
2. Pesquisar métodos, técnicas ou ferramentas, processos, normas e modelos correlatos de gestão de riscos adotadas em conjunto com a metodologia ágil;
3. Desenvolver método de identificação dos riscos inerentes ao processo de desenvolvimento de *software* ágil, segundo contexto da CGSI;
4. Desenvolver método de análise e avaliação dos riscos inerentes aos projetos ágeis;
5. Desenvolver estratégia para tratamento e resposta ao risco;
6. Desenvolver *template* do Relatório de Gestão e Comunicação dos Riscos;

1.5 Contribuição Esperada

No contexto do INEP, a contribuição será a elaboração de uma metodologia para gestão de riscos dos projetos de software, aumentando sua probabilidade de sucesso, onde os riscos fiquem evidentes para todos os envolvidos e seja criado plano de resposta aos riscos, assim como o monitoramento possa ser realizado e acompanhado. Esta metodologia de gestão de riscos será criada de forma customizada à necessidade e à realidade da CGSI, com base no diagnóstico da situação atual, respeitando a metodologia de desenvolvimento já adotado na coordenação, visando integrar a gestão de riscos a este processo ágil de desenvolvimento de *software*.

1.6 Estruturação dos Capítulos

O trabalho está estruturado a fim de construir um conhecimento evolutivo sobre a necessidade da adoção da gestão de riscos no desenvolvimento ágil de software na coordenação estudada. Para isso passará pelos capítulos indicados abaixo, onde cada um trará um conteúdo específico que auxilia de forma progressiva na construção do entendimento do tema proposto. O Capítulo 1 trata da Introdução, trazendo a contextualização, a problemática do tema, a justificativa e a contribuição esperada deste trabalho. O Capítulo 2 trata da base conceitual e revisão da literatura, onde são apresentadas pesquisas correlatas ao tema que auxiliará no entendimento do leitor e contemplará o estado da arte do tema proposto no trabalho, passando por conceitos, definições, mostrando a evolução dos modelos de processos de desenvolvimento de software. São abordadas as normas técnicas mais utilizadas nacionalmente e internacionalmente para a gestão de riscos, os artigos científicos que propõem modelos de gestão de riscos integrados ao desenvolvimento ágil ou que discorrem sobre a gestão de riscos em projetos de software. Também são abordadas as legislações governamentais atuais que recomendam a adoção da gestão de riscos na APF. O Capítulo 3 apresenta a metodologia científica: métodos e técnicas de pesquisa utilizadas. Exibe de forma diagramada e consolidada a estrutura macro da pesquisa realizada. O Capítulo 4 expõe o estudo de caso em si, no contexto externo e interno da CGSI. Possui uma análise da situação atual do objeto de estudo com discussão dos resultados. O Capítulo 5 propõe, a partir do diagnóstico e da revisão de literatura, o desenvolvimento da metodologia de gestão de riscos compatível com o desenvolvimento ágil de software já adotado na coordenação estudada. Por fim, o Capítulo 6 apresenta a conclusão do trabalho. Ao final, as referências bibliográficas e o apêndice, onde constarão o *checklist* de identificação de riscos (Apêndice A) e o *template* do Relatório de Gestão e Comunicação de Riscos (Apêndice B).

Capítulo 2

Base Conceitual e Revisão da Literatura

2.1 Modelos de Processos de Desenvolvimento de Software

Um modelo de processo de desenvolvimento de software fornece um guia específico para o trabalho de engenharia de software, definindo o fluxo das atividades e tarefas, suas iterações, os artefatos gerados e a organização do trabalho a ser desenvolvido [79].

Os modelos de processo de desenvolvimento de software fornecem os passos para a realização de um trabalho de engenharia de software disciplinado. Propicia também estabilidade, controle e organização para uma atividade que poderia, sem controle, tornar-se caótica [79]. Ao longo dos anos, muito modelos de processos de desenvolvimento de software foram propostos, houve então uma evolução destes processos a medida que paradigmas eram quebrados [79]. No decorrer deste trabalho são apresentados os modelos de processos de desenvolvimento de software que historicamente foram mais significativos abordando os modelos prescritivos e as metodologias ágeis.

2.1.1 Modelos de Processos Prescritivos

Este tipo de modelo concentra-se em estruturar e ordenar o desenvolvimento de software. O processo é considerado prescritivo uma vez que prescrevem um conjunto de elementos de processo - atividades metodológicas, ações de engenharia de software, tarefas, artefatos, garantia da qualidade e mecanismos de controle de mudanças para cada projeto. Descreve também um fluxo de trabalho que é a forma na qual os elementos do processo estão relacionados [79].

Dentre os modelos de processo prescritivos, destacam-se o: cascata, incremental, evolucionário, especializado e o unificado.

Modelo Cascata

O modelo cascata, também chamado de "ciclo de vida clássico", sugere uma abordagem sequencial e sistemática para o desenvolvimento de software. Conforme Figura 2.1, inicia-se com o levantamento de requisitos, avançando para as fases de planejamento (planejamento do projeto), modelagem (análise do projeto e desenho arquitetural), construção (codificação) e entrega do software. A próxima fase só será iniciada quando a anterior for totalmente concluída. Isto faz com que o esforço da mudança seja maior, uma vez que o software foi entregue por completo e uma mudança poderia impactar em muitas outras funcionalidades do software, consequentemente isto poderá aumentar também o custo para a realização da mudança [87].

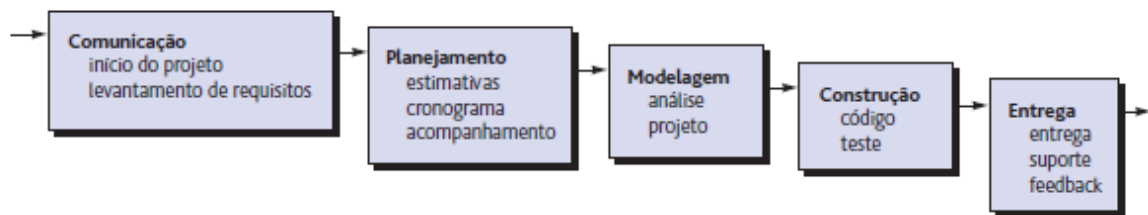


Figura 2.1: Modelo Cascata. Fonte: [79].

Este processo é o paradigma mais antigo da engenharia de software e demonstrou possuir várias falhas, como por exemplo: a impossibilidade do cliente estabelecer todas as suas necessidades já no início do projeto. Outra falha, seria que o cliente somente receberia uma versão funcional do software quando estivesse próximo à data final do projeto. Além disto, induz a equipe a "estados de bloqueios" que seria o tempo em que alguns profissionais teriam que aguardar outros finalizarem suas tarefas para então conseguirem iniciar a deles. Então este modelo tornou-se adequado apenas para casos onde os requisitos sejam fixos e o trabalho deva ser realizado até a sua finalização de forma linear [79], [87].

Modelo de Processo Incremental

O modelo incremental aplica sequências lineares de forma escalonada, a medida que o tempo vai avançando. Cada sequência linear produz "incrementos" de entrega do software. Por exemplo: determinado cliente exige a entrega em uma data impossível de atender. O recomendado é entregar um ou mais incrementos nesta data e o restante do software posteriormente (incrementos adicionais) [79]. Conforme Figura 2.2 o software é entregue

em incrementos, onde os requisitos básicos são atendidos, porém muitas funcionalidades complementares ainda não são entregues. Após avaliação do cliente é então desenvolvido um planejamento para o próximo incremento. Este processo é repetido até que seja gerado um produto completo [79]. No entanto, existem problemas com a entrega incremental.

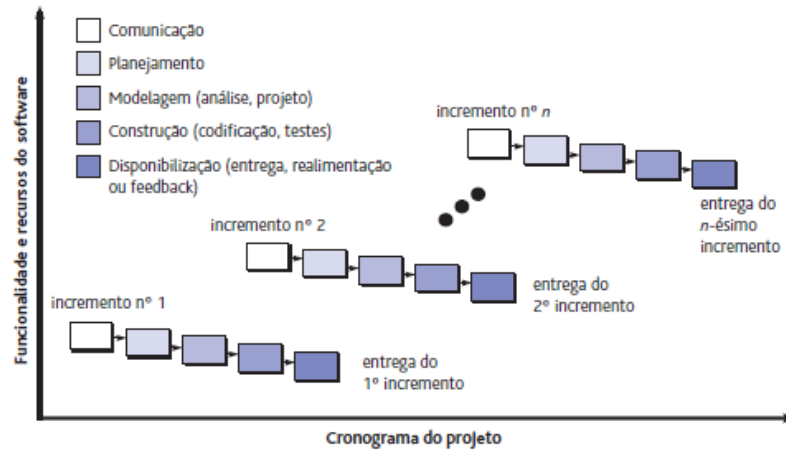


Figura 2.2: Modelo Incremental. Fonte: [79].

Os incrementos devem ser relativamente pequenos e cada um deve entregar uma funcionalidade do sistema. Além disso, a maior parte dos sistemas requerem uma quantidade mínima de recursos básicos para serem usados em diferentes partes do sistema. Como os requisitos não são definidos detalhadamente, até que um incremento seja implementado, pode ser difícil identificar os recursos comuns exigidos por todos os incrementos, dificultando o reuso [87].

Modelo de Processo Evolucionário

Este modelo compreende que sistemas complexos evoluem ao longo do tempo e que frequentemente os requisitos mudam, por isso é inadequado seguir um planejamento inicial até o produto final. Os prazos apertados tornam impossível construir um produto de software abrangente. Os modelos evolucionários são iterativos, apresentam características que viabilizam construções mais refinadas até a completude do software. Existem dois modelos comuns de processos evolucionários [79]:

- 1) Prototipação: A prototipação é comumente utilizada como uma técnica para levantamento dos requisitos que estão obscuros, ela auxilia os envolvidos a compreender melhor o que está para ser construído. Conforme Figura 2.3 faz-se uma reunião com envolvidos para identificar o objetivo e requisitos já conhecidos. É feita uma esquematização de quais necessitam de um maior detalhamento, então, planeja-se uma iteração de prototipação e o projeto rápido leva à construção deste protótipo. Por fim, os envolvidos fornecem um *feedback* que é utilizado para refinar ainda mais os requisitos [87].

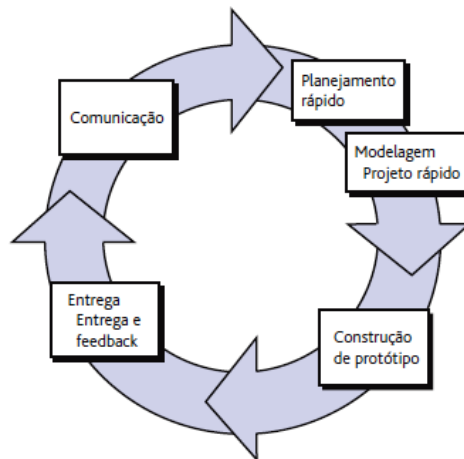


Figura 2.3: Prototipação. Fonte: [79].

- 2) **Modelo Espiral:** Originalmente proposto por Barry Boehm [18], este modelo possui duas características principais que o distingue: estratégia cíclica voltada para ampliar de forma incremental o grau de definição e implementação de um sistema enquanto diminui o grau de risco do mesmo. A segunda característica é que possui uma série de marcos para garantir o comprometimento dos envolvidos. Conforme Figura 2.4, cada volta na espiral é uma iteração e a cada volta, ajustes no planejamento e cronograma são realizados de posse do *feedback* do cliente. A análise de riscos é realizada e o gerente de projetos faz ajustes no número de iterações planejadas para completar o projeto [79], [87].

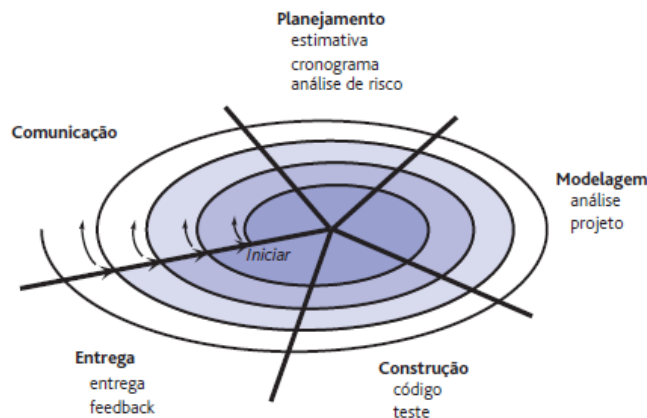


Figura 2.4: Modelo Espiral. Fonte: [79].

Modelo de Processos Especializado

Incluem muitas características dos modelos tradicionais e são adotados em casos restritos [79].

- 1) **Modelo baseado em componentes:** Este modelo está baseado na boa prática da reutilização do software. Ao se construir componentes, estes podem fazer parte de vários sistemas que utilizem a mesma arquitetura, reduzindo assim, o tempo do ciclo de desenvolvimento de software e o seu custo [79]. Os componentes são unidades funcionais independentes que compõem um sistema provendo um serviço [87]. Quando o sistema necessita de um serviço, ele chama este componente sem se preocupar com onde este componente está sendo executado ou qual a linguagem de programação utilizada para desenvolver o componente [87]. Todas as interações ocorrem por meio de interface. A interface é expressa por meio de operações parametrizáveis e seu estado interno nunca é exposto. O código fonte não está disponível de forma que

o componente não precisa ser compilado antes de ser utilizado com outros componentes do sistema [87]. Há um inevitável compromisso entre a reusabilidade e a usabilidade de um componente, ao projetar um componente reusável, aumenta-se a sua complexidade devida a generalização, porém deve-se manter um compromisso entre generalidade e facilidade de compreensão para que assim ele possa ser de fato reusável [87].

- 2) **Métodos Formais:** Embora não seja o método mais adotado, devido o seu alto consumo de tempo e dinheiro, inclui um conjunto de atividades que conduzem à especificação matemática formal do software. Tais métodos permitem especificar, desenvolver e verificar um sistema baseado em computador pela aplicação de notação matemática rigorosa. Problemas como: ambiguidade, incompletude e inconsistência podem ser descobertos e corrigidos mais facilmente. Estes métodos são comumente utilizados em softwares de tráfego aéreo e na área médica [79].
- 3) **Orientado a Aspectos:** É um dos novos paradigmas da engenharia de software, oferece uma abordagem metodológica para definir, especificar, projetar e construir aspectos (mecanismos além da programação para localizar a expressão de uma preocupação cruzada). Estas preocupações podem ser de várias ordens, tais como: exigência do negócio ou da área técnica (desempenho, segurança, tolerância a falhas), outras afetam funções (como a aplicação de regras de negócio) enquanto outras são sistêmicas (gerenciamento de memória ou sincronização de tarefas) [79]. Muitas abordagens estão muito atentas ao aspecto funcional do software se esquecendo que os requisitos/aspectos não funcionais atuam como restrição no software, por isso esta abordagem é tão importante.

Modelo de Processo Unificado

É um processo dirigido à caso de uso, centrado na arquitetura, iterativo e incremental. É uma tentativa de unificação das melhores características dos métodos individuais de análise e projeto orientado a objetos com características adicionais propostas por vários especialistas. Conforme Figura 2.5, este processo define quatro fases principais (concepção, elaboração, construção e transição).

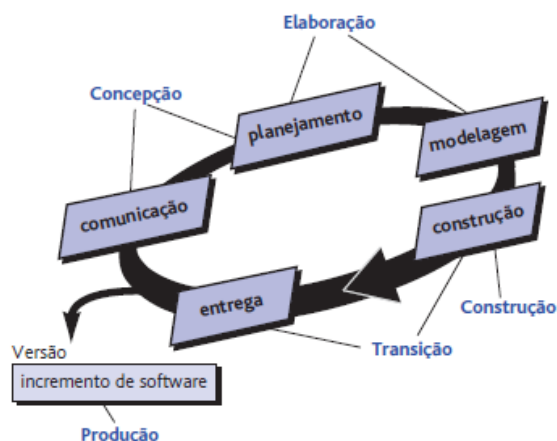


Figura 2.5: Modelo Unificado. Fonte: [79].

Na concepção será realizado o levantamento dos requisitos, o foco é no escopo do produto e o objetivo é definir se há viabilidade de continuar o projeto avaliando os custos e riscos do mesmo. Na elaboração é proposta uma arquitetura rudimentar e se inicia a documentação. A fase de construção é similar a construção em outros métodos, com a codificação e a realização de testes alfa. A transição abrange a entrega e os testes beta dos usuários e seu *feedback*. Como consequência da execução de cada tarefa são entregues artefatos, que são customizados às necessidades da equipe.

2.1.2 Metodologias Ágeis

Os métodos ágeis ou adaptativos baseiam sua essência nos valores e princípios definidos no manifesto ágil. Portanto se faz necessária a compreensão das ideias abrangidas neste manifesto, que estão revolucionando a forma de se construir software nos últimos anos.

Um manifesto frequentemente é associado a um movimento político emergente, propondo mudanças revolucionárias em face do conceito já estabelecido, de certa forma, é exatamente este o objetivo do desenvolvimento ágil. Agilidade é mais do que uma proposta à mudança, ela abrange também uma filosofia que está contida nos valores e princípios citados no manifesto[79].

O Manifesto ágil é uma declaração guiada por 4 valores e 12 princípios, escrita por 17 membros da comunidade ágil que se reuniram em fevereiro de 2001 em Utah nos Estados Unidos, para a elaboração do documento [14]. Os métodos ágeis que colaboraram para a elaboração deste manifesto foram: o *Scrum* (1986), Crystal Clear, XP (Programação extrema, 1996), *Adaptive Software Development*, FDD (*Feature Driven Development*, 1995), and DSDM (*Dynamic Systems Development Method*, 1995) [14].

O manifesto veio como resposta contra métodos pesados de desenvolvimento de *software*, como o modelo cascata, que na percepção dos autores do manifesto, era um modelo burocrático, lento e pesado. Assim as metodologias ágeis inicialmente também ficaram conhecidas por serem métodos "leves" ou " enxutos" [14].

Os métodos ágeis incentivam a estruturação e as atitudes em equipe que tornam a comunicação mais fácil. Também enfatiza a entrega rápida do software funcional e diminui a importância dos artefatos intermediários, aceita o cliente como parte da equipe de desenvolvimento e principalmente reconhece que no mundo real há muitas incertezas e por isso o plano de projeto deve ser flexível [79].

Os valores citados no manifesto ágil [14], foram escritos de forma que se valorize mais os itens a seguir:

- 1) Indivíduos e interações mais que processos e ferramentas;
- 2) *Software* em funcionamento mais que documentação abrangente;
- 3) Colaboração com o cliente mais que negociação de contratos;
- 4) Responder a mudanças mais que seguir um plano;

Os princípios citados no manifesto ágil [14] explicam melhor os seus valores e linha de pensamento da nova metodologia, são eles:

- 1) A maior prioridade é satisfação do cliente, através da entrega contínua e adiantada de *software* com valor agregado;
- 2) Mudanças nos requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente;
- 3) Entregar frequentemente *software* funcionando, em poucas semanas ou poucos meses, com preferência utilizando a menor escala de tempo;
- 4) Pessoas envolvidas com o negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto;
- 5) Construção de projetos em torno de indivíduos motivados, por meio do provimento do ambiente, suporte necessário e confiança para executar o trabalho;
- 6) O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face;
- 7) *Software* funcionando é a medida primária de progresso;

- 8) Os processos ágeis promovem desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente;
- 9) Contínua atenção à excelência técnica e bom *design* aumentam a agilidade;
- 10) Simplicidade, a arte de maximizar a quantidade de trabalho ainda não realizado é essencial;
- 11) As melhores arquiteturas, requisitos e *designs* emergem de equipes auto-organizáveis;
- 12) Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e então refina e ajusta seu comportamento de acordo com este *feedback* [14].

Estes valores e princípios definidos no manifesto, inspirou muitos métodos ágeis. As publicações sobre as metodologias ágeis em geral, iniciaram massivamente em 2001 com o manifesto ágil e entre 2008 e 2009 atingiram o seu pico. Nesta época, os números de artigos apresentados eram recordes em conferências sobre XP e metodologias ágeis [33].

A IEEE *Software*, ranqueava em número de artigos sobre metodologias ágeis, da mesma forma a quantidade de publicações era em sua maioria originárias dos Estados Unidos, Canadá e Europa Ocidental em um total de 63 países, incluindo o Brasil que aparece na 13º posição à frente por exemplo da Austrália, Japão, Coreia do Sul e Índia [33].

Porém apesar do aumento na utilização das metodologias ágeis na indústria de desenvolvimento de *software*, a complexidade de adotá-las é alta, devido a fatores como: cultura organizacional, resistência à mudança, necessidade de patrocínio e envolvimento da alta gerência [21].

De acordo com a pesquisa realizada em 2014, em 400 empresas Brasileiras, foram levantados os perfis das organizações que desenvolvem *software* utilizando princípios ágeis, e assim foram segregados três perfis, através de clusterização: o primeiro perfil, possuía altos níveis de utilização de princípios ágeis e altos níveis de sucesso em projetos. O segundo perfil, possuía alto nível de utilização de princípios ágeis, mas baixo nível de sucesso e o terceiro perfil, tinha baixo nível de utilização e baixo nível de sucesso. Desta forma, concluíram que somente o uso das metodologias ágeis não garante o sucesso dos projetos de desenvolvimento de *software*, porém os resultados obtidos nesta pesquisa mostraram que as organizações com as mais altas taxas de sucesso em *software*, são aquelas que têm maiores taxas de capacidade em termos de equipe, fatores culturais ágeis, a comunicação com o cliente, configuração do ambiente e relações com parceiros externos, ficando claro que de forma isolada, o uso de princípios ágeis, não é suficiente para o sucesso dos projetos de *software* [15].

As metodologias ágeis mais utilizadas: *Framework Scrum*, *XP(Programação Extrema)*, *DSDM(Método de Desenvolvimento de Sistemas Dinâmicos)* e *AUP(Processo Unificado Ágil)*, serão detalhadas a seguir.

Framework Scrum

Historicamente o termo *Scrum* surgiu em um artigo intitulado “The new new product development game”, de autoria de Takeuchi e Nonaka (1986) que descreveram uma abordagem holística, na qual equipes de projetos, são compostas de pequenas equipes multifuncionais trabalhando com sucesso rumo a um objetivo comum, que os autores compararam à formação *Scrum* do Rugby [92], [73].

Porém a criação do processo *Scrum*, como é conhecido hoje, foi fruto de um trabalho em conjunto. Jeff Sutherland, queria um processo semelhante ao *Scrum*, em que ao final de iterações curtas, o executivo chefe da organização, pudesse ver a exibição do incremento funcional do software ao invés de cronogramas em papel. Neste mesmo período, Ken Schwaber, pesquisava como poderia ajudar sua empresa a melhorar seu processo de *software*, com o objetivo de aumentar a produtividade das equipes [73]. Após uma profunda análise de como outras vendedoras de *softwares*, bem-sucedidas construíam *softwares*, Schwaber percebeu que o processo de desenvolvimento de todas era semelhante, pois usavam processos empíricos que exigiam inspeção e adaptação constantes. A pedido da Object Management Group (OMG) em 1995, Jeff Sutherland e Ken Schwaber trabalharam em conjunto para resumir o que haviam aprendido ao longo dos anos, eles criaram o que foi chamado *Scrum*, descrito em um artigo de Schwaber, “*Scrum and the perfect storm*” [73], [90].

O *Scrum* foi fortemente influenciado pelos processos de produção e desenvolvimento *just-in-time* (JIT) implementados na Toyota e Honda pela gestão *Lean* em seus processos de produção [91]. O foco do *Scrum* é contudo diferente do JIT, onde a principal ênfase é colocada sobre a utilização máxima de mão de obra, tudo o que não entregar valor para o cliente é omitido, o foco está no resultado e não no processo [91].

O pilar para a criação do *Scrum* é o fato do desenvolvimento de *software* ser um processo complexo, onde muitos fatores influenciam o resultado final. Isso dificulta ou mesmo impossibilita um planejamento muito antecipado como ocorre em um processo de desenvolvimento tradicional ou cascata [32].

Scrum estendeu o desenvolvimento de *software* incremental para o que é chamado de processo empírico de controle, onde o *feedback* é o elemento central. *Scrum* é inspirado na teoria da complexidade, na dinâmica de sistemas e na teoria da criação do conhecimento de Takeuchi e Nonaka, adaptado para um conjunto de desenvolvimento de *software* [32].

O *Scrum* é um gerenciamento de projeto iterativo e incremental que provê um *framework* simples de inspeção e adaptação, ao utilizá-lo o *software* é entregue em incrementos ou *Sprints*. O Scrum deixa claro a eficácia relativa das práticas de gerenciamento e desenvolvimento de produtos, de modo que se possa melhorá-las [85].

O Scrum enfatiza o uso de um conjunto de padrões de processos de software que provaram ser eficazes para projetos com prazos de entrega apertados, requisitos mutáveis e urgência do negócio. Cada um destes padrões de processos (*backlog*, *sprints* e reuniões ou cerimônias) definem um conjunto de atividades de desenvolvimento e permite à equipe, construir um processo que se adapte às necessidades do projeto [79].

Conforme Figura 2.6, o processo *Scrum*, tem início com uma lista de necessidades do produto, chamada de *backlog*, que foi elicitada e priorizada na visão do cliente, ou seja, do PO (*Product Owner*). Durante a reunião de planejamento a funcionalidade priorizada será destrinchada em itens para compor a *Sprint backlog* sendo esta, uma lista de tarefas que o time terá que executar, e que possa ser realizada dentro de um tempo limitado conhecido por *time-box*. Este é o conceito de *Sprint*, que geralmente varia entre duas a quatro semanas [85].

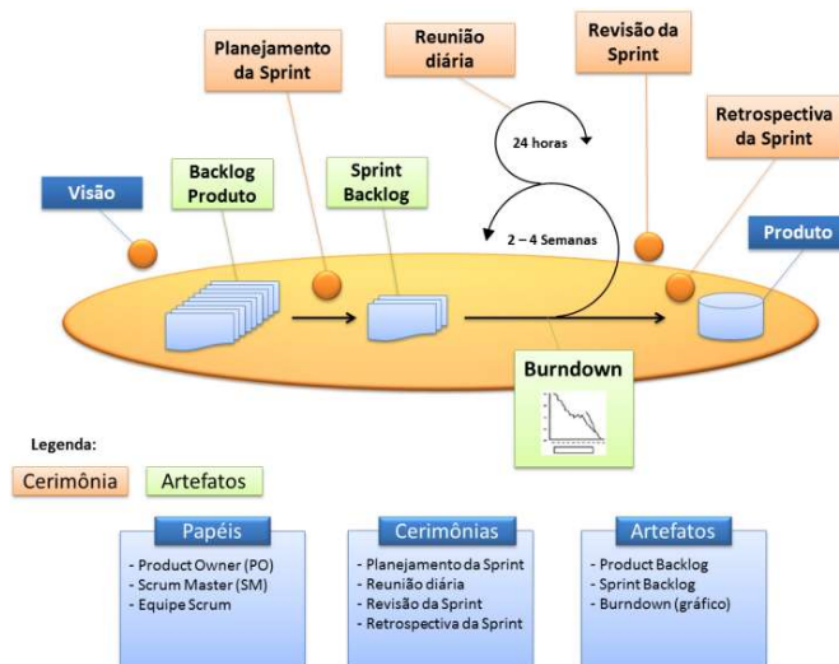


Figura 2.6: Framework Scrum. Fonte:[81].

A execução em si, inicia-se com a reunião de planejamento com toda a equipe, onde todos são apresentados aos itens que devem ser executados na *sprint* e após avaliação a equipe fornece um prazo ao gerente de projetos. O processo de implementação só termina

com a reunião de revisão, onde o *software* funcional é exibido para o PO, que pode aceitar ou solicitar ajustes. Após cada *sprint*, a equipe deve se reunir e realizar a reunião de retrospectiva, que irá levantar as lições aprendidas que o time levará para a próxima *sprint*, se adaptando e melhorando [85].

O papel do PO é certificar-se de que a visão de um produto é mantida ao longo do processo. Isso é realizado durante a elicitación e especificação de requisitos, além de manter uma estreita cooperação com a equipe, o PO deve ainda, homologar ou rejeitar os resultados das *sprints*, tais tarefas são realizadas durante a reunião de revisão, onde a equipe de desenvolvimento exhibe o incremento de software funcional [91].

Um outro papel de extrema importância no *Scrum* é o *scrum master*, que é um membro da equipe encarregado de resolver problemas que impeçam a equipe de trabalhar de forma eficaz, ou seja, ele é um removedor de impedimentos, seu trabalho é manter o foco da equipe na entrega planejada [36].

As reuniões diárias, ajudam a equipe revelar problemas em potencial, o mais cedo possível. Também leva à socialização do conhecimento, diminuindo a assimetria de informações no projeto[79].

Em um estudo de 2008 que avaliou o impacto das práticas ágeis sob o desenvolvimento de *software*, foi indicado que as práticas ágeis utilizadas nos projetos, trouxeram efeitos positivos sobre a comunicação no interior das equipes de desenvolvimento. Por exemplo, o planejamento da *sprint*, área de trabalho aberta e reuniões diárias foram sugeridas como práticas eficientes para comunicar requisitos, características e tarefas do projeto em equipes de desenvolvimento de *software* ágeis. Nas situações em que estas práticas foram utilizadas em conjunto, houve um indicativo do aumento da comunicação informal como um fator que diminui a necessidade de documentação em equipes de desenvolvimento de *software* e, portanto, corrobora para uma maior produtividade do desenvolvimento de *software*, quando comparado ao desenvolvimento impulsionado apenas por documentações [74].

Conforme Figura 2.7 na versão de 2016, da pesquisa anual sobre metodologias ágeis, intitulada: *Version One State of Agile*, dentre os métodos e práticas ágeis, citados: *Scrum/XP* híbrido, múltiplas metodologias, *Scrumban*, *Kanban*, desenvolvimento iterativo, desenvolvimento *Lean*, modelagem ágil, *FDD*, *AgilUP*, *DSDM/Atern* e outras, o *scrum* se destaca e lidera com folga sendo a metodologia ágil mais utilizada no mundo, correspondendo a 58% de utilização.

Percentual de Metodologias Ágeis usadas mundialmente

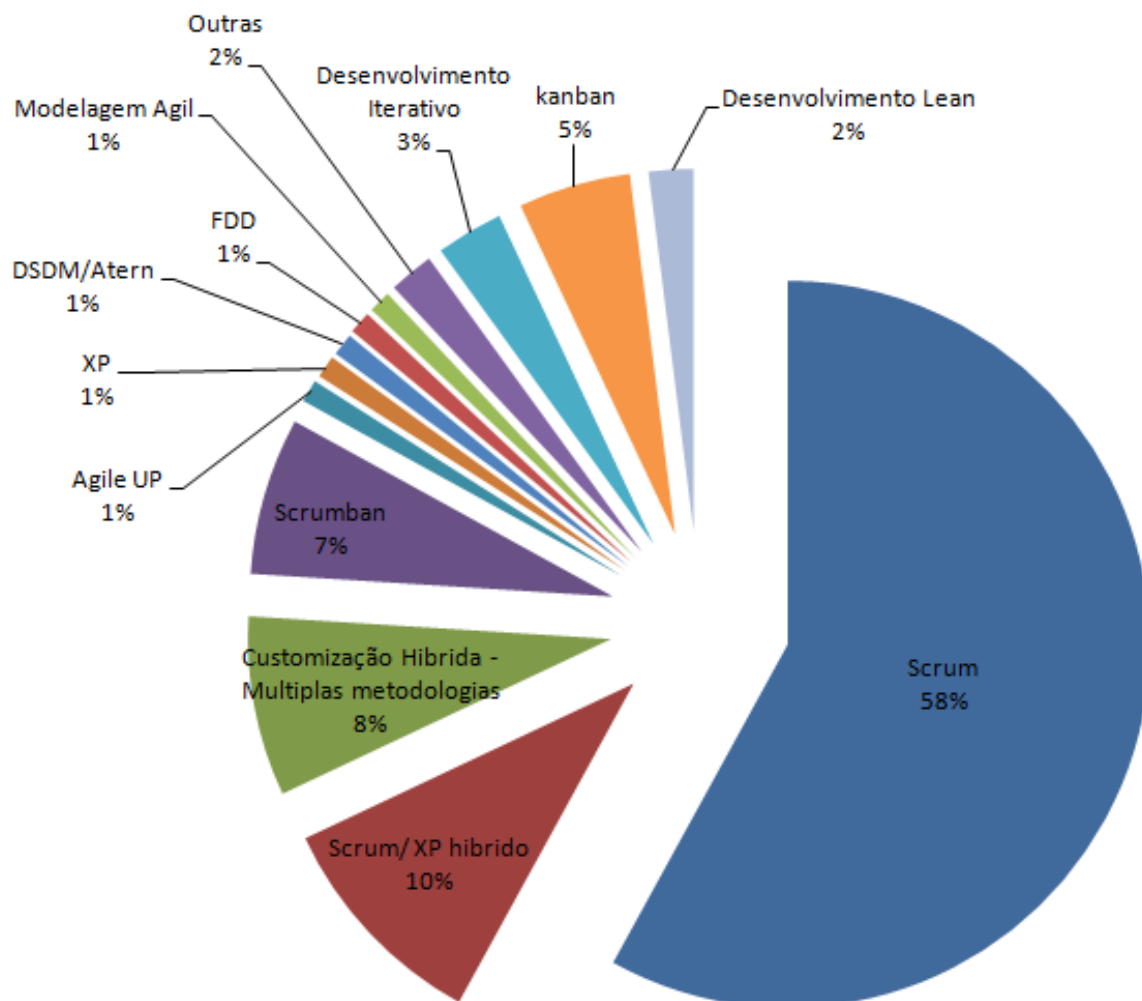


Figura 2.7: Gráfico de métodos ágeis. Fonte:[99].

Já as demais não possuem uma porcentagem expressiva, atingindo no máximo a marca de 10% individualmente [98]. Nesta mesma pesquisa, realizada na versão de 2013, o *Scrum*, era utilizado em 56% das vezes [99] e nota-se também que de 2013 para 2016 o *Scrum* teve um aumento de 2% na sua utilização mundial [98], hoje atingindo um percentual de utilização de 58% entre todas as metodologias ágeis pesquisadas. E ainda, na mesma pesquisa, citam-se as cinco práticas mais utilizadas, do *Scrum*, sendo elas: reuniões diárias, iterações curtas, *backlogs* priorizados, planejamento da iteração e reunião de retrospectiva.

Extreme Programming - XP (Programação Extrema)

A Programação Extrema, utiliza uma metodologia orientada a objetos e envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas: planejamento, projeto, codificação e testes. As atividades-chave são exibidas na Figura 2.8, e são sintetizadas a seguir:

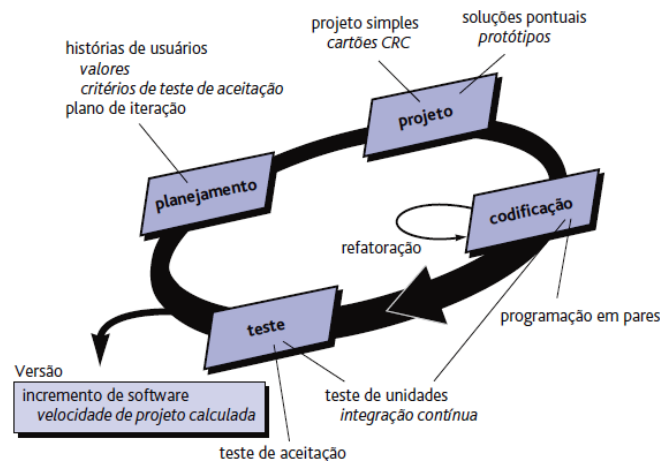


Figura 2.8: Extreme Programming-XP (Programação Extrema). Fonte:[79].

- **Planejamento:** É realizado a partir de histórias de usuários (que descrevem as características e funcionalidade solicitada para construção). No chamado jogo do planejamento, cada história é colocada em uma ficha e o cliente atribui um valor (uma prioridade) baseando-se no valor comercial. A equipe avalia cada história e atribui um custo, medido em semanas de desenvolvimento. Se ela demandar mais de três semanas é solicitado ao cliente que a divida em histórias menores e segue novamente este ciclo. Novas histórias podem ser escritas a todo momento. Há algumas estratégias para definir a ordem de desenvolvimento e deve ser combinada com a equipe, por exemplo: todas as histórias de usuário priorizadas de maior valor são desenvolvidas primeiro, ou as histórias de maior risco são desenvolvidas primeiro [79].
- **Projeto:** O projeto XP deve utilizar o princípio da simplicidade, pois XP estimula a refatoração, ou seja aperfeiçoamento do projeto de codificação depois dele ter sido criado. Significando que o ato de projetar seja realizado continuamente, enquanto o sistema estiver em elaboração [79]. A XP estimula o uso de cartões CRC (classe-responsabilidade-colaborador) como um mecanismo eficaz para pensar o software em um contexto orientado a objetos. Os cartões CRC identificam e organizam as classes

orientadas a objetos relevantes para o incremento de software corrente [79]. Se for encontrado um problema de projeto complexo, a XP recomenda a criação imediata de um protótipo operacional dessa parte do projeto. Denominada solução pontual, o protótipo do projeto é implementado e avaliado com a finalidade de reduzir o risco antes da implementação de fato desta funcionalidade [79].

- **Codificação:** Desenvolvimento de histórias e testes unitários automatizados a serem incluídos na versão corrente. A programação é realizada em pares, onde dois desenvolvedores trabalham juntos em uma mesma estação de trabalho. Enquanto um assume um papel de codificador o outro, garante que padrões estejam sendo seguidos [79].
- **Teste:** Os testes unitários automatizados, devem ser chamados em caso de testes regressivos, toda vez que o código for modificado. Os testes de integração e validação podem ocorrer diariamente. Os testes de aceitação são elaborados com base nas histórias de usuário [79].

Após a primeira versão do projeto (incremento de software) ser entregue, a equipe XP calcula a velocidade do projeto. De forma simples, a velocidade do projeto é o número de histórias de clientes implementadas durante a primeira versão [79]. Assim, a velocidade do projeto pode ser utilizada para ajudar a estimar as datas de entrega e o cronograma para versões subsequentes e determinar se foi assumido um compromisso exagerado para todas as histórias ao longo de todo o projeto de desenvolvimento [79]. Se ocorrer um exagero, o conteúdo das versões é modificado ou as datas finais de entrega são alteradas.

Conforme o trabalho de desenvolvimento prossegue, o cliente pode acrescentar histórias, mudar o valor de uma já existente, dividir algumas ou eliminá-las. Em seguida, a equipe XP reconsidera todas as versões remanescentes e modifica seus planos de forma correspondente [79].

Método de Desenvolvimento de Sistemas Dinâmicos - DSDM

Fornece uma metodologia para construir e manter sistemas que satisfaçam restrições de prazo apertado por meio do uso da prototipação incremental. Baseia-se numa versão modificada do princípio de Pareto - 80% de uma aplicação pode ser entregue em 20% do tempo que levaria para entregar a aplicação completa (100%). Ou seja, somente o trabalho suficiente é requisitado para cada incremento, os detalhes restantes podem ser concluídos depois, quando outros requisitos do negócio forem conhecidos ou alterações tiverem sido solicitadas e acomodadas [79]. O DSDM define três ciclos iterativos:

- **Iteração de Modelos funcionais:** Produz um conjunto de protótipos incrementais que provê funcionalidade para o cliente [79].

- **Iteração de Projeto e Desenvolvimento:** Revê os protótipos desenvolvidos para garantir que agreguem valor de negócio em termos operacionais [79].
- **Implementação:** Coloca a última versão do incremento de software no ambiente operacional[79].

O DSDM pode ser combinado com a XP para fornecer uma abordagem integrada que defina um modelo de processos confiável. Poderia-se mesclar o ciclo de vida do DSDM juntamente com as práticas básicas da XP que são necessárias para se construir incrementos de software.

Agile Unified Process - AUP (Processo Unificado Ágil)

O Processo Unificado Ágil, adota uma filosofia serial do mais abrangente (geral) para o que é mais específico. As fases clássicas são: concepção, elaboração, construção e transição, que permite visualizar o fluxo geral do processo de desenvolvimento de software. Porém para cada fase a equipe itera para alcançar a agilidade e entregar incrementos de software que agreguem valor, o mais rápido possível [79]. Em acréscimo, o UP também se propôs a ser adaptativo, de forma a atender às necessidades específicas de cada projeto. O surgimento do Processo Unificado trouxe diversas vantagens ao processo de desenvolvimento de softwares quando comparado ao modelo em cascata. Algumas delas são: a entrega de versões com mais constância, precipitando o *feedback* sobre o produto, a antecipação em identificar as mudanças, diminuindo o custo das alterações no decorrer do desenvolvimento, o controle dos riscos inerentes ao projeto, o foco no produto, e o aumento da qualidade do produto final [94].

Kanban

Kanban é uma palavra japonesa e significa "cartão visual". Essa palavra é utilizada para descrever o sistema que a empresa Toyota utiliza desde a década de 1950 para controlar a linha de produção de seus veículos e aumentar a eficiência da produção pela eliminação contínua de desperdícios. A metodologia Kanban para desenvolvimento de software foi proposta por David J. Anderson e define um framework para melhoria incremental de processos e sistemas em organizações. A adoção do Kanban é ancorada na filosofia de que, para aperfeiçoar um processo, deve-se começar com o que se está fazendo agora, concordar em buscar mudanças incrementais e evolucionárias, além de respeitar o processo atual, com seus papéis, responsabilidades e cargos. Destaca-se também a necessidade de fomentar a postura de liderança em todos os níveis da organização [94].

Kanban contempla o uso das seguintes práticas:

- Visualizar o trabalho em andamento: Criação de um quadro eletrônico ou físico dividido por raias para identificar estágios do trabalho, conforme mostra a Figura 2.9, onde a velocidade inicial definida para a equipe, permite que se desenvolva até 3 estórias de usuário, e que se teste e implante até duas estórias por vez. Esta é a capacidade da equipe que vai se ajustando para evitar tanto a inanição (falta de tarefas) ou possíveis gargalos.

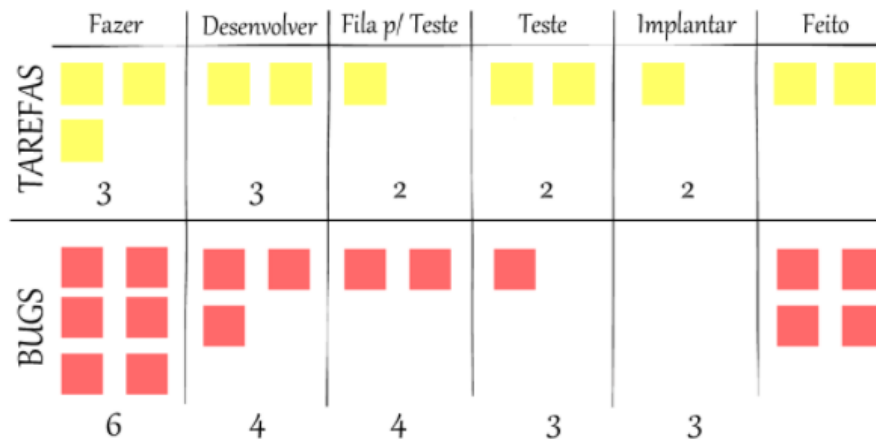


Figura 2.9: Kanban. Fonte:[4].

- Limitar o trabalho em progresso: WIP - *Work-in-Progress* necessita ser limitado. Para evitar gargalos no fluxo ou ociosidade. O WIP é um parâmetro que deverá ser acertado no decorrer do processo, na tentativa e erro. A definição do WIP de cada estágio deve seguir o bom senso e as particularidades da equipe, como o cálculo da velocidade da equipe (número de estórias de usuário implementadas em uma iteração). Os limites iniciais são apenas palpites que são dados em momentos de pouca informação. À medida que se obtém informações sobre o sistema e que a forma de trabalho é aprimorada, os limites são ajustados. Para evitar gargalos, o tamanho dos itens também é considerado, pois itens grandes bloqueiam recursos por muito tempo, enquanto itens pequenos fluem mais rapidamente pelo sistema [94].
- Explicitar as políticas que estão sendo seguidas: As políticas são representadas por padrões e listas de verificação para completar uma tarefa. Para deixar explícitas, as políticas são acrescentadas ao quadro, observando que estágios podem ser inteiramente dedicados à garantia da qualidade. O objetivo final dessa prática é a melhoria na qualidade do produto [94].
- Medir e gerenciar o fluxo: São vários os tipos de medições que podem ocorrer, mas a maioria deles diz respeito à quantidade de trabalho realizado ou que falta

ser realizado, tempo de duração do ciclo, índice de defeitos do produto e itens bloqueados em estágios no sistema. Gráficos com as medições são expostos junto ao quadro. O gerenciamento do fluxo se dá por meio de algumas definições: a fila de entrada dos itens a serem tratados deve estar sempre ordenada de acordo com a prioridade, a fim de que as pessoas possam retirar os itens que estejam no topo. Para fazer a priorização, são levados em consideração diversos fatores: riscos e incertezas; necessidades básicas, como infraestrutura; manutenção de tamanho dos itens equilibrados para um fluxo constante; tipos de história também equilibrados, para manutenção do fluxo de entrega de valor; e dependências entre os itens. Definir um item como prioritário significa abrir mão de priorizar outro. As escolhas devem ser conscientes [94].

- Incentivar a melhoria contínua: A adoção das práticas anteriores já permite a identificação de oportunidades de aperfeiçoamento, mas o Kanban atinge processo contínuo de melhoria com a adoção de eventos de retrospectiva de forma regular. Com as reuniões de retrospectiva, o time concentra-se no fluxo e identifica oportunidades de mudanças estruturais maiores [94].

O kanban é uma estrutura de melhoria de processo que muito contribuiu para o sistema Toyota de produção [73]. Denota uma técnica de cartões, utilizada para limitar o trabalho em processo, aumentando assim o fluxo de trabalho a medida que corrige gargalos e ajusta WIP [73]. Provê transparência ao fluxo de trabalho corrente e realiza adaptações e melhorias constantes por meio dos *feedbacks*, obtidos diretamente do quadro que exhibe o fluxo de trabalho em andamento. O seu uso vem sendo amplamente difundido na comunidade de desenvolvimento de software, sendo adaptável a qualquer tipo de método ágil [4].

As empresas e organizações que estão migrando para os métodos ágeis passam pelo desafio de adequar a gestão dos tradicional dos projetos de software em gestão ágil. A mudança cultural e metodológica, necessita ser construída em conjunto.

A maior parte dos projetos ágeis em sua maioria são projetos de software, por isso, uma vez que a organização possui outros tipos de projetos, a convivência dual entre os dois estilos de gerenciamento de projetos (tradicional x ágil), foi definida pelo Gartner como TI Bimodal [39]. Este conceito é uma realidade nos escritórios de projetos e no dia a dia das organizações, definindo estilos diferentes e coerentes, aplicado a cada tipo de situação.

Na seção a seguir são evidenciadas as diferenças básicas entre os modelos gerenciais: tradicional e ágil.

2.2 Gestão de Projetos

2.2.1 PMBOK - Project Management Body of Knowledge (Corpo de Conhecimento em Gerência de Projetos)

O PMBOK 5º edição fornece diretrizes para o gerenciamento de projetos individuais e define os conceitos relacionados com o gerenciamento de projetos. Ele também descreve o ciclo de vida de gerenciamento de projetos e seus respectivos processos. O projeto é 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 do projeto deixar de existir [75].

Um projeto também poderá ser encerrado se o cliente (cliente, patrocinador ou financiador) desejar encerrá-lo. Cada projeto cria um produto, serviço ou resultado único. O resultado do projeto pode ser tangível ou intangível como por exemplo: a criação de um software [75].

A extensão do PMBOK 5º edição, para *softwares* contém boas práticas de gerenciamento de projetos de software, tais contribuições foram feitas pelo PMI - Project Management Institute em conjunto com a IEEE (*Computer Software Extension Committee*) [76]. A extensão assim como o PMBOK não cobre todas as situações mas ao invés disto, apresenta boas práticas. A principal contribuição é a descrição dos processos do PMBOK que são aplicáveis ao gerenciamento adaptativo do ciclo de desenvolvimento de software. Os métodos de desenvolvimento adaptativos e o ciclo de vida do software são bem adequados para o gerenciamento de projetos de software pois sua vantagem é reconhecer a natureza intangível do software [76].

Segundo a extensão, os métodos ágeis não são ciclos de vida de projeto e sim de desenvolvimento, que podem ser incluídos dentro do ciclo de vida dos projetos de software adaptativos. Esta extensão também faz inúmeras adequações entre os cinco grupos de processos e as disciplinas da gestão de projetos tradicional, para a gestão de projetos de software e além disto abarca soluções de gerenciamento de *software* que utilizam métodos ágeis.

Para ilustrar as diferenças entre os vários fatores do método tradicional de gerenciamento de projetos e o *Framework Scrum*, o Quadro 2.1, esclarece como são tratados estes fatores em ambos métodos de gerenciamento de projetos.

Embora muitas melhorias tenham sido alcançadas na engenharia de *software*, a maioria dos projetos de *softwares* continuam a usar mais recursos do que o planejado, levar mais

Quadro 2.1: Checklist de fatores àgeis Scrum e métodos tradicionais. Fonte: [88]

Risco	Item de Risco	Referência	Tradicional	Ágil
1	Controle	[25],[71],[80]	Centrado no processo	Centrado nas pessoas
2	Estilo de gerenciamento	[25],[29]	Comando controle	Liderança e colaboração tácita
3	Gerenciamento do conhecimento	[25],[71],[29]	explícito	Tácito
4	Atribuição de função	[25],[95],[71],[29]	Individual-favorecendo a especialização	Equipes auto-organizáveis-encoraja a permutabilidade
5	Comunicação	[25],[71],[29],[96],[80]	Formal e somente quando necessário	Auto - organizada
6	Modelo de desenvolvimento	[25],[43]	Modelo de ciclo de vida (cascata, espiral ou outras variações)	Modelo de entrega evolucionário- iterativo e incremental
7	Estrutura organizacional desejada	[25],[96]	Mecanicista(burocrático com alto formalismo)	Orgânica (flexível e participativa, encoraja a ação social cooperativa)
8	Localização da equipe	[25],[95]	Predominantemente distribuída	Predominantemente alocada
9	Métricas tradicionais da engenharia	[95]	Milestone (marcos do projeto) e valor agregado	Requisitos queimados(<i>burnt</i>) conclusão da história de usuário
10	Tamanho da equipe	[25],[71]	Frequentemente mais de 10	De 5 a 15
11	Documentação	[25],[95],[71],[62]	substancial	pouca
12	Utilização de recursos	[95]	Ótimo(bem planejado)	Necessita de planejamento contínuo
13	Envolvimento do cliente	[25],[29]	Importante durante a análise de requisito	Importância crítica e contínua
14	Políticas e processos de RH - recompensas e reconhecimento	[25],[95]	Adequado para organizações com matriz projetizada	Precisam de reconhecer o desempenho, enfatizar o desempenho da equipe sob o indivíduo e fomentar a cultura ágil
15	Papeis, responsabilidades e perfis	[25],[95],[71],[55],[29]	Favorece o individualismo	Favorece a cultura ágil
16	Estimativa de custo	[95]	Os requisitos são definidos antes, portanto possui estimativas precisas.	A estimativa é descoberta em cada sprint.
17	Controle de qualidade	[95],[62],[29]	A maior parte do controle de qualidade são atividades planejadas ao final de cada fase	Contínuo controle de qualidade
18	Assinatura das partes interessadas nos requisitos	[95]	Requerida para começar as atividades da fase, mais importante no início do projeto.	Não definido
19	Questões contratuais	[95],[43]	As estimativas são claras no início do projeto, os marcos são definidos e o progresso é medido em termos de valor.	As estimativas evoluem e o progresso não é medido em termos de valor agregado.
20	Rotatividade de pessoal	[84] e entrevistas	É um desafio	É um desafio muito grande

tempo para ser concluído, fornecer menos funcionalidades e menos qualidade do que o esperado [12].

Muitos estudos relacionam essas falhas com o gerenciamento inadequado de projetos, apontando para problemas de comunicação, alocação errada das habilidades da equipe, capacitações insuficientes e incapacidade do gerente em prever e ajustar o comportamento do projeto [12].

A gestão de riscos tem ganhado importância no gerenciamento de projetos de *software* nos últimos anos. As incertezas enfrentadas pelos projetos de *software* devem ser levadas em conta ao planejar e controlar o desenvolvimento de sistemas de *software*. A gestão de riscos é uma disciplina do processo da gestão de projetos, segundo PMBOK [75].

Em estudo brasileiro, sobre gestão de portfólio de projetos, se propôs a priorização dos projetos, por meio do nível de risco de cada projeto, que é probabilidade de um projeto falhar em atingir seus objetivos propostos [26]. O nível de risco de um projeto, pode ser definido em um único número, ou indicador de risco. Por exemplo: se um projeto tem um nível de risco de 30%, então tem 70% de chance de ser bem sucedido [26]. O nível de risco ajuda os gestores a compararem dois ou mais projetos com base em seus riscos. Além disso, ao quantificar os riscos e aspectos destacando qual o projeto pode ser mais propenso a riscos, os gerentes podem identificar melhor onde aplicar o seus recursos limitados na tentativa de alcançar os objetivos dos projetos [26].

O risco pode ocorrer em cada fase do desenvolvimento de software, tais como os riscos a seguir: na compreensão da necessidade, no desenvolvimento do *software*, em recursos humanos, técnicos, de integração de módulos, de viabilidade, etc. À medida que cada projeto é único e distinto, os riscos apresentam variações e medi-los é muito importante. Embora riscos ocorram em cada fase do desenvolvimento, a identificação dos riscos deve ser realizada com a maior brevidade, pois o contrário da gestão de riscos é a gestão de crises [82].

2.3 Riscos

Pode-se conceituar o risco como sendo um problema em potencial, que pode ou não ocorrer. Portanto, é caracterizado por sua incerteza [79]. Risco diz respeito a acontecimentos futuros, por isso reconhecer o que pode dar errado é o primeiro passo, chamado de "identificação de risco". Tão importante quanto identificar os riscos é identificar os riscos "corretos". Para que isso aconteça, é importante identificar todos os riscos evidentes, tanto na visão dos gerentes de projetos, quanto na visão da equipe [79]. A análise e gestão de riscos são uma série de passos que ajudam uma equipe de software a entender e gerenciar a incerteza [79]. Além de identificar o risco é necessário avaliar sua probabilidade

de ocorrência e impacto (consequências associadas ao risco) e então estabelecer plano de resposta aos riscos, que é composto de estratégias de tratamento que ajudarão a equipe a lidar de forma pró-ativa com o risco. Além disto, relatórios de contingência são descritos com ações a serem tomadas quando o risco vier a se concretizar [79].

Os riscos podem ser classificados em:

- 1) Riscos Conhecidos: são os que podem ser descobertos após uma detalhada avaliação do Plano de Gerenciamento do Projeto, do ambiente onde está sendo desenvolvido, e de outras informações confiáveis como (data irreal de entrega, ambiente de desenvolvimento ruim, ou falta de documentação dos requisitos)
- 2) Riscos Previsíveis: são baseados em experiências de projetos anteriores, como: rotatividade de pessoal, comunicação deficiente com o cliente, diluição do esforço da equipe conforme solicitações de manutenções vão sendo atendidas.
- 3) Riscos Imprevisíveis: alguns riscos são impossíveis de prever, ou são extremamente difíceis de identificar com antecedência. Exemplo: tentar prever um terremoto.

Existem ainda riscos genéricos, que são ameaças em potencial a todos os projetos de software e riscos específicos de produto. Um método bastante utilizado para se identificar riscos é criar um *checklist* dos itens de risco conhecidos e previsíveis em categorias genéricas [79]. As equipes de software não possuem recursos para resolver todos os riscos possíveis com o mesmo rigor, por isso, priorizando os riscos ela se torna capaz de alocar recursos onde eles terão maior impacto [79]. Risco envolve escolha e a incerteza que a própria escolha traz. Paradoxalmente, a única certeza é que há riscos em tudo [79].

2.4 Gestão de Riscos

2.4.1 ABNT NBR ISO/IEC 31000

Segundo a ISO 31000(2009), a gestão de riscos é sistemática, estruturada e feita sob medida, levando em consideração aspectos do contexto externo e interno da organização.

A norma fornece princípios e orientações genéricas sobre gestão de riscos para que possa ser aplicada ao contexto específico de cada empresa.

A norma pode ainda, ser usada por qualquer público, iniciativa privada ou comunidade, associação, grupo ou de forma individual, não sendo específica para um ramo industrial ou setor específico. A norma é aplicável a toda a estrutura de uma organização, e em uma vasta gama de atividades, incluindo estratégias e decisões, operações, processos, funções, projetos, produtos, serviços e ativos. Aplica-se a qualquer tipo de risco, qualquer que seja a sua natureza, quer positiva ou tendo consequências negativas.

Conforme a Figura 2.10 a norma é composta por três componentes: um conjunto de princípios, um *framework*(estrutura) e um processo para o gerenciamento dos riscos [37].

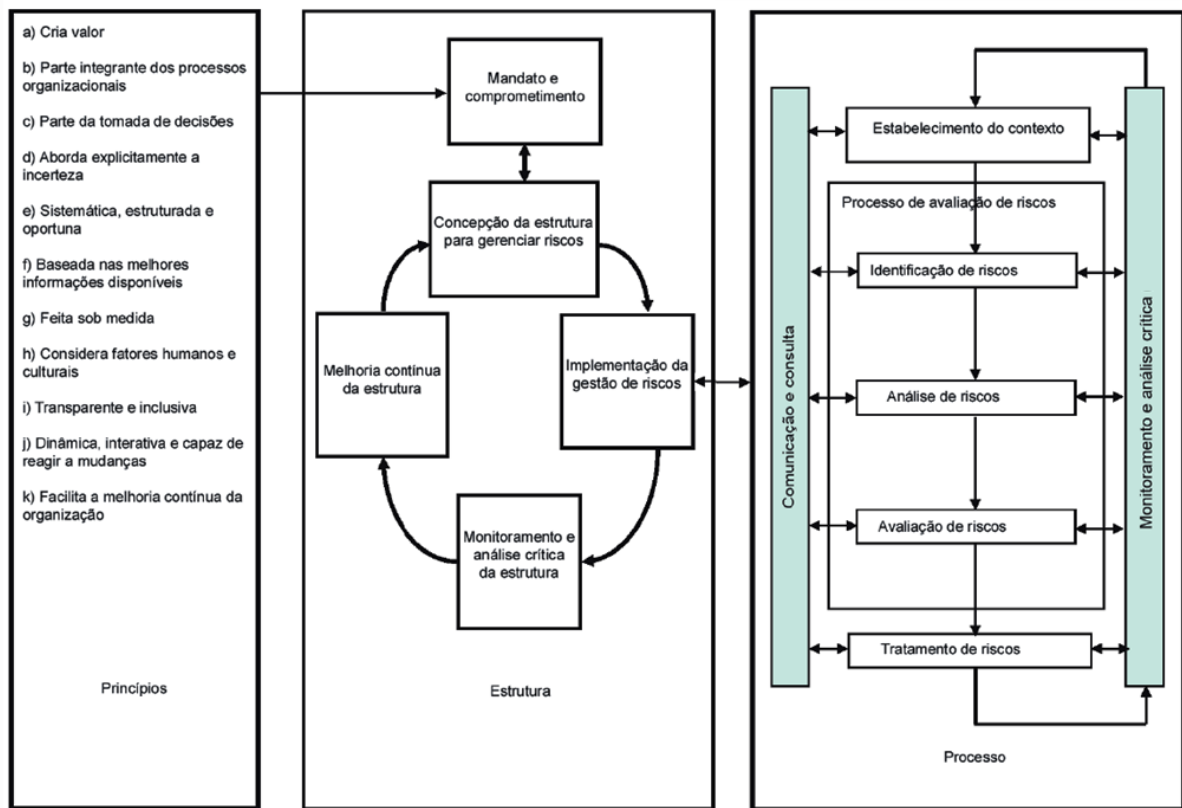


Figura 2.10: Processo da gestão de riscos. Fonte: Adaptado da ABNT NBR ISO/IEC 31000:2009.

Seguem os princípios da gestão de riscos:

- Cria e protege o valor para os objetivos de aumento de segurança, *compliance*, eficiência de operações e serviços, proteção ambiental e aumenta a segurança das pessoas;
- É parte integral de todos os processos da organização;
- Trata explicitamente da incerteza;
- É sistemática e estruturada, segue um conjunto de atividades estruturadas;
- É baseada na melhor informação disponível;
- É feita sob medida, é customizada;
- Leva em consideração fatores humanos e culturais;
- É transparente e inclusiva;

- É dinâmica, iterativa e responsiva à mudanças e
- Facilita a melhoria contínua da organização, a medida que as respostas aos riscos muitas vezes agregam melhorias nos processos.

O *framework* é a base para o gerenciamento de riscos na organização em todos os níveis e provê a integração com o sistema de gestão. Os elementos são [37]:

- Mandato e comprometimento - Requer o comprometimento dos executivos da organização e da alta administração;
- Projeto do *framework* para gerenciamento de riscos - Compreende o entendimento do contexto da organização, uma política para a gestão de riscos, a atribuição de responsabilidades e prestação de contas, a integração dos processos, a alocação de recursos, e o estabelecimento dos mecanismos de reporte e comunicação externa e
- Implementação do Gerenciamento de riscos - compreende a implementação do *framework* e dos processos de gerenciamento de riscos, o monitoramento, a revisão e a melhoria contínua deste *framework*.

Os processos de gerenciamento de riscos devem ser parte integrante do gerenciamento da organização, assim como devem estar integrados ou embutidos na cultura e nas praticas da organização bem como adaptados aos processos de negócio existentes [37].

O processo da gestão de riscos, se baseia nas seguintes atividades [37]:

- **Estabelecimento do Contexto** - compreende um estudo e estabelecimento do contexto do ambiente externo e interno da organização. Inclui variáveis políticas, econômicas, regulatórias, culturais, estruturais e políticas da organização, assim como sistemas de informação, normas e padrões. Também inclui a definição do objetivo da gestão de riscos, metas, responsabilidades, escopo, metodologia de avaliação de riscos, forma de medir o desempenho e definição dos critérios de riscos;
- **Avaliação dos Riscos** - compreende a identificação dos riscos sua análise e avaliação;
- **Tratamento dos Riscos** - Compreende a avaliação das alternativas de tratamento dos riscos, o nível de tolerância dos riscos residuais e a avaliação da eficácia do tratamento escolhido;
- **Monitoramento e revisão** - Realizado para monitorar e acompanhar os processos e riscos. Assegura que os controles estão funcionando bem, verifica mudanças no contexto interno ou externo que podem afetar os riscos que estão sendo gerenciados e fica em alerta para o surgimento de novos riscos e registra as lições aprendidas;

- **Comunicação e consultoria** - Compreende os requisitos relevantes internos e externos, assim como da equipe que vai auxiliar na definição do contexto dos riscos. Assegura que diferentes pontos de vista sejam utilizados para definir os critérios de riscos e sua avaliação e obtém o apoio para plano de tratamento dos riscos;
- **Registro do processo de gerenciamento de riscos** - Refere-se aos documentos gerados e à definição de acesso. Quem terá acesso às informações e a definição de quais informações são sigilosas, necessidades de *compliance*, período de retenção dos registros e etc.

Embora a norma internacional forneça orientações genéricas, não se destina a promover a uniformidade de gestão de risco entre as organizações. A concepção e implementação de planos de gestão de riscos e estruturas da norma visam levar em conta as diferentes necessidades de uma organização específica, os seus objetivos específicos, de contexto, estrutura, operações, processos, funções, projetos, produtos, serviços ou bens e práticas específicas utilizadas. Além disto, as orientações normativas descritas no texto da norma podem ser utilizadas para harmonizar os processos de gestão de riscos em outras normas e padrões. O texto fornece ainda, uma abordagem comum para apoiar normas de riscos ou setores específicos, não substituindo tais normas [1].

2.4.2 ABNT NBR ISO/IEC 31010

A norma NBR ISO/IEC 31010:2012 é um padrão de apoio para a ISO/IEC 31000:2009 e fornece orientação sobre a seleção e aplicação de técnicas e ferramentas sistemáticas de avaliação de riscos, conforme exibido no Quadro 2.2.

A avaliação de risco realizada em conformidade com esta norma contribui para outras atividades de gestão de riscos. No Quadro 2.2 a aplicação de uma gama de técnicas é introduzida, com referências específicas a outros padrões internacionais, onde o conceito e aplicação de técnicas são descritos em maiores detalhes. A norma não se destina à certificação, regulamentação ou uso contratual. Não prevê critérios específicos para identificar a necessidade de análise de risco, nem especifica o tipo de método de análise de risco que são necessários para uma aplicação particular.

A importância da norma está em indicar para cada atividade da gestão de riscos, técnicas que possam ser aplicáveis, fortemente aplicáveis e não aplicáveis de acordo com a atividade, fornecendo ao gestor de risco informações plausíveis do correto uso das técnicas. Porém a norma não cita, nem invalida as demais técnicas disponíveis para gestão de riscos [3].

Quadro 2.2: Ferramentas utilizadas para o processo de avaliação de riscos. Fonte: [3].
(continua)

	Processo de Avaliação de riscos				
	Identificação de Riscos	Análise de Riscos			Avaliação de Riscos
		Consequência	Probabilidade	Nível de risco	
Brainstorming	FA	NA	NA	NA	NA
Entrevistas estruturadas ou semi estruturadas	FA	NA	NA	NA	NA
Delphi	FA	NA	NA	NA	NA
Listas de verificação	FA	NA	NA	NA	NA
Análise preliminar de perigos (APP)	FA	NA	NA	NA	NA
Estudo de perigos e operabilidade (HAZOP)	FA	FA	A	A	A
Análise de Perigos e pontos críticos de controle (APPCC)	FA	FA	NA	NA	FA
Avaliação de risco ambiental	FA	FA	FA	FA	FA
Técnica estruturada "e se"SWIFT	FA	FA	FA	FA	FA
Análise de cenários	FA	FA	A	A	A
Análise de impactos de negócio	A	FA	A	A	A
Análise de causa-raiz	NA	FA	FA	FA	FA
Análise de modos de falha e efeito	FA	FA	FA	FA	FA
Análise de árvore de falhas	A	NA	FA	A	A
Análise de árvore de eventos	A	FA	A	A	NA
Análise de causa e consequência	A	FA	FA	A	A
Análise de causa e efeito	FA	FA	NA	NA	NA
Análise de camada de proteção (LOPA)	A	FA	A	A	NA
Árvore de decisões	NA	FA	FA	A	A

Quadro 2.2: Ferramentas utilizadas para o processo de avaliação de riscos. Fonte: [3].
(continuação)

	Processo de Avaliação de riscos				
	Identificação de Riscos	Análise de Riscos			Avaliação de Riscos
		Consequência	Probabilidade	Nível de risco	
Análise da confiabilidade humana	FA	FA	FA	FA	A
Análise Bow tie	NA	A	FA	FA	A
Manutenção centrada em confiabilidade	FA	FA	FA	FA	FA
Análise de circuitos ocultos	A	NA	NA	NA	NA
Análise de Markov	A	FA	NA	NA	NA
Simulação de Monte Carlo	NA	NA	NA	NA	FA
Estatística Bayesiana e redes de Bayes	NA	FA	NA	NA	FA
Curvas FN	A	FA	FA	A	FA
Índices de risco	A	FA	FA	A	FA
Matriz de probabilidade e consequência	FA	FA	FA	FA	A
Análise de custo/benefício	A	FA	A	A	A
Análise de decisão por Multicritérios (MCDA)	A	FA	A	FA	A
FA - Fortemente aplicável. NA - Não aplicável. A- Aplicável.					

2.4.3 ABNT NBR ISO/TR 31004

O relatório NBR ISO/TR 31004:2015 é um guia para as organizações realizarem uma gestão de risco efetiva por meio da implementação da ABNT NBR ISO 31000:2009. Essa implementação é customizada e adaptada a cada organização. Adicionalmente, o relatório fornece uma explicação dos principais conceitos, princípios e o modelo de gerenciamento de riscos da norma 31000. Este relatório pode ser aplicado a todas as atividades e todas as partes da organização. A norma enfatiza que todas as organizações gerenciam riscos em alguma extensão. Independente do motivo para a implantação da ISO 31000, espera-se possibilitar que a organização consiga gerenciar melhor seus riscos, identificando como já está gerenciando. O processo de implementação, avalia os arranjos existentes, adaptando e modificando, se necessário, para alinhá-los com a ISO 31000. [2].

2.5 Riscos em Metodologias Ágeis

Um dos grandes riscos no desenvolvimento de *software* está atrelado à mudança de requisitos. A qualidade dos requisitos do sistema podem não encontrar a correteza, a completude e a consistência, e assim a próxima fase do processo de desenvolvimento pode ser atrasada [53]. No modelo iterativo de desenvolvimento os usuários podem prover os requisitos de forma incremental [53]. O risco de desenvolvimento de *software* pode ser altamente reduzido, através de protótipos que criam um canal de comunicação com os usuários [53]. Os protótipos podem ajudar no reconhecimento de requisitos incompletos, inconsistentes ou incorretos [53].

No *Scrum*, os riscos referentes ao levantamento de requisitos, podem ser atenuados pelo fato do usuário fazer parte da equipe. O dono do produto (PO), entende o progresso do desenvolvimento, testando e auditando os requisitos da nova versão. Todos os dias nas reuniões diárias de quinze minutos é estabelecida uma comunicação efetiva entre cliente e time de desenvolvedores. Adicionalmente as histórias de usuários, reduzem a complexidade do *software* e o PO verifica os entregáveis documentais da próxima iteração enquanto valida o *software* entregue na iteração corrente [53].

A análise da arquitetura do projeto necessita ser pensada de forma antecipada, assim como as dependências técnicas para determinar quais componentes são necessários antes do início da implementação das histórias de usuários [77]. Também aconselha-se controlar os débitos técnicos, ligados às lacunas tecnológicas, imperfeições estruturais, diferentemente das que podem ser analisadas por ferramentas de verificação de código [77]. Além disto, é recomendável medir também as opções através da análise de custo VPL (valor presente líquido), pois segundo o autor as previsões econômicas, apesar de imprecisas,

por vezes são mais convincentes do que dogmas ágeis, princípios de *design* genérico ou experiências passadas [77].

Os projetos ágeis distribuídos possuem alguns fatores de risco, tais como: distância espacial, distância temporal (fuso-horários), barreira da língua, cultura de trabalho e um projeto com um escopo grande. Todos estes fatores geralmente, são arriscados quando se utiliza métodos ágeis, onde a colaboração face a face é um requisito primordial [97]. Nesta situação, as cerimônias do *Scrum* não são realizadas com a devida frequência e quando ocorrem levam mais que o tempo recomendado de 15 minutos, muitas ficam entre 45 a 60 minutos e comprometem a agilidade [97]. Não obstante, o PO está fora do ambiente de desenvolvimento comprometendo a comunicação, assim como o *Scrum Master* que também não consegue capturar rapidamente os impedimentos da equipe [97].

Pesquisas indicam que de forma geral, as principais limitações no uso dos métodos ágeis são: a falta de uma fase de planejamento, restrições orçamentárias, documentação insuficiente, desobediência aos passos do ciclo de vida de desenvolvimento do *software*, falta de previsibilidade, excesso de reuniões, atendimento apenas regular aos requisitos de conformidade, necessidade de capacitação e por fim, não é recomendado para pequenas organizações [5].

Além disto, alguns métodos ágeis não estão seriamente comprometidos com os usuários finais. Eles focam na entrega de valor ao cliente, e não na entrega de qualidade para os usuários finais [54]. O papel de dono do produto - PO, deveria expressar as necessidades do usuário final, enquanto responsável pelas necessidades do cliente [54]. Porém, muitas vezes este papel tem sido delegado a pessoas do time de desenvolvimento, ao invés de um representante da população de usuários ou o próprio cliente [54].

Fatores como cultura organizacional são importantes quando se fala em métodos ágeis. Pode-se assumir que quanto mais formalizado um método ágil se tornar, mais cedo será considerado disfuncional em organizações com forte cultura em desenvolvimento [56]. Culturas organizacionais hierárquicas não favorecem a implantação de metodologias ágeis [56]. Uma estrutura de matriz funcional não provê a autoridade necessária ao *Scrum Master* para a retirada dos impedimentos, fazendo com que ele necessite negociar com o gerente funcional das demais áreas para priorizar o atendimento de certas demandas, quebrando a agilidade da equipe.

Para ilustrar os riscos que comumente ocorrem em projetos de desenvolvimento de software e como eles podem ser reduzidos com os métodos ágeis, o Quadro 2.3 demonstra que ao utilizar as metodologias ágeis, a maioria dos impactos dos riscos são minimizados. Como pode-se verificar apenas o risco de inadequação de *design* do software poderá ter um impacto aumentado utilizando métodos ágeis, isso se deve ao fato que o *design* não é pensado de forma geral e sim a cada iteração, sendo criado de forma contínua, pois o

Quadro 2.3: Riscos comuns de Software. Adaptado de Albadarneh et al. (2015) [6]

Riscos comuns no processo de desenvolvimento de software	Impacto ágil sob o risco
Escopo que se arrasta	Reduz
Cronograma extremamente otimista	Reduz
Requisitos aprimorados ou desenvolvedores aprimoradores (gold-plating)	Reduz
Qualidade alterada	Reduz
<i>Design</i> (no sentido de projetizar/arquiteturar) inadequado	Provavelmente aumenta
Problemas entre desenvolvedores e clientes	Reduz
Risco de orçamento	Reduz
Custo de cancelamento	Reduz
Erro de requisito	Reduz
Risco tecnológico	Reduz
Risco de segurança	Reduz
Baixa competência técnica da equipe	Indiferente
Falha do contratado	Indiferente

Scrum, não cobre o ciclo de vida completo do desenvolvimento de software [6]. Porém na Iteração Zero no XP e no estudo do negócio no DSDM, a criação de um *design* com a visão do todo, seria possível [6]. Outros fatores de risco como: baixa competência técnica da equipe e falhas que podem ocorrer no contrato não são minimizados pelo simples fato de se utilizar métodos ágeis, devendo então estes riscos serem tratados como em qualquer outro método de desenvolvimento. Porém, mais de 75% dos riscos levantados no Quadro 2.3, são reduzidos com o uso dos métodos ágeis pelo fato do escopo e do prazo serem fixados na *Sprint* e com a comunicação diária da equipe provendo transparência, torna-se difícil esconder uma situação de risco [6]. No caso do risco: "escopo que se arrasta", ou seja, quando há aumento do escopo, este irá para o *backlog* e necessitará ser priorizado pelo cliente. Cronogramas otimistas demais podem ser corrigidos com um *feedback* do tempo das entregas realizadas em poucas semanas, isto irá ajustar a velocidade da equipe e as possíveis solicitações de mudanças também serão poucas se comparadas a pacotes de meses de trabalho.

A qualidade é flexível no ágil, podendo ser negociada, assim como aumenta a interação entre clientes e desenvolvedores criando uma maior familiaridade e reduzindo entraves pessoais [6]. O orçamento é de fácil controle pois são apenas estimativas, assim como o custo do cancelamento, pois a medida que se tem entregas mais rápidas e contínuas fica evidente o aumento do custo e assim a qualquer momento o trabalho poderá ser parado, diferentemente de um método tradicional onde se demorariam meses até a primeira entrega para que então fosse realizada tal avaliação. Erros de requisitos poderão ser descobertos durante a implementação (reuniões diárias) e na reunião de revisão onde é realizada a apresentação da unidade funcional do software para o cliente. Nos métodos ágeis problemas tecnológicos e de segurança são descobertos mais cedo, pois a cada iteração se tem um pedaço do produto sendo testado em um intervalo de poucas semanas, assim, os problemas são descobertos e tratados mais cedo [6].

2.6 Gestão de Riscos em Projetos de Desenvolvimento de Software

Conforme levantamento intitulado de Chaos Manifesto 2013, realizado pelo *The Standish Group*, apenas 39% dos projetos de *software* são entregues no prazo e dentro do orçamento [42]. Uma parte significativa das falhas podem ser atribuídas a um gerenciamento insatisfatório de riscos [11]. Apesar do fato de muitas organizações terem investido tempo, dinheiro e esforço para desenvolver o seu *software*, o fracasso de muitos projetos de *software* ainda é frequente. Um risco de *software* pode aumentar a taxa de falha de um projeto se ele for ignorado. Assim, o principal objetivo da gestão de riscos é identificar problemas gerenciais e técnicos antes que eles ocorram para que ações possam ser tomadas afim de eliminar ou mitigar o seu impacto [44].

Apesar do conhecimento sobre os riscos e sua gestão serem altamente relevantes, 75% dos gerentes de projetos não seguem nenhuma técnica de gestão de riscos dentro do gerenciamento de projetos [11].

A gestão de risco implica em: avaliar a importância de um risco (por meio da sua probabilidade de ocorrência e seu impacto sobre o desempenho do projeto) além do desenvolvimento de estratégias para controlá-lo [44]. O gerenciamento de riscos depende da percepção e reconhecimento das fontes de riscos que podem ser identificadas durante todo o projeto. Sob todas as perspectivas negociais, o sucesso de muitas organizações tem se tornado dependente do sucesso ou da falha do *software* construído, neste contexto, gerenciar riscos não soa apenas como uma prática de desenvolvimento, mas uma prática vital de negócios [100].

Porém, a maioria dos desenvolvedores de *software* e gerentes de projeto, veem o processo e atividades de gestão de risco como trabalho extra e despesa. Muitas vezes, o processo de gestão de risco é a primeira coisa a ser removida do projeto quando o cronograma atrasa [52]. A cultura de espírito livre em muitas empresas de desenvolvimento de *software* está em conflito com a quantidade de controle, muitas vezes necessária, para desenvolver sistemas de *software* complexos de uma forma disciplinada. Além disso, muitos profissionais de desenvolvimento de *software* entendem a gestão de risco e controle como uma forma de inibição da criatividade [52].

Segundo a base *Web of Science*, o artigo mais citado quando se fala de gerenciamento de riscos na área de ciência da computação e engenharia de *software* é o de Boehm (1991) que discorre sobre os princípios e práticas para gerenciar riscos antes que eles causem a ruína do projeto [18]. Neste mesmo artigo, o autor já utilizava o critério de exposição ao risco, que hoje é bastante difundido e conhecido por matriz de probabilidade e impacto e já se definiam as atividades da gestão de riscos como: identificação, avaliação e monitoramento,

que hoje são claramente atividades identificadas na norma internacional ISO 31000.

No guia de gerenciamento de riscos contínuos, criado pelo SEI (*Software Engineering Institute*), em 1996, há um modelo de gestão de riscos. O propósito foi colocar nas mãos da comunidade um livro onde seria possível executar o gerenciamento de riscos dentro dos projetos. O guia é uma prática genérica com uma variedade de métodos e técnicas à escolha. Isso significa que o modelo pode ser adaptado para determinada organização e projeto [35].

O modelo de desenvolvimento de *software* em espiral, concentra muitos riscos de desenvolvimento e pode ter uma alta flexibilidade [53]. Tal modelo propõe de forma explícita uma análise e resolução dos riscos a cada volta da espiral, ou seja a cada iteração do processo de desenvolvimento de *software* [17].

Outros autores propuseram um *framework* para identificação de riscos em projetos de *software* sob duas dimensões: uma relacionada à exposição ao risco (probabilidade x impacto) a outra relacionada ao nível de controle, que representa o grau em que o gerente de projeto percebe que suas ações podem evitar a ocorrência do risco. Na pesquisa, os riscos não são tratados individualmente, e sim agrupados em quatro quadrantes: cliente, escopo/requisitos, ambiente e execução. Fica claro que para o autor, os riscos relacionados ao comportamento dos clientes e ao escopo são altos, pois estão fora do controle do gerente de projeto. Efetivamente, não há controles que impeçam a mudança de escopo, pois os requisitos são voláteis. Porém quando comparado aos controles sobre o ambiente e a execução, estas variáveis são praticamente fixas, permanecendo sem grandes alterações do início ao fim do projeto [49].

Os riscos percebidos como altos na maior parte das vezes, estão fora do controle direto do gerente de projeto [49]. Talvez por isso, autores como Boehm (1991), não tenham citado nenhum risco relacionado a estes fatores comportamentais do usuário, pela falta de controle exercida do gerente sob eles, porém os fatores humanos e culturais devem ser levados em consideração na gestão de riscos, pois fazem parte dos princípios da norma ISO 31000 (2009) [18], [1].

Desde 1988, iniciando formalmente com o modelo que previa a gestão de riscos explicitamente no modelo de Boehm, até este momento, vários autores discorreram sobre a gestão de riscos em projetos de softwares [11] [83] [101]. Dado o custo e as perdas potenciais em projetos de softwares com falhas, os pesquisadores e profissionais devem aprender uns com os outros para reduzir falhas de projetos.

2.7 Técnicas e Ferramentas da Gestão de Riscos Aplicadas a Projetos de Software

Sabe-se que na NBR ISO/IEC 31010, há uma gama de técnicas e ferramentas que poderiam ser utilizadas em cada atividade da gestão de riscos. Porém neste trabalho, a ideia é utilizar as técnicas que são mais aplicáveis a gestão de riscos em projetos de TI. Mais especificamente, o Quadro 2.4 analisa as principais técnicas e ferramentas que vêm sendo utilizadas na gestão de riscos de projetos ágeis de desenvolvimento de software, referente às atividades: identificação, análise e avaliação, seguindo o padrão da ISO/IEC 31010.

Quadro 2.4: Técnicas e Ferramentas Utilizadas nos Modelos de Gestão de Riscos integrados ao Desenvolvimento de Software. Fonte: Elaboração própria.

Técnica / Ferramenta	Identificação	Análise	Avaliação
Entrevista semi-estruturada	[40], [97], [16], [24], [41]	[97]	————
Questionário	[97], [16], [86]	[97]	————
Observação	[24]	————	————
Grupo Focal on line	[60], [41]	————	————
Formulário	[70]	————	————
Revisão de Literatura com base em artigos científicos	[23], [57], [65], [19]	————	————
Base histórica de riscos típicos – base própria ou compartilhada	[50],[28],[63]	————	————
Reunião com equipe - Brainstorming	[58], [7],[19],[20]	[58],[19]	[19]
WORKSHOPS	[34]	————	————
SWOT	————	[86], [65]	————
KANBAN	[19]	[34], [19]	[34], [19]
Rede Bayesiana	[72]	————	————
FMEA	————	[63]	[63]
Matriz de riscos (probabilidade e impacto)	————	[9],[86],[7],[51]	[9],[86],[7],[51]
Livre – não citou ferramentas	[9], [69], [48], [89]	[40],[70],[72], [60],[16], [23],[50],[24], [57],[69], [41], [34],[38], [48],[20], [28],[89]	[40],[70], [72], [60], [16], [23], [50], [24],[57], [69],[58],[41], [34],[65],[38], [48],[20], [28], [89]

O Quadro 2.4, consolida uma pesquisa bibliográfica realizada para levantamento das técnicas e ferramentas mais aplicadas às atividades de: identificação, análise e avaliação de riscos em projetos de software. A pesquisa foi realizada utilizando artigos das bases de conhecimento: "*Web of Science*" e "*IEEE Xplore*". Foram realizadas pesquisas em ambas as bases, com os filtros: "*Risk Management*" e "*Agile*", referente aos artigos publicados nos últimos 10 anos (de 2006 a 2016). Todos os artigos retornados foram analisados e aqueles que propunham algum modelo ou processo de gestão de riscos em conjunto com os métodos ágeis foram considerados para efeito de composição do Quadro 2.4.

Para a atividade de identificação de riscos as técnicas mais recorrentes nos artigos foram: entrevistas, revisão de literatura, base histórica e *Brainstorming*.

- Entrevista semi-estruturada: forma de capturar dados realizada pelo entrevistador com base em um rol de perguntas básicas, tornando-a mais flexível do que apenas um jogo de perguntas e respostas. É fundamental que o entrevistador possua capacidade de ouvir atentamente, estimule o entrevistado a falar e garanta o anonimato, preservando a identidade do entrevistado;
- Revisão de literatura: baseado em pesquisas realizadas em artigos científicos publicados na área de riscos de TI. Por meio dos artigos, é possível elencar vários riscos comuns em projetos de software;
- Base histórica: base histórica mantida pela própria organização ou compartilhada de outra organização, com a lista de riscos típicos ou comuns aos seus projetos de software;
- *Brainstorming*: Reunião informal realizada com a equipe do projeto para colher diversos pontos de vista sobre o mesmo assunto, de forma a não expressar julgamentos de certo e errado, fomentando a liberdade de expressão das ideias dos participantes.

Para as atividades de análise e avaliação de riscos, uma ferramenta muito citada nos artigos, foi a matriz de risco, baseada na probabilidade e impacto do risco. Outra ferramenta também citada foi o uso do *Kanban*, *SWOT* (Forças, Fraquezas, Oportunidades e Ameaças) e do *brainstorming* para análise dos riscos.

Para as atividades de tratamento e monitoramento dos riscos, não foram encontrados artigos que definissem técnicas ou ferramentas. A maioria dos artigos se focaram em propor modelos de gestão de riscos integrado aos métodos ágeis e deixaram as ferramentas a serem utilizadas, à livre escolha da organização, principalmente no que tange as atividades de análise e avaliação de riscos, tão pouco houve definição para as atividades de tratamento e monitoramento. A pesquisa evidenciou que ao buscar a flexibilidade e possível facilidade de adaptação às organizações, os modelos não indicaram "como" a gestão de riscos poderia ser realizada em sua totalidade, criando então, modelos muito genéricos.

2.8 Modelos de Gestão de Riscos aplicados às Metodologias Ágeis

Nesta seção são apresentados cinco modelos de gestão de riscos aplicados aos métodos ágeis. Estes cinco modelos foram escolhidos por sua atualidade, nenhum possui mais de quatro anos desde sua publicação e se destacaram por demonstrarem como poderiam se integrar de fato aos métodos ágeis.

2.8.1 Modelo de Gestão de Riscos Segundo Andrat e Jaswal (2015)

Conforme o Quadro 2.5, as metodologias ágeis possuem vantagens sobre as metodologias tradicionais, no que diz respeito a alta adaptabilidade, envolvimento do usuário e baixo custo, porém os riscos são desconhecidos.

Quadro 2.5: Comparação entre métodos ágeis e tradicionais. Fonte: [8]

Parâmetros	Ágil	Tradicional
Documentação	baixo	alto
Adaptabilidade	alto	baixo
Envolvimento do Usuário	alto	baixo
Custos	baixo	alto
Riscos	desconhecidos	bem compreendidos

Nos métodos ágeis, é essencial a adoção das atividades de gestão de riscos uma vez que, as fases do desenvolvimento de software se sobrepõem, ao contrário dos métodos tradicionais, o que pode levar a um negligenciamento dos riscos [8].

Também foi proposta pelos mesmos autores, uma pirâmide de risco, conforme Figura 2.11 para auxiliar na atividade de avaliação dos riscos, de forma ágil, visual e transparente a todos envolvidos no projeto. O eixo "x" representa o impacto (que pode ser negativo ou positivo) e o "y" a probabilidade [8].

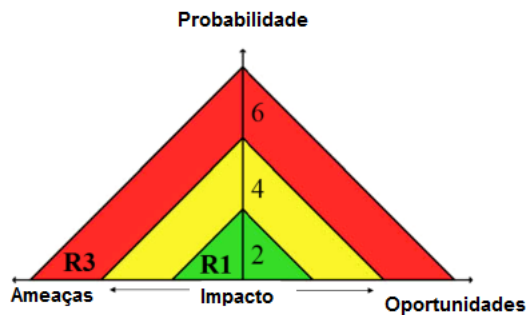


Figura 2.11: Pirâmide de Risco. Fonte: [8].

As cores indicam a prioridade: vermelho para alta, amarela para média e verde para baixa. Assim R3 possui prioridade maior do que R1, enquanto R3 precisa ser evitado o risco R1 nem mesmo precisa ser tratado, por ser de baixa probabilidade e impacto.

O mesmo estudo também propõe a criação de um diagrama de rede, exibido na Figura 2.12, para riscos interdependentes, onde, os riscos que mais influenciam em outros riscos, devem ser tratados prioritariamente. Por exemplo, o risco 1, causa os riscos 5 e 6, enquanto o risco 2 só causa o risco 6. Assim o risco 2 é menos prioritário que o tratamento ao risco 1.

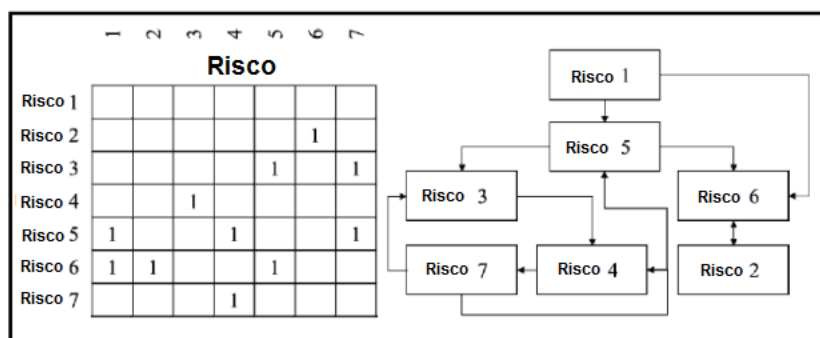


Figura 2.12: Diagrama de Rede dos Riscos. Fonte: [8].

O modelo de pirâmide de risco utilizado na análise de risco, às vezes pode fornecer avaliações de risco com resultados incorretos, assim para superar essa limitação, o diagrama de rede de risco ágil foi proposto como uma abordagem alternativa. Ajuda a melhorar compreensão do impacto de cada risco identificado. E também fornece a relação entre os riscos prevacentes e ajuda na identificação da ordem em que múltiplos riscos possam ser eliminados ou reduzidos simultaneamente.

2.8.2 Modelo de Gestão de Riscos Segundo a Extensão de Software do PMI - 5º Edição

Para a disciplina de gestão de riscos a extensão de software do PMI, traz um modelo de gestão aplicável a métodos ágeis, ou adaptativos. Conforme Figura 2.13, observa-se que as atividades que compõem a avaliação de riscos (identificação, análise e avaliação) são realizadas, já de posse do backlog do produto, ou seja das estórias de usuário já priorizadas pelo PO [76].

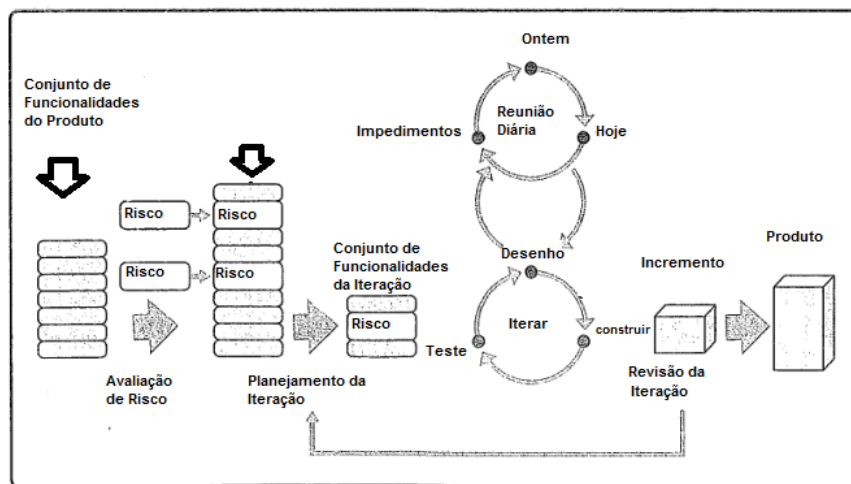


Figura 2.13: Gestão de Riscos para Métodos Adaptativos. Fonte: Adaptado de PMI, IEEE Computer Society (2013) .

De posse do *backlog* do produto, o gerente do projeto pode realizar uma avaliação dos riscos junto ao time. E como forma de evitar ou mitigar a ocorrência ou impacto destes riscos no projeto, os riscos de mais alto nível de criticidade podem ser tratados primeiro. As reuniões diárias servem para o time monitorar e identificar novos riscos. A reunião de revisão pode trazer novos riscos técnicos e de cronograma, durante esta reunião mudanças podem ser requeridas pelo cliente, assim planos e repriorização de mitigação dos riscos podem ser necessários. Na reunião de retrospectiva, o time pode fechar riscos e reavaliar a probabilidade e ocorrência daqueles riscos voltarem a ocorrer durante a próxima iteração [76].

2.8.3 Modelo de Gestão de Riscos Segundo Popli e Naresh (2013)

Foi apresentado um modelo para transformar métodos tradicionais de desenvolvimento de software em ágeis [10]. Tal modelo traz um mapeamento para transformar grandes equipes em pequenas equipes, grandes tarefas em pequenas estórias de usuário, longa

iteração em pequenas *sprints*, o ciclo de *feedback* muito longo, em *feedback* instantâneo, entrega tardia em pequenas entregas mais rápidas, longas reuniões em reuniões diárias em pé, teste tardio, em desenvolvimento dirigida a testes e um gerenciamento de projeto do método comando controle, para o método de coordenação efetiva [10].

Além disto explica que no ciclo ágil, a avaliação de riscos é realizada para as histórias de usuário futuras, para que coisas possam ser melhoradas ou levadas para o próximo nível de qualidade. Então é necessário detectar o mais cedo possível as histórias que possuem uma entrega mais arriscada. Isto deve estar claro, antes de se finalizar a história corrente [10].

2.8.4 Modelo de Gestão de Riscos Segundo Cunha et al. (2013)

O modelo proposto para o processo de gestão de riscos ágil, foi baseado na gestão de riscos de Barry Boehm, do SEI e do PMBOK. No processo de Boehm, a gestão de riscos é dividida em avaliação e controle, no modelo de *Continuous Risk Management - CRM* (Gerenciamento de risco contínuo), do SEI, se baseia em 5 fases: identificação, análise, planejamento, monitoração e controle e no centro do modelo está a comunicação. Já o modelo PMBOK, define seis fases: planejamento, identificação, análise qualitativa, avaliação quantitativa, planejamento de resposta e monitoramento e controle [28].

O processo proposto pelos autores, conforme Figura 2.14 possui dois passos.

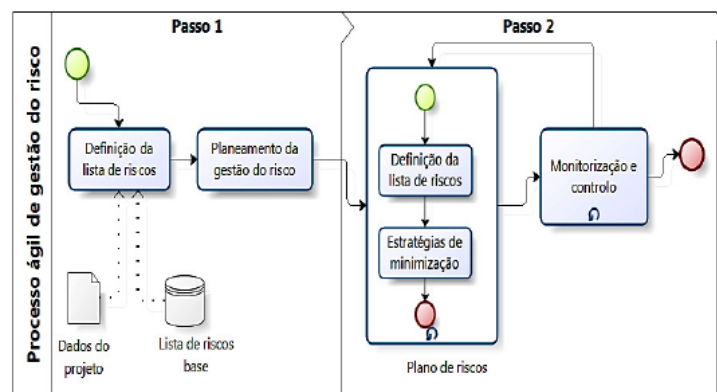


Figura 2.14: Processo de gestão de risco ágil. Fonte: [28].

No primeiro passo, há uma definição da lista de riscos com base em lista de riscos comuns e também a partir de dados do projeto. A partir disto, é realizado o planejamento da gestão de riscos dando origem ao plano de riscos que possui a lista de riscos e as respectivas estratégias de minimização. Realizadas as atividades de monitoração e controle à cada iteração, sendo o plano atualizado na eventualidade de novos riscos. Assim, conforme ilustrado na Figura 2.14 o passo 1 é realizado apenas uma vez no projeto ágil e

o passo 2 realizado a cada iteração [28] . Um dos pontos falhos é que o monitoramento e controle deveria permear todo o processo, pois a qualquer momento poderá existir a necessidade de monitorar riscos de iterações já passadas. Outra informação faltante é a comunicação que não é citada neste processo, apesar de constar em normas da família ISO/IEC 31000 e estar explicitamente citada no CRM do SEI, um dos modelos citados no mesmo artigo. Também existe a lacuna de atividades como análise e avaliação dos riscos neste modelo, que deveriam vir antes das estratégias de minimização constantes no passo 2.

2.8.5 Modelo de Gestão de Riscos Segundo Khatri et al.(2014)

O modelo da Figura 2.15 propõe uma gestão de riscos com base no *Scrum*. O *framework Scrum* possui lacuna estrutural de gestão de riscos, assim o modelo exibido na Figura 2.15 propõe um diagrama de fluxo do processo da gestão de risco ágil, assim durante a reunião de planejamento da *Sprint*, além da esperada análise de esforço das estórias de usuário para embasar os prazos, também seria realizada a análise de risco da *Sprint*.

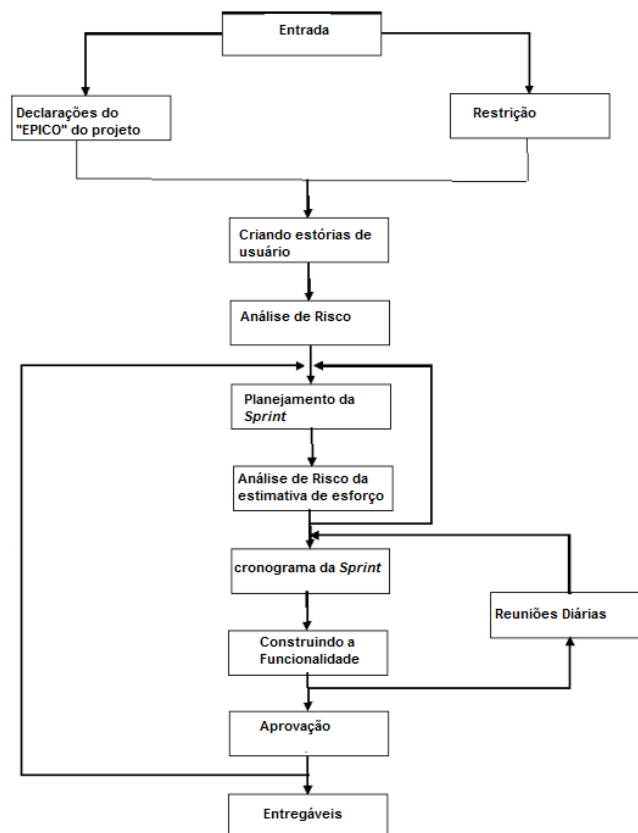


Figura 2.15: Diagrama de fluxo do Processo da gestão de risco ágil. Fonte: [51].

A Figura 2.16, complementa com o modelo integrado ao Scrum, onde cada nível de abstração se preocupa com diferentes tipos de riscos e envolve diferentes papéis ou partes interessadas [51]. Tal processo é controlado por quatro níveis de abstração, vide a seguir:

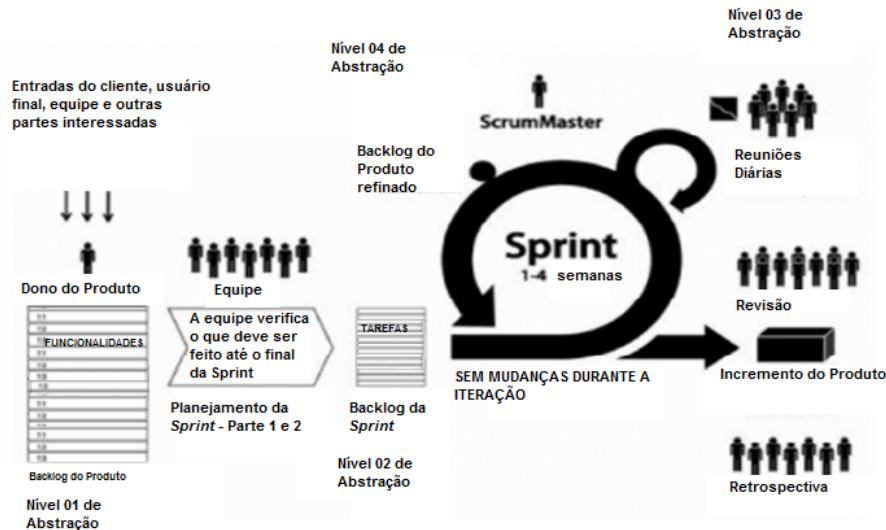


Figura 2.16: Nível de abstração no processo Scrum para análise e documentação do risco. Fonte: [51].

1. Nível 01: Neste nível os requisitos são detalhados e preocupações com o cronograma e riscos relacionados a tecnologia utilizada são capturados. A parte interessada é o dono do produto.
2. Nível 02: Neste nível há um detalhamento dos itens da *Sprint* e assim, riscos relacionados a esforço, novas tecnologias e recursos devem ser decididos e analisados. A parte interessada é a equipe.
3. Nível 03: Riscos em potencial podem surgir oriundos de inexatidão na estimativa de esforço, tempo e recursos humanos. A parte interessada é a equipe e o *Scrum Master*.
4. Nível 04: Ajuda a identificar a correlação entre a necessidade do usuário e o software entregue. Auxilia na identificação de riscos causados por problemas de má comunicação durante o desenvolvimento de funcionalidades críticas. A matriz de risco deve ser preenchida por diferentes partes interessadas em diferentes níveis, devendo toda a análise ser documentada [51].

Este modelo no entanto, não menciona todas as atividades da gestão de riscos, focando apenas na identificação dos riscos em cada nível de abstração. Diferentemente do modelo

proposto na Extensão de software do PMI, que menciona todas as atividades da gestão de riscos e em que momento podem ser executadas no *Scrum*. A similaridade seria no fato de ambos os modelos deixarem em aberto os tipos de técnicas e ferramentas que poderiam ser empregados na execução de cada atividade da gestão de riscos.

O Quadro 2.6, consolida os modelos analisados anteriormente. O modelo mais aderente ao *Scrum* é o modelo sugerido pela Extensão de Software do PMI, uma vez que alinha totalmente as atividades da gestão de riscos ao processo de desenvolvimento do *Scrum*. O modelo de gestão de Riscos proposto por Cunha et al. (2013), também será utilizado, no que tange ao passo 01 de identificação de riscos. Tal modelo, apresenta como proposta de identificação dos riscos em projetos de software o uso da lista de riscos comuns ou *checklist* e já para identificar os riscos específicos do produto, conta com a análise da documentação específica do projeto.

Quadro 2.6: Consolidação dos Modelos de Gestão de Riscos integrados aos Métodos Ágeis. Fonte: Elaboração própria

Modelo/ Autor	Pontos Fortes	Pontos Fracos	Ano
Modelo Segundo Andrat e Jaswal - Pirâmide	Pirâmide intuitiva, facilita levantamento de riscos por meio de brainstorming	Dificuldade em manter um grande número de riscos via pirâmide. Faixas de nível de risco grosseiramente delimitadas, dificuldade na decisão do tratamento do risco quando se tem mais de dez riscos.	2015
Modelo Segundo Andrat e Jaswal - Diagrama de Rede	Ajuda a melhorar compreensão do impacto de cada risco identificado. Auxilia na identificação de que múltiplos riscos possam ser eliminados ou reduzidos simultaneamente	O diagrama se torna proporcionalmente complexo quanto maior for o número de riscos. Demanda grande tempo para sua construção.	
Segundo a Extensão de Software do PMI (5º Edição)	Identificação, análise a avaliação de riscos utilizada para priorizar o backlog. Riscos com alta criticidade são tratados primeiro. Altamente adaptado ao Scrum.	No desenho do modelo as atividades de tratamento e monitoramento dos riscos, não está explícita. Deixa em aberto os tipos de técnicas e ferramentas que poderiam ser empregados na execução de cada atividade da gestão de riscos	2014
Segundo Cunha et al.	No primeiro passo, há uma definição da lista de riscos com base em lista de riscos comuns e também a partir da documentação específica do projeto.	O passo 2 - o monitoramento e controle deveria permear todo o processo, pois a qualquer momento poderá existir a necessidade de monitorar riscos, identificar novos riscos, fechar riscos existentes ou definir ações de contingência. Não cita a comunicação e consulta no processo.	2013
Segundo Khatri et al.	Define níveis de abstração de riscos em cada fase do Scrum. É como se em cada fase mudasse o foco da visão de riscos.	Deixa em aberto os tipos de técnicas e ferramentas que poderiam ser empregados na execução de cada atividade da gestão de riscos e não cita todas as atividades da gestão de riscos, foco demais apenas na identificação.	2014
Segundo Popli e Naresh	Propõe mapeamento para se transformar métodos tradicionais em ágeis	Não foca em gestão de riscos integrada ao ágil.	2013

A metodologia sugerida nesta dissertação será baseada nestes dois modelos (PMI e Cunha), porém o processo será adaptado para que contemple as ferramentas e técnicas que podem ser utilizadas em cada atividade da gestão de riscos, fruto do resultado do estudo realizado no item 2.7.

Por fim, outros autores também se dedicaram a tentar estabelecer hipóteses de configuração de gestão de risco em ambientes ágeis [66], [97], [68], [13], ressaltando no entanto, o grau embrionário das soluções e da necessidade de colaboração e complementação de seus modelos.

2.9 Legislação Governamental em Gestão de Riscos

2.9.1 Avaliação da Governança de TI na Administração Pública Federal-TC 003.732/2014-2

A avaliação trata-se de relatório de levantamento realizado com o objetivo de acompanhar a situação da Governança de Tecnologia da Informação na Administração Pública Federal, realizado a cada dois anos pelo TCU. É importante frisar que o tema "Gestão de Riscos" foi abordado pela primeira vez em 2014 [93].

Para avaliar a situação de governança de TI na Administração Pública Federal, o TCU tem realizado levantamentos baseados em questionários que abordam práticas de governança e de gestão de TI previstas em leis, regulamentos, normas técnicas e modelos internacionais de boas práticas [93].

No ano de 2016 informações sobre gestão de riscos, foram novamente coletadas, porém os dados apresentados a seguir, dizem respeito ao levantamento de 2014 uma vez que, a análise e o comparativo entre 2014 e 2016, não foi realizada nem divulgada ainda pelo TCU.

O resultado deste levantamento de 2014 mostrou que em relação a gestão de riscos apenas 25% das organizações declaram estabelecer diretrizes para a gestão de riscos de TI, mas apenas 8% adotam a prática integralmente. A implementação parcial da prática, que compreende 17% das organizações, pode estar relacionada à adoção somente para algumas atividades ligadas à TI, como nas contratações de serviços de TI, ou à adoção em apenas algumas unidades da organização, ou ainda à não formalização da gestão de riscos. Com relação à definição do apetite ao risco de TI, os números demonstram que essa ainda é uma prática distante para a Administração Pública, haja vista que apenas 14% a adotam. Desses, 4% o fazem de forma integral, ou seja, somente 13 de 355 organizações avaliadas afirmam definir formalmente os níveis de risco de TI aceitáveis na consecução de seus

objetivos. Coerentemente, apenas 4% das organizações declaram adotar integralmente a prática de tomar decisões estratégicas com base no apetite ao risco [93].

Os números apurados revelam, em geral, que a alta administração das organizações públicas federais ainda não reconhece a importância da gestão de riscos para a consecução de seus objetivos, apesar dos altos valores geridos, em grande parte dos casos, e dos diversos riscos aos quais suas ações estão expostas, em geral. Segundo este mesmo relatório, a principal consequência disso é a ineficácia das ações e o consequente desperdício de dinheiro público, com projetos inacabados ou inviáveis em decorrência de situações que constituíam riscos não considerados quando da tomada de decisão [93].

2.9.2 Conhecimento acerca da Utilização de Métodos Ágeis nas Contratações para Desenvolvimento de Software pela Administração Pública Federal-TC 010.663/2013-4

O resultado da auditoria realizada pelo TCU em algumas instituições públicas federais, seja decorrente da análise teórica sobre metodologias ágeis ou das visitas realizadas pela equipe de fiscalização, seja decorrente do exame dos contratos dessas instituições, possibilitou vislumbrar alguns riscos nas contratações públicas para desenvolvimento de software por meio de métodos ágeis. Nesse sentido, a seguir são apresentados 16 riscos inerentes ao novo paradigma de desenvolvimento de software que está sendo difundido nas contratações da Administração Pública. Para fins didáticos, os riscos elencados estão reunidos em três grupos distintos: processos, produtos e pessoas [94].

Riscos relativos a processos:

- Risco 1: contratação de desenvolvimento de software com adaptação de metodologia ágil que desvirtue sua essência;
- Risco 2: alteração da metodologia ágil adotada no instrumento convocatório no decorrer da execução contratual;
- Risco 3: ausência de definição dos artefatos ou alteração dos artefatos exigidos da contratada no instrumento convocatório durante a execução contratual;
- Risco 4: exigência de artefatos desnecessários ou que se tornam obsoletos rapidamente;
- Risco 5: utilização de contrato para desenvolvimento de software por metodologias tradicionais para desenvolvimento por métodos ágeis;

Riscos relativos a pessoas:

- Risco 6: falta de comprometimento ou colaboração insatisfatória do responsável indicado pela área de negócios (*Product Owner*) no desenvolvimento do software;
- Risco 7: falta do conhecimento necessário do indicado pela área de negócios (*Product Owner*) para o desenvolvimento do software;
- Risco 8: excessiva dependência da visão do indicado pela área de negócios (*Product Owner*);
- Risco 9: equipe da empresa contratada não ter expertise em desenvolvimento de software com métodos ágeis;
- Risco 10: dificuldade de comunicação entre a equipe de desenvolvimento da contratada com o indicado pela área de negócios (*Product Owner*);

Riscos relativos a produtos:

- Risco 11: alteração constante da lista de funcionalidades do produto;
- Risco 12: iniciação de novo ciclo sem que os produtos construídos na etapa anterior tenham sido validados;
- Risco 13: falta de planejamento adequado do software a ser construído;
- Risco 14: pagamento pelas mesmas funcionalidades do software mais de uma vez, em virtude de funcionalidades impossíveis de serem implementadas em um único ciclo, ou em virtude da alteração de funcionalidades ao longo do desenvolvimento do software;
- Risco 15: não disponibilização do software em ambiente de produção para a utilização e avaliação dos reais usuários e
- Risco 16: forma de pagamento não baseada em resultados.

Cumprido frisar que não se trata de enumeração exaustiva de riscos, e sim de um subconjunto identificado com o conhecimento adquirido. Também é importante observar que alguns dos riscos expostos não são inerentes somente ao uso de métodos ágeis, podendo ocorrer também com metodologias tradicionais de desenvolvimento de software. Ao final do relatório, concluiu-se pela viabilidade da adoção de metodologias ágeis em contratações destinadas ao desenvolvimento de software pela APF, assim como outras tantas metodologias que têm sido amplamente utilizadas ao longo dos últimos anos. Como em todo processo de contratação, há riscos que precisam ser considerados e mitigados. Contudo, no caso específico de adoção de métodos ágeis, tratados como novidade no mercado especializado nacional, sobretudo no âmbito da APF, a gestão de riscos inerentes às

características do método merece atenção especial, no sentido de possibilitar que as instituições públicas possam fazer uso das práticas previstas sem incorrer em descumprimento dos normativos vigentes [94].

2.9.3 Instrução Normativa Conjunta MP/CGU N° 01/2016

A Instrução Normativa elaborada pelo Ministério do Planejamento, Orçamento e Gestão (MP) em conjunto com a Controladoria Geral da União (CGU), determina que os órgãos do Poder Executivo Federal deverão adotar medidas para a sistematização de práticas relacionadas à gestão de riscos, aos controles internos, e à governança.

Os controles internos da gestão devem integrar as atividades, planos, ações, políticas, sistemas, recursos e esforços de todos que trabalhem na organização, para assegurar de forma razoável que a organização atinja seus objetivos. Os controles internos e de gestão de riscos, aplicam-se a todos os níveis, unidades e dependências do órgão ou da entidade pública. Sendo que é de responsabilidade dos dirigentes máximos dos órgãos e entidades, assegurar a efetiva implantação destes controles e que façam parte das práticas de gerenciamento de riscos.

Os controles internos da gestão devem ser elaborados e implementados em consonância com quinze princípios descritos na norma e listados a seguir:

1. Aderência à integridade e a valores éticos;
2. Competência da alta administração em exercer a supervisão do desenvolvimento e do desempenho dos controles internos da gestão;
3. Coerência e harmonização da estrutura de competências e responsabilidades dos diversos níveis de gestão do órgão ou entidade;
4. Compromisso da alta administração em atrair, desenvolver e reter pessoas com competências técnicas, em alinhamento com os objetivos da organização;
5. Clara definição dos responsáveis pelos diversos controles internos da gestão no âmbito da organização;
6. Clara definição de objetivos que possibilitem o eficaz gerenciamento de riscos;
7. Mapeamento das vulnerabilidades que impactam os objetivos, de forma que sejam adequadamente identificados os riscos a serem geridos;
8. Identificação e avaliação das mudanças internas e externas ao órgão ou entidade que possam afetar significativamente os controles internos da gestão;

9. Desenvolvimento e implementação de atividades de controle que contribuam para a obtenção de níveis aceitáveis de riscos;
10. Adequado suporte de tecnologia da informação para apoiar a implementação dos controles internos da gestão;
11. Definição de políticas e normas que suportem as atividades de controles internos da gestão;
12. Utilização de informações relevantes e de qualidade para apoiar o funcionamento dos controles internos da gestão;
13. Disseminação de informações necessárias ao fortalecimento da cultura e da valorização dos controles internos da gestão;
14. Realização de avaliações periódicas para verificar a eficácia do funcionamento dos controles internos da gestão; e
15. Comunicação do resultado da avaliação dos controles internos da gestão aos responsáveis pela adoção de ações corretivas, incluindo a alta administração.

A gestão de riscos deverá ser implementada, mantida, monitorada e revisada de forma compatível com a missão e objetivos estratégicos de cada órgão. A norma traz ainda, os princípios da gestão de riscos, seus objetivos e a estrutura do modelo de gestão de riscos que deverá ser observada pelos órgãos.

A norma define ainda que a política de gestão de riscos deverá ser instituída pelos órgãos e entidades do Poder Executivo Federal em até doze meses, a contar da sua data de publicação. O mapeamento e a avaliação dos riscos, deverão considerar as seguintes tipologias de riscos: riscos operacionais, de imagem/reputação do órgão, legais e financeiros/orçamentários. A instrução normativa traz o conceito de *accountability* que é a obrigação dos agentes ou organizações que gerenciam recursos públicos de assumir a responsabilidade por suas decisões e pela prestação de contas de sua atuação de forma voluntária, assumindo integralmente a consequência de seus atos e omissões. Para finalizar, a CGU no cumprimento de suas atribuições institucionais, poderá: avaliar a política de gestão de riscos, avaliar os procedimentos da gestão de riscos, avaliar a eficácia dos controles internos para mitigação, e outras respostas aos riscos elaborados pela organização ou entidade avaliada [64].

2.9.4 Metodologia de Gestão de Riscos de Segurança da Informação e Comunicações - SISP

A Metodologia de Gestão de Riscos de Segurança da Informação e Comunicações do Sistema de Administração de Recursos da Tecnologia da Informação – SISP do Poder Executivo Federal (MGR-SISP), visa padronizar e sistematizar a gestão de riscos de SIC (Segurança da Informação e Comunicações) na APF. Almeja-se assim atingir níveis satisfatórios de SIC e ao mesmo tempo, racionalizar os investimentos, pela priorização de ações e por evitar redundâncias na gestão de riscos [61]. A MGR-SISP está na versão 2.0 e é referente a novembro de 2016, sendo compatível com iniciativas anteriores voltadas à SIC na APF, como:

- Norma Complementar nº 04/IN01/DSIC/GSIPR : Norma do Gabinete de Segurança Institucional da Presidência da República, publicada em 15 de fevereiro de 2013, que estabelece diretrizes para o processo de Gestão de Riscos de SIC (GRSIC) [61];
- Instrução Normativa Conjunta MP/CGU nº 01/2016: publicada em 10 de maio de 2016, pela então CGU e pelo Ministério do Planejamento, Orçamento e Gestão, que dispõe sobre controles internos, gestão de riscos e governança no âmbito do Poder Executivo federal. A partir desta norma é obrigatório a implementação da gestão de riscos na APF, conforme foi explanado no tópico 2.8.3;
- A MGR também está baseada nos conceitos das normas das famílias : NBR ISO 27000 e NBR ISO 31000 e ainda em normas BSI *Standard* 100-2 (2008) e NIST SP 800-39 (2011). Conta ainda com uma infinidade de normas complementares relacionadas com a gestão de riscos, como por exemplo: diretrizes para elaboração de política de SIC, criação de equipe de resposta à incidentes, diretrizes para gestão da continuidade de negócio, diretrizes para gerenciamento de incidentes, dentre tantas outras referenciadas na MGR-SISP [61].

A norma deixa claro que embora seja desenvolvida com foco em segurança da informação e comunicação, ela poderá ser adaptada para realização da gestão de riscos em geral [61].

O Processo da MGR-SISP, conforme Figura 2.17, possui as atividades semelhantes ao processo genérico de gestão de riscos contido na norma NBR ISO 31000:2009.



Figura 2.17: Processo da MGR-SISP. Fonte: [61] .

Além destas atividades existem formas para avaliar o nível de risco. Na Figura 2.18, a linha superior mostra a classificação das probabilidades de risco. Já a coluna à esquerda mostra a classificação dos impactos dos riscos. Os valores interiores representam os níveis de risco estimados em cada situação e combinam os efeitos da probabilidade e do impacto dos riscos. As letras interiores definem as diferentes classes para o tratamento de riscos (MB: Muito Baixo; B: Baixo; M: Moderado; A: Alto; MA: Muito Alto). Cada classe de risco também é sinalizada com uma cor diferente [61].

Probabilidade		Muito Baixa	Baixa	Moderada	Alta	Muito alta
Impacto	Muito baixo	1 (MB)	2 (MB)	3 (B)	4 (B)	5 (M)
	Baixo	2 (MB)	3 (B)	4 (B)	5 (M)	6 (A)
	Moderado	3 (B)	4 (B)	5 (M)	6 (A)	7 (A)
	Alto	4 (B)	5 (M)	6 (A)	7 (A)	8 (MA)
	Muito alto	5 (M)	6 (A)	7 (A)	8 (MA)	9 (MA)

Figura 2.18: Critério de classificação para o tratamento e aceitação de riscos da MGR-SISP. Fonte: [61] .

A norma traz ainda a estratégia de tratamento dos riscos, a depender do nível em que cada um estiver enquadrado pelo quadro apresentado na Figura 2.18. É possível verificar a estratégia de tratamento pelo critério de cada risco [61]:

- **Muito baixo** : risco tolerável nenhuma ação é necessária.
- **Baixo**:risco tolerável, nenhuma ação imediata é necessária porém necessita ser monitorado.

- **Moderado:** situação de atenção. Se possível, o risco deve ser tratado em médio prazo. O risco deve ser monitorado frequentemente. Restrições como custo e esforço podem ser utilizadas para priorizar o seu tratamento.
- **Alto:** risco intolerável e de grande preocupação, devem ser tomadas ações rapidamente e os resultados precisam ser monitorados para verificar se situação mudou com as ações. Recomenda-se o tratamento independente de restrições como custo e esforço.
- **Muito Alto:** intolerável, requer ações de tratamento imediatas. As ações devem ser monitoradas continuamente para avaliar se os efeitos são os esperados. Devem ser tratados independente de restrições como custo e esforço.

A norma traz ainda exemplos e *templates* para: Registro de Profissionais e Papéis, Mapa de Riscos, Plano de Tratamento de Riscos e Plano de Comunicação dos Riscos.

Na próxima seção é apresentada a Metodologia de Pesquisa, explicando como a pesquisa foi desenvolvida, quais métodos, técnicas e procedimentos utilizados para a elaboração deste trabalho.

Capítulo 3

Metodologia de Pesquisa

3.1 Método da Pesquisa

O método é a ordem que se deve impor aos diferentes processos para se atingir um certo fim ou resultado desejado. Entende-se por método o conjunto de processos empregados na investigação e na demonstração da verdade. O método não é inventado, ele depende essencialmente do objeto da pesquisa [22]. Os cientistas cujas investigações foram bem sucedidas tiveram o cuidado de anotar os passos percorridos e os meios que os conduziram aos resultados. Outros depois deles analisaram tais processos e justificaram sua eficácia. Então estes processos empíricos foram gradativamente sendo transformados em métodos científicos [22]. O método não é um modelo, fórmula ou receita, ele é um conjunto ordenado de procedimentos que se mostraram eficientes ao longo da história na busca do saber. Sendo assim, o método é um instrumento de trabalho porém o seu resultado depende do usuário. Isso significa que necessita de bom senso, excluindo das investigações o caráter subjetivo e o acaso, deve adaptar o esforço às exigências do objeto a ser estudado, selecionar os meios e os processos mais adequados. Tudo isso é dado pelo método. Assim, o bom método se torna fator de segurança e economia na ciência. O método científico utiliza a observação, descrição, comparação, a análise e a síntese, além de processos mentais como da dedução e da indução tão necessários à reflexão do cientista [22].

Pode-se assumir, então, que a metodologia científica consiste numa série de atividades sistemáticas e racionais para se buscar, de maneira confiável, soluções para um dado problema [59]. Sendo assim, toda investigação nasce de um problema observado ou sentido, de tal modo que não se pode prosseguir a menos que se faça uma seleção da matéria a ser tratada. Essa seleção requer alguma hipótese ou pressuposição que vai orientar e delimitar o assunto a ser investigado [22].

É necessário distinguir ainda método de técnica. Por método entende-se o dispositivo ordenado, procedimento sistemático de forma geral. A técnica é a aplicação do plano

metodológico e a forma especial de o executar. A técnica está subordinada ao método, sendo sua auxiliar. O método é um conjunto de passos ou etapas (técnicas) que devem ser dados para a realização da pesquisa [22].

A pesquisa por sua vez, é uma atividade voltada para a solução de problemas teóricos ou práticos com o uso de processos científicos [22]. A pesquisa parte de uma dúvida ou problema e com o uso do método científico, busca uma resposta ou solução. Nesta busca, admite-se níveis diferentes de aprofundamento e enfoques específicos, conforme o objeto de estudo, objetivos visados e qualificação do pesquisador [22].

Conforme exibido na Figura 3.1, neste trabalho, a natureza da pesquisa é aplicada, onde o investigador é movido pela necessidade de contribuir para fins práticos mais ou menos imediatos, buscando soluções para problemas concretos [22].

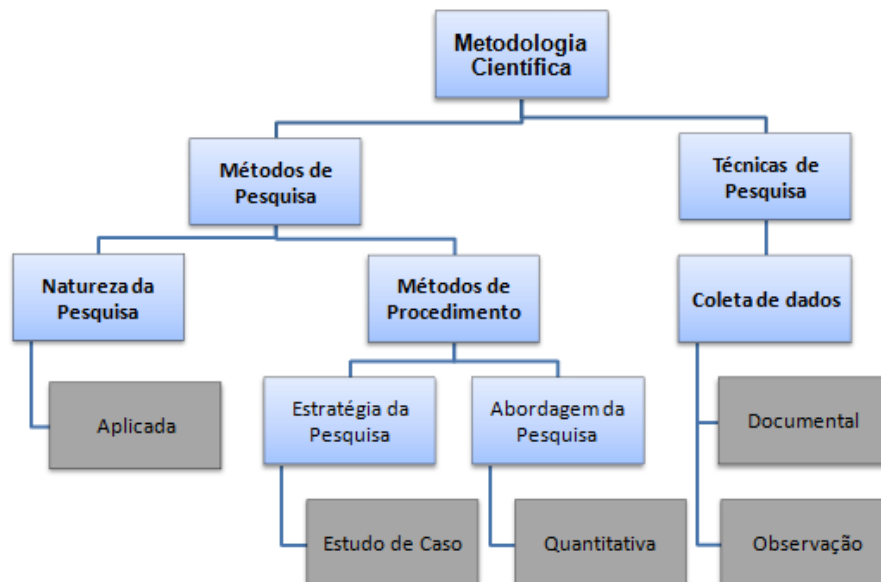


Figura 3.1: Caracterização do método de pesquisa. Fonte: Elaboração própria.

Para isso, adotou-se como estratégia da pesquisa um estudo de caso para investigar o cenário atual em um contexto específico. O estudo de caso visa a realização de uma exploração aprofundada e detalhada, do caso objeto da pesquisa [27]. O objeto desta pesquisa é a CGSI e o estudo de caso está embasado no diagnóstico da situação atual dos seus projetos de software. As análises são realizadas por meio de informações obtidas de amostra referente a 3 anos de ordens de serviços de desenvolvimento de software que foram emitidas à contratada. Ainda são analisados dados do catálogo de sistemas (planilha eletrônica), atualizado no final de 2016. Por tanto, a pesquisa terá uma abordagem

quantitativa, devido a análise destes dados. Da mesma forma, a técnica de pesquisa contará com o levantamento documental de informações (planilhas e consultas em banco de dados) e ainda terá o auxílio da observação participante.

3.2 Estruturação da Pesquisa

Conforme mostra a Figura 3.2 a estrutura da pesquisa está dividida em várias etapas cada parágrafo a seguir discorrerá sobre uma destas etapas.

A pesquisa bibliográfica referencia artigos científicos consultados nas bases: *Web of Science*, *Science Direct* e *IEEE Xplore*. Também são utilizados livros, normas e relatórios técnicos sobre assuntos correlatos à gestão de riscos. Como o objeto deste estudo de caso se trata de um Órgão Público Federal, legislações e normas vigentes foram consultadas para garantir que o resultado deste trabalho esteja alinhado com as recomendações e Instrução Normativa desenvolvida por órgãos como TCU e CGU/MP.

A análise da situação atual dos projetos de software da CGSI é realizada em duas etapas:

1. A primeira etapa : com base nas OSs de desenvolvimento de software, contando com uma amostra de 3 anos de OSs emitidas e concluídas, onde é possível verificar a taxa de atraso dos projetos de software, assim como os índices de não conformidades, possíveis causas-raízes dos atrasos e divergências entre os prazos estimados e efetivamente realizados;
2. A segunda etapa : são analisados dados retirados do catálogo de sistemas e módulos de sistemas da CGSI. Com base nestas análises é possível obter um panorama da situação atual de todos os sistemas da coordenação objeto deste estudo de caso.

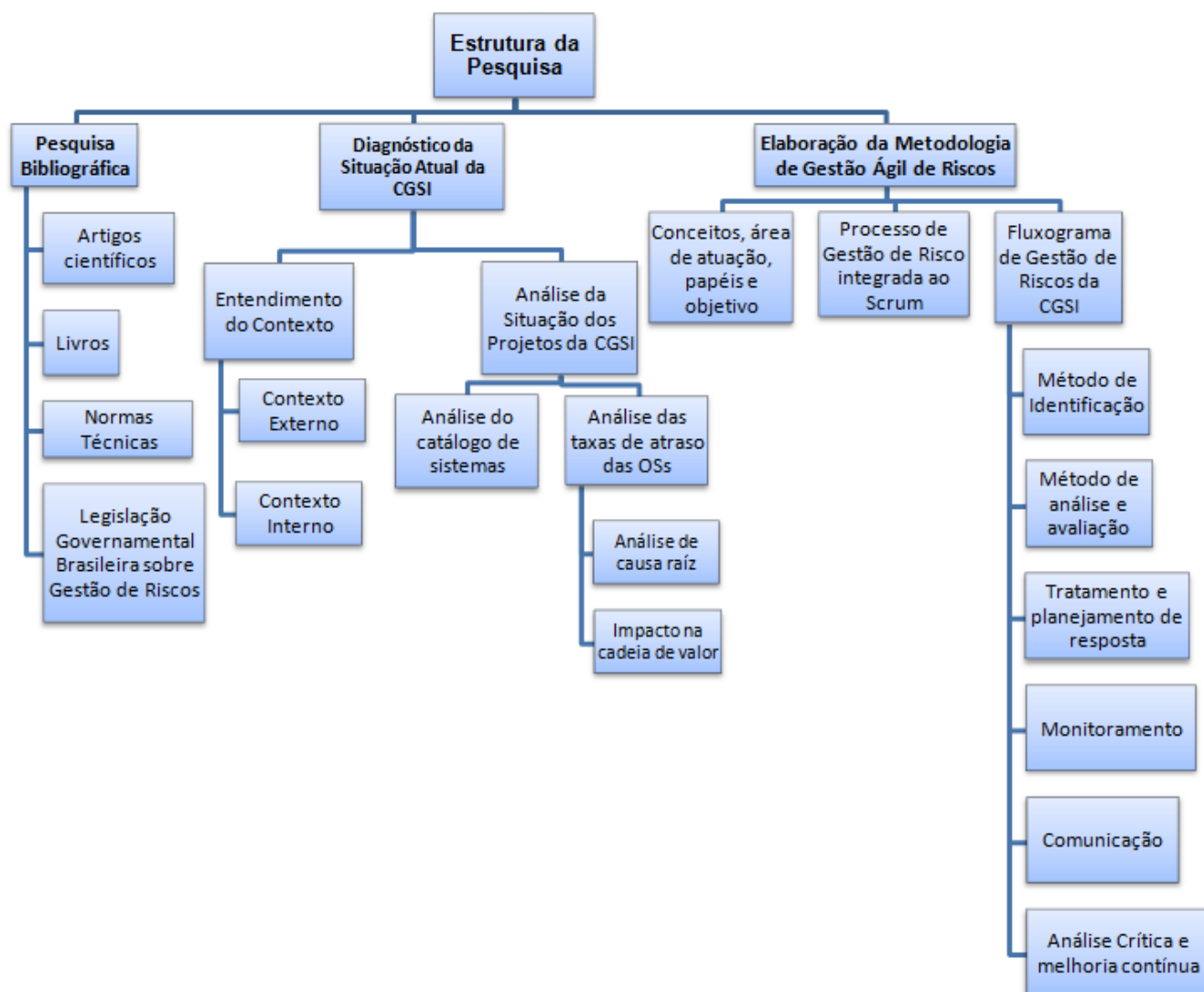


Figura 3.2: Estrutura da Pesquisa. Fonte: Elaboração própria.

A metodologia de gestão de riscos proposta visa integrar a gestão de riscos tradicional com o processo de desenvolvimento ágil de software. Assim, abrange: conceitos, processos, papéis, objetivo, atividades, ferramentas, artefatos e exemplos. São apresentados também os métodos de identificação, análise/avaliação, tratamento, resposta aos riscos, monitoramento, análise crítica e melhoria contínua para o processo da gestão de riscos.

Para a elaboração do *checklist* de riscos, a pesquisa contará com três fontes principais de riscos:

1. Levantamento de dados secundários, onde é possível extrair riscos levantados na análise de *SWOT* constante no PDTI da DTDIE [4];

2. 16 riscos no uso das metodologias ágeis em contratações para desenvolvimento de *software*, listados no relatório do TCU [94];
3. Observação participante - observação do pesquisador exercendo o papel de gerente de projetos durante três anos nesta coordenação.

Após esta etapa de identificação dos riscos comuns aos projetos de software da CGSI, o *checklist* de riscos foi complementado e validado por especialistas de TI de diferentes perfis e áreas, a citar: arquitetos de software, desenvolvedores, analistas de requisitos, de sistemas, de testes e gerentes de projetos, ou seja, profissionais que trabalham em cada fase do desenvolvimento de *software*, auxiliaram na produção da lista de riscos, intitulada de *checklist* de riscos da CGSI.

Esta lista será um guia para identificação de riscos inerentes a todas as disciplinas do desenvolvimento de *software*, não tendo o objetivo de esgotar todos os riscos, mas servir como apoio durante a atividade de identificação dos riscos comuns em projetos de software da CGSI. Esse *checklist* consta do apêndice A deste trabalho.

Na próxima seção, será mostrado o estudo de caso realizado na CGSI onde será analisado o contexto interno e externo do INEP obedecendo o processo indicado na norma ISO/IEC 31000. O entendimento do contexto no qual o objeto do estudo de caso está inserido, é fundamental para a identificação dos riscos, ou seja do perfil de risco do órgão.

Será exibido o diagnóstico da situação atual dos projetos de software e manutenções de sistemas por meio da análise do catálogo de sistemas e das ordens de serviço verificando o valor agregado das entregas dos projetos. Será verificado ainda a taxa de sucesso e atrasos com investigação das possíveis causas. Para atender ao órgão e propor uma metodologia customizada às necessidades do INEP é necessário primeiro entender a situação atual e diagnosticar as principais causas raízes do problema, para em um segundo momento elaborar uma proposta.

Capítulo 4

Estudo de Caso: Diagnóstico da Situação Atual

4.1 Contexto Externo

4.1.1 INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira

O INEP, que em 2017 completa 80 anos, é responsável pela formulação, aplicação e acompanhamento de exames, avaliações e coletas de dados da educação básica e superior do Brasil. O INEP é responsável por exames nacionais como por exemplo, o ENADE (Exame Nacional de Desempenho dos Estudantes) e ENCCEJA (Exame Nacional para Certificação de Competências de Jovens e Adultos). Dentre vários outros, destaca-se o ENEM (Exame Nacional do Ensino Médio), que é a segunda maior prova de acesso ao Ensino Superior do mundo. Sua criação ocorreu por lei em 1937, quando sua denominação era Instituto Nacional de Estudos Pedagógicos. Em 1944 foi lançada a Revista Brasileira de Estudos Pedagógicos (RBEP). Em 1952, assumiu a direção do Instituto o professor Anísio Teixeira, que passou a dar maior ênfase ao trabalho de pesquisa [46]. Em 1972, o INEP foi transformado em órgão autônomo, passando a denominar-se Instituto Nacional de Estudos e Pesquisas Educacionais, que objetivava realizar levantamentos da situação educacional do País [46]. Em 1981 foi lançada outra revista intitulada "Em Aberto", que possuía um caráter técnico e de assessoramento interno ao MEC - Ministério da Educação, sendo modificado, mais tarde, para o atendimento de professores e especialistas fora da estrutura do Ministério [46]. Com o governo da Nova República, em 1985, o INEP passou por um novo desenho institucional. Retirou-se do fomento à pesquisa, para retomar sua função básica de suporte e assessoramento aos centros decisórios do Ministério da Educação [46]. Após o período de dificuldades pelas quais passou no início do governo

Collor, quando quase foi extinto, o INEP iniciou um outro processo de reestruturação e redefinição de sua missão, centrada em dois objetivos:

1. Reorientação das políticas de apoio a pesquisas educacionais, buscando melhorar sua performance no cumprimento das funções de suporte à tomada de decisões em políticas educacionais e
2. Reforço do processo de disseminação de informações educacionais, incorporando novas estratégias de modalidades de produção e difusão de conhecimentos e informações [46].

No início dos anos 90, o INEP atuou como um financiador de trabalhos acadêmicos voltados para a educação. A partir de 1995, houve o processo de reestruturação do órgão. Com a reorganização do setor responsável pelos levantamentos estatísticos, pretendia-se que as informações educacionais pudessem, de fato, orientar a formulação de políticas do MEC [46]. O primeiro passo se deu com a incorporação do Serviço de Estatística da Educação e Cultura (Seec), em 1996, à Secretaria de Avaliação e Informação Educacional (Sediae), do MEC. O Seec, criado em 1937, era um órgão do Poder Executivo, com longa tradição na realização de levantamentos estatísticos na educação brasileira. Em 1997, a Sediae é integrada à estrutura do INEP, passando a existir, a partir desta data, um único órgão encarregado das avaliações, pesquisas e levantamentos estatísticos educacionais no âmbito do governo federal. Nesse mesmo ano, o INEP foi transformado em Autarquia Federal. Nos últimos anos, o Instituto reorganizou o sistema de levantamentos estatísticos e teve como eixo central de atividades as avaliações em praticamente todos os níveis educacionais [46].

Ambiente social

O INEP busca, através da coleta e disseminação de dados, promover estudos, pesquisas e avaliações sobre o Sistema Educacional Brasileiro com o objetivo de subsidiar a formulação e implementação de políticas públicas para a área educacional a partir de parâmetros de qualidade e equidade, bem como produzir informações claras e confiáveis aos gestores, pesquisadores, educadores e público em geral [46]. Para gerar seus dados e estudos educacionais o INEP realiza levantamentos estatísticos e avaliativos em todos os níveis e modalidades de ensino. Alguns sistemas importantes que são elaborados, desenvolvidos e mantidos pelo INEP para o atingimento de suas finalidades estratégicas:

- Censo Escolar: levantamento de informações estatístico-educacionais da educação básica (pública e privada) de âmbito nacional, realizado anualmente;

- Censo Superior: coleta anual de dados do ensino superior no País, incluindo cursos de graduação, presenciais e à distância.
- Avaliação dos Cursos de Graduação: é um procedimento utilizado pelo MEC para o reconhecimento ou renovação de reconhecimento dos cursos de graduação representando uma medida necessária para a emissão de diplomas.
- Sistema Nacional de Avaliação da Educação Superior: Criado pela Lei n 10.861, de 14 de abril de 2004, o Sinaes é um instrumento de avaliação superior do MEC/INEP. Ele é formado por três componentes principais: a avaliação das instituições, dos cursos e do desempenho dos estudantes.
- Exame Nacional do Ensino Médio (ENEM): exame de saída facultativo aos que já concluíram e aos concluintes do ensino médio, aplicado pela primeira vez em 1997.
- Exame Nacional Para Certificação de Competências (ENCCEJA): é uma proposta do Ministério da Educação de construir uma referência de avaliação nacional para jovens e adultos que não puderam concluir os estudos na idade própria.
- Sistema Nacional de Avaliação da Educação Básica (SAEB): pesquisa por amostragem, do ensino fundamental e médio, realizada a cada dois anos.
- ANASEM - Avaliação Nacional Seriada dos Estudantes de Medicina, visa avaliar os estudantes por meio de instrumentos e métodos que considerem os conhecimentos, as habilidades e as atitudes previstas nas diretrizes curriculares nacionais dos cursos de graduação em medicina. Aplicado pela primeira vez em 2016.

Além dos levantamentos estatísticos e das avaliações, censos e exames o INEP promove encontros para discutir os temas educacionais e disponibiliza também outras fontes de consulta sobre educação.

Ambiente político

O INEP está vinculado ao MEC e a outros órgãos no intuito de estabelecer parcerias, como o FNDE, CAPES pois todos são responsáveis pelo atingimento das metas estabelecidas no Plano Nacional de Ensino (PNE), atualmente vigente de 2014 a 2024. O PNE, Lei nº 13.005/2014, é um instrumento de planejamento do Estado democrático de direito que orienta a execução e o aprimoramento de políticas públicas do setor.

Ambiente legal

Com base na Portaria nº. 2.255, de 25 de agosto de 2003, Art. 1, O INEP criado pela Lei n. 378, de 13 de janeiro de 1937, é transformado em autarquia federal vinculada ao

Ministério da Educação, nos termos da Lei nº. 9.448, de 14 de março de 1997, alterada pela Lei nº. 10.269, de 29 de agosto de 2001, tem por finalidades [46]:

- I - Organizar e manter o sistema de informações e estatísticas educacionais;
- II - Planejar, orientar e coordenar o desenvolvimento de sistemas e projetos de avaliação educacional, visando o estabelecimento de indicadores de desempenho das atividades de ensino no País;
- III - Apoiar os Estados, o Distrito Federal e os Municípios no desenvolvimento de sistemas e projetos de avaliação educacional;
- IV - Desenvolver e implementar, na área educacional, sistemas de informação e documentação que abranjam estatísticas, avaliações educacionais, práticas pedagógicas e de gestão das políticas educacionais;
- V - Subsidiar a formulação de políticas na área da educação, mediante a elaboração de diagnósticos e recomendações decorrentes da avaliação da educação básica e superior;
- VI - Coordenar o processo de avaliação dos cursos de graduação, em conformidade com a legislação vigente;
- VII - Definir e propor parâmetros, critérios e mecanismos para a realização de exames de acesso ao ensino superior;
- VIII - Promover a disseminação de informações sobre avaliação da educação básica e superior; e
- IX - Articular-se, em sua área de atuação, com instituições nacionais, estrangeiras e internacionais, mediante ações de cooperação institucional, técnica e financeira, bilateral e multilateral.

Pode-se observar que os itens I e II da lei citam explicitamente as atribuições da DT-DIE, que dizem respeito a manter sistemas e desenvolver projetos de avaliação educacional. Isso fomenta a discussão se a TI do INEP seria área meio ou área fim.

Ambiente tecnológico

Para cumprir sua missão institucional, o INEP conta hoje com infraestrutura de informação complexa, construída com variadas tecnologias. São diversos elementos heterogêneos que compõem a infraestrutura computacional desta Autarquia, agrupados em segurança, rede, telefonia, suprimento energético redundante, banco de dados, servidores,

sistema operacional, sistema aplicativo, armazenamento de dados, cabeamento e suporte. O INEP também conta hoje com aproximadamente 87 sistemas (segundo catálogo de sistemas) construídos em diversas arquiteturas tecnológicas nas linguagens PHP, Java e Aplicativos *Mobile*. A manutenção de toda essa estrutura é realizada por equipe composta de servidores do INEP, servidores temporários, consultores de organismos internacionais (OEI) e colaboradores terceirizados. A constante evolução dos sistemas de informação, somada à demanda interna e externa por novos serviços, faz com que o INEP tenha a necessidade de manter um alto nível de segurança dos dados associado a uma performance cada vez maior dos sistemas. Portanto, para assegurar a qualidade e a confiabilidade dos serviços prestados hoje à sociedade, é necessária a constante evolução de sistemas e programas, aquisições de ferramentas de última geração bem como a correspondente atualização de conhecimentos técnicos [46].

Fatores-chave e tendências que tenham impacto sobre os objetivos da organização

- Mudança de Presidente da República;
- Mudança de Ministro da Educação;
- Mudança de Presidente do INEP;
- Mudança do Secretario Geral de Educação;
- Alteração do PNE;
- Alteração do Decreto de Criação do INEP e
- Mudanças econômicas que reflitam nos investimentos à educação.

Mudanças no cenário político e econômico, podem gerar impactos nos objetivos da organização, seja por mudança na visão de seus dirigentes, ou por cortes no orçamento, dando origem a novas necessidades ou até mesmo, causando cancelamento de iniciativas assumidas anteriormente.

Relações com as partes interessadas externas e suas percepções e valores

A maior parte interessada é a sociedade. Todas as informações recolhidas por meio dos exames, das avaliações e dos censos realizados na educação básica e superior são disponibilizadas para a sociedade em forma de micro dados, além de publicações em suas duas revistas: Em aberto e Revista Brasileira de Estudos Pedagógicos (RBEP) e por meio de boletins eletrônicos e consultas que podem ser realizadas diretamente no portal do INEP. Os pesquisadores, as universidades, os alunos, as escolas e demais órgãos recebem estes

dados, que são disseminados pelo INEP como forma de melhoria na educação, repasse de verbas e insumo para propostas de melhorias das políticas públicas importantes para o cidadão. Atualmente a percepção da sociedade em relação do INEP tem aumentado por conta do ENEM, que desde 2009 passou a ser instrumento de entrada nas universidades, extinguindo, assim aos poucos, os vestibulares. O valor percebido pela sociedade diz respeito ao aumento também da conscientização da importância da educação dos brasileiros.

O INEP é reconhecido no Governo Federal como provedor de soluções de estudos e pesquisas, sistemas de estatísticas, avaliações de educação básica e superior, exames de acesso à educação superior e disseminação de resultados de avaliações. Nesse contexto, a DTDIE desenvolve ações de tecnologia da informação e comunicação para oferecer ao INEP soluções de apoio à execução de programas oriundos de políticas públicas de educação [46].

4.2 Contexto Interno

4.2.1 Diretoria de Tecnologia e Disseminação de Informações Educacionais (DTDIE)

Conforme organograma exibido na Figura 4.1, o INEP possui seis diretorias com responsabilidades bem distribuídas, todas culminando para um atingimento comum das finalidades do instituto:

1. DGP - Diretoria de Gestão e Planejamento;
2. DTDIE - Diretoria de Tecnologia e Disseminação de Informações Educacionais;
3. DEED - Diretoria de Estatísticas Educacionais;
4. DAEB - Diretoria de Avaliação da Educação Básica;
5. DIREDE - Diretoria de Estudos Educacionais e
6. DAES - Diretoria de Avaliação da Educação Superior.

No contexto interno do INEP, será discutida e abordada a importância da DTDIE, mais especificamente de uma de suas coordenações, a CGSI - Coordenação Geral de Sistemas de Informação que será particularmente o objeto deste estudo de caso. Conforme a Figura 4.1, a CGSI é subordinada a DTDIE e possui um papel fundamental sendo a responsável pelos projetos de desenvolvimento de software para atender às demandas das demais diretorias do INEP. Estes sistemas, em sua maioria, são sistemas críticos de avaliações, exames e censos de coleta de dados da educação básica e superior de todo o

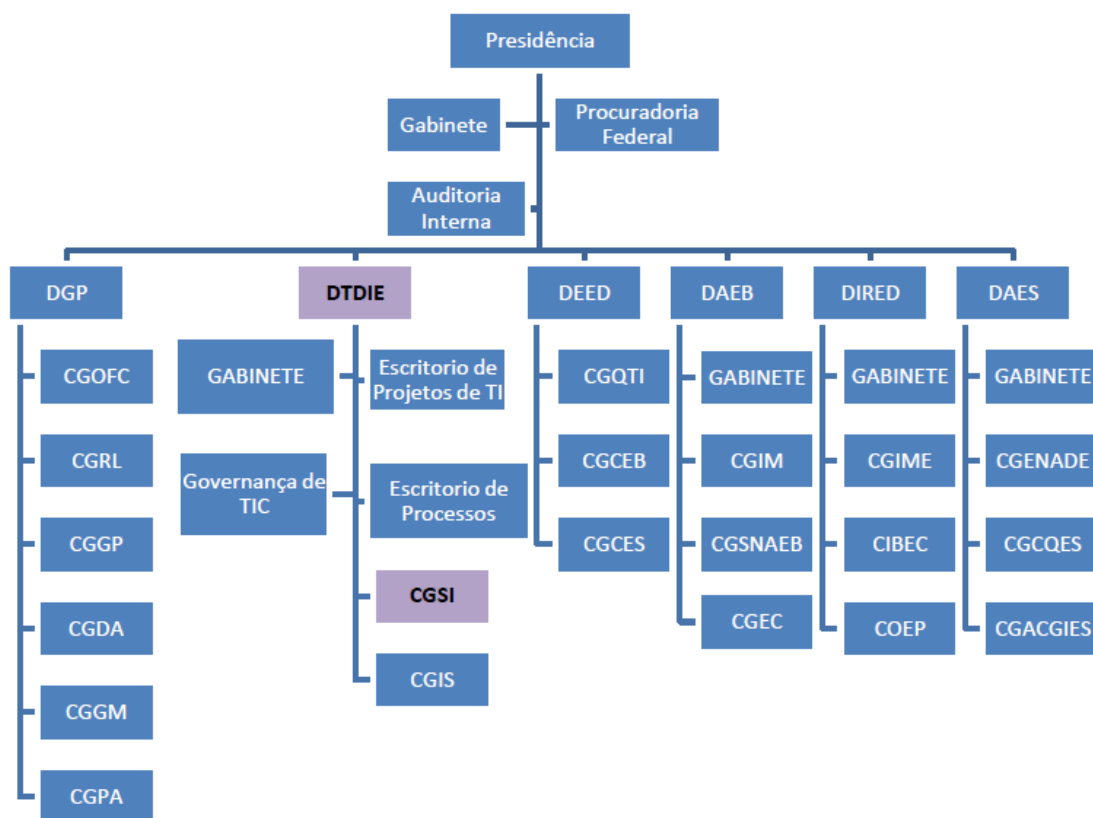


Figura 4.1: Organograma INEP. Fonte: Adaptado do portal INEP [46].

Brasil, demandando atendimento à prazos legais, alta disponibilidade e performance dos serviços. Possuem a característica de terem datas de abertura e fechamento do sistema que segue um planejamento anterior, comunicado à sociedade, e por isso precisam estar disponíveis nestes períodos, correndo o risco de causar danos à imagem do órgão e até mesmo do MEC caso não cumpram o planejamento já divulgado [46]. Muitos sistemas não possuem flexibilidade para médias ou grandes postergações de prazo, pois por meio do resultado dos dados capturados por estes sistemas críticos é que são planejados e realizados os repasses de verbas públicas, para transporte e merenda escolar, assim como melhorias na escola. Ou seja, se estes sistemas críticos atrasarem, poderá haver um impacto em todo o planejamento, que excede o do próprio projeto. Nota-se que tais sistemas desenvolvidos e mantidos pela CGSI são de fundamental importância para que o órgão atinja seus objetivos institucionais [46].

Para que a TI possa suportar e apoiar o alcance destes objetivos institucionais, a DTDIE formulou o Plano Diretor de Tecnologia da Informação (PDTI), alinhando as ações da TI com os objetivos estratégicos do órgão. Ainda que não exista um Planejamento Estratégico Institucional (PEI), é possível realizar este alinhamento com base no decreto

de criação do órgão, no PNE e no Plano Plurianual (PPA), dentre outras informações que foram levantadas com a colaboração dos servidores e contratados da DTDIE.

4.2.2 Plano Diretor de Tecnologia da Informação - PDTI

O PDTI do INEP (2016-2019) expressa as expectativas e o trabalho conjunto entre os órgãos de assistência direta e imediata à presidência, órgãos seccionais e áreas correlatas da DTDIE. A Instrução Normativa SLTI 04/2014, em seu art. 2, inc. XXVII, define o PDTI como um “instrumento de diagnóstico, planejamento e gestão dos recursos e processos de Tecnologia da Informação que visa atender às necessidades tecnológicas e de informação de um órgão ou entidade para um determinado período” [30].

O PDTI é o principal instrumento de planejamento, comunicação e gestão de recursos e processos de Tecnologia da Informação e Comunicação. Busca atender aos objetivos estratégicos institucionais do INEP, que estão alicerçados em ações do MEC, do FNDE e da Capes, todos alinhados ao PNE. O PDTI perfila o planejamento da DTDIE às necessidades estratégicas institucionais por meio da declaração de sua identidade organizacional, objetivos, metas e indicadores em concordância com práticas reconhecidas pela administração pública. Esse planejamento abarca toda a DTDIE – coordenações e áreas subordinadas – e demais áreas negociais do INEP por meio da identificação e do atendimento das necessidades e da melhoria da prestação de serviços de TI [30]. O PDTI traz ainda declarações estratégicas como missão, visão e valores que traduzem a identidade da organização, sua missão e os objetivos para o alcance da visão de futuro deixando claro os valores que são desejáveis para seus colaboradores.

Objetivos Estratégicos

Conforme o Quadro 4.1, os objetivos estratégicos são resultados que a DTDIE pretende atingir para o alcance da Visão.

Pode-se observar que a proposta do trabalho está em consonância com o objetivo 07, "Aprimorar a Gestão de Projetos", visto que a gestão de riscos é uma disciplina da gestão de projetos.

Análise SWOT

A análise ambiental envolveu a participação dos núcleos estratégico, tático e operacional da DTDIE, sendo realizada por meio da análise SWOT, acrônimo de Forças (Strengths), Fraquezas (Weaknesses), Oportunidades (Opportunities) e Ameaças (*Threats*).

Quadro 4.1: Objetivos Estratégicos de TI. Fonte: [30]

ID	Perspectivas	Objetivo
Obj1	Governança	Aprimorar a Governança de TI
Obj2		Modernizar e Integrar as práticas de gestão de pessoas na TI
Obj3		Aprimorar o uso dos recursos financeiros
Obj4		Aprimorar governança de dados, informação e conhecimento
Obj5		Aprimorar a gestão de processos
Obj6		Intensificar o uso da TI na gestão do INEP
Obj7	Projetos	Aprimorar a Gestão de Projetos
Obj8	Operações	Disponibilizar infraestrutura tecnológica
Obj9		Aprimorar a gestão de serviços
Obj10		Aprimorar a gestão de segurança da informação e comunicações

- Forças e fraquezas referem-se à avaliação do ambiente interno da DTDIE – coordenações e divisões subordinadas à diretoria.
- Oportunidade e ameaças referem-se ao ambiente externo à DTDIE – presidência, outras diretorias e situações externas ao INEP.

Para um maior entendimento do contexto interno, são apresentadas as forças e fraquezas levantadas na DTDIE, que constam no PDTI [30]. O conhecimento destas informações já levantadas auxilia na percepção do caso a ser estudado.

As Forças:

- Boa comunicação entre as equipes de arquitetura, PHP e testes;
- Capacidade da Governança de TI envolver-se nas ações estratégicas institucionais;
- Capacidade de estabelecer parcerias e bons relacionamentos em âmbito interno e externo;
- Capacidade de planejamento voltado à execução de orçamento para as contratações e capacitações;
- Capacidade de união (sinergia) entre as equipes de desenvolvimento para alcançar resultados;
- Capacidade dos colaboradores em realizar trabalhos cooperativos e multissetoriais;
- Dedicção, comprometimento e disponibilidade dos colaboradores com os trabalhos do INEP;
- Diretoria com foco em planejamento e entrega de resultados;

- Eficiência e eficácia na resolução de situações adversas, críticas e pontuais da DT-DIE;
- Empenho da diretoria em atender, de forma justa, às solicitações das demais diretorias, de forma a prover o bem-estar de todos;
- Equipe (servidores e colaboradores) com grande *know-how* (conhecimento prático);
- Existência de servidores do quadro efetivo do INEP ocupando cargos de gestão;
- Infraestrutura (*desktops* e redes) adequada ao ambiente e necessidades de trabalho;
- Manual de contratações de TI em consonância com a Instrução Normativa N° 04/SLTI/ MP 2014;
- Parceria e bom relacionamento com as contratadas para a solução de conflitos e melhorias;
- Participação ativa da diretoria nas ações de realização do PDTI;
- Proximidade da diretoria no acompanhamento dos trabalhos e
- Salvaguarda e apoio da gestão de TI nas decisões internas para a solução de problemas.

As fraquezas:

- Ações individuais de configuração e mudanças realizadas sem planejamento, comunicação e avaliação de impacto sobre outros sistemas;
- Alta dependência de terceiros;
- Alta rotatividade de pessoal (servidores, temporários e colaboradores);
- Ausência de padrões e falta de clareza nos padrões existentes;
- Criação de demandas internas sem planejamento, avaliação de impacto ou justificativa;
- Cultura interna de achar e apontar culpados em vez de buscar soluções;
- Deficiências nas comunicações entre as áreas;
- Desconhecimento da capacidade de atendimento interno pelos gestores, favorecendo atrasos e suprimindo etapas necessárias do processo de desenvolvimento;
- Descontinuidade do uso de tecnologias sem avaliação prévia de impacto;
- Dificuldades da DTDIE em impor autonomia perante outras diretorias;

- Distribuição desproporcional do trabalho entre servidores e colaboradores;
- Entrega de produtos com baixa qualidade para a sociedade;
- Falhas na comunicação das ações e estratégias da Diretoria;
- Falhas no planejamento e controle das ações de projetos;
- Ineficácia no dimensionamento de equipes e recursos, favorecendo a entrega de produtos com qualidade comprometida por motivo de promessas de prazos inatingíveis;
- Falta de padronização de conteúdo, critérios de aceitação e fluxo de trabalho de OSs, gerando dificuldades para as fábricas de software, uma vez que cada PO possui uma forma de validação;
- Falta de priorização de demandas, causando conflitos entre as áreas;
- Falta de objetividade e de pontualidade nas reuniões internas;
- Falta de informações históricas, premissas e soluções técnicas adotadas em sistemas antigos;
- Fragilidades na gestão de contratos e serviços terceirizados;
- Incapacidade das contratadas em estimar prazos realistas;
- Indefinição de ações estratégicas para as atividades internas;
- Indefinição de papéis e responsabilidades de servidores e colaboradores;
- Indefinição ou inexistência de processos de trabalho interno;
- Inexistência de cultura de compartilhamento de problemas e soluções;
- Pouca sinergia entre as equipes de infraestrutura, banco de dados e de desenvolvimento de sistemas;
- Pouco ou nenhum prazo para o planejamento e execução de testes;
- Processos executados de forma heurística (por descoberta), com base na experiência de pessoas;
- Quantidade insuficiente de funções e gratificações na DTDIE;
- Quantidade insuficiente de pessoas para atender ao grande volume de demandas e
- Sobrecarga de trabalho e excesso de atividades e horas extras.

As fraquezas são associadas ao contexto interno da organização e, portanto, possuem um foco especial neste trabalho. Ao usar a SWOT para se identificar riscos, faz com que

todas as fraquezas possam ser encaradas como um risco ou como um fator agravante para que as ameaças se concretizem [66] [86]. Como as ameaças estão fora do contexto da organização e não podem ser totalmente controladas, serão usadas as fraquezas listadas acima, a fim de possibilitar a extração de riscos, que mais tarde irão compor também a lista de riscos comuns referente a projetos de software. A elaboração de lista/*checklist* de riscos comuns é uma boa prática citada por vários autores [79] [50] [28] [63].

4.2.3 Coordenação Geral de Sistemas de Informações (CGSI)

A CGSI possui contrato de fábrica de software para o levantamento e desenvolvimento dos projetos de software. No contrato atualmente vigente, o fluxo do processo de desenvolvimento ou manutenção de sistemas é detalhado na MGDS. A seguir, é explicado de forma detalhada o fluxo de trabalho desta diretoria. Segundo o contrato, as OSs podem ser emitidas conforme a necessidade da demanda e se enquadram nos tipos abaixo:

- Iniciação - Refere-se ao levantamento de necessidade e requisito. Demanda a participação em reuniões para entendimento negocial e descrição de casos de uso ou estórias de usuário.
- Desenvolvimento - Tem por objetivo o desenvolvimento de nova funcionalidade e está atrelada a uma OS de iniciação.
- Manutenção - O objetivo é a realização de manutenção de sistemas já em produção. Manutenção adaptativa ou evolutiva de funcionalidades já existentes.
- Garantia - Tem por objetivo a correção de funcionalidade já entregue no período de até 1 ano.

Segundo a MGDS, a definição da visão do novo sistema a ser desenvolvido ou de sistema legado a sofrer manutenção dá-se na etapa de iniciação, fase que pode ser executada com ou sem auxílio da contratada [94]. A iniciação constitui-se no planejamento propriamente dito. Nessa fase, além da elaboração da visão do produto e do estabelecimento da definição de ‘pronto’, também são elicitados os requisitos que constarão do *backlog* do produto. Caso haja necessidade de participação da contratada na iniciação, uma Ordem de serviço específica é aberta para a realização de atividades de levantamento preliminar de escopo junto ao cliente ou de detalhamento da demanda. Nesse tipo de ocorrência, a contratada tem remuneração correspondente a 10% da quantidade de pontos de função estimada para a respectiva OS [94].

A fase de iniciação é encerrada quando a demanda planejada, seja desenvolvimento de novo sistema, seja manutenção de sistema legado, é entregue ao INEP e passa para o

status de conforme no sistema Controle de Contratos e Projetos (CCP) [94]. Na sequência, inicia-se a fase de execução, que é desenrolada segundo os preceitos do *framework Scrum*, com reunião de planejamento da *sprint*, onde se obtém o esforço das funcionalidades, sendo uma das saídas deste processo o prazo que é requisito obrigatório para a emissão da OS à fábrica. Durante a execução, são realizadas reuniões diárias e reunião de revisão, na qual a contratada entrega os produtos da *sprint*, além da reunião de retrospectiva, onde são levantadas as lições aprendidas. Para a execução da *sprint*, a contratada é remunerada em 100% do total obtido na medição final dos pontos de função da iteração entregue e aceita. Ao final da fase de execução também pode se dar a implantação, no ambiente de produção, do produto gerado na *sprint*, mediante a conformidade em testes e a homologação da funcionalidade pelo cliente.

A última fase do processo de desenvolvimento ou manutenção de sistemas, de acordo com a MGDS, consiste no encerramento, que tem como objetivo encerrar as ordens de serviço de iniciação e da *sprint*. No encerramento, ocorre a aceitação definitiva das Ordens de Serviço, habilitando-as para faturamento pela contratada.

Antes da execução da fase de encerramento, a MGDS prevê a homologação dos artefatos produzidos pela contratada em atendimento às Ordens de serviço de iniciação e da *sprint* em dois momentos.

No primeiro, a avaliação da qualidade dos artefatos é realizada pela área de TI quanto à conformidade com os padrões estabelecidos pelo INEP. No segundo, a avaliação é feita sob o ponto de vista do negócio pelo cliente demandante [94].

Para as ordens de serviço de iniciação são realizadas avaliações levando em conta a aderência ao negócio. Ou seja, se a documentação (especificações de casos de uso ou histórias de usuário) refletem o que foi solicitado pelo cliente, se está aderente ao negócio do cliente. Caso sejam entregues, além desta documentação, também *scripts* de banco de dados, os mesmos devem passar por avaliação de um administrador de dados (AD) do INEP para que seja garantido o atendimento à boas práticas de mercado, como criação das tabelas e atributos padronizados, sem redundância e com dicionário de dados que facilite uma futura manutenção.

Já para as ordens de serviço de execução, a avaliação consiste em validação de código-fonte, de *scripts* de banco de dados e de testes funcionais, todos solicitados via demanda às áreas responsáveis na DTDIE. Para tais avaliações, o INEP possui células de equipes que não fazem parte do contrato de fábrica. Possui equipe de arquitetura, equipe de administração de dados, de testes e os Líderes de projetos, responsáveis pelo planejamento e controle das OSs (emissão, solicitação de validação e homologação) e principalmente, são responsáveis por orquestrar todas as atividades de diferentes equipes e subáreas da DTDIE por meio de demandas e reuniões, alinhando também as expectativas dos clientes

sobre o andamento do projeto.

Caso estejam conforme em todos os quesitos avaliados, a OS prossegue para a homologação do cliente. Caso seja encontrada inconformidade em qualquer um dos quesitos, a OS é devolvida via sistema à fábrica para que seja corrigida no prazo estipulado no contrato de até 48 horas. A sanção é aplicada dependendo do tipo da inconformidade. As inconformidades se classificam em:

1. Conforme com ressalvas - é atribuído às inconformidades que não são impeditivas e demandam pequenas alterações. Não possui caráter punitivo e não acompanha sanção.
2. Não conforme - é atribuído às inconformidades encontradas que são impeditivas, ou ainda em casos de reincidência de inconformidade do tipo 'conforme com ressalva' que não tenha sido corrigida, mesmo após ter sido apontada em verificação anterior. Há sanção prevista em contrato.

Após a OS seguir este fluxo de emissão, entrega e validação, caso esteja em total conformidade, será implantada em produção mediante homologação/autorização do cliente que é o fiscal requisitante do contrato. Caso o cliente não homologue no prazo estipulado, a OS é homologada tacitamente pelo Líder de projeto para que a contratada receba pelo esforço já realizado, porém o código fonte não será implantado no ambiente de produção até que o cliente autorize. Qualquer erro descoberto após implantação em produção será corrigido por meio de OS de garantia.

4.3 Análise da Situação dos projetos de software da CGSI

Nesta seção são analisados o catálogo de sistemas e as Ordens de serviços a fim de identificar a situação atual.

4.3.1 Análise do Catálogo de Sistemas

Segundo o documento Catálogo de Sistemas, mantido pela Governança da DTDIE, atualizado em setembro de 2016, existem 87 sistemas e mais de 200 módulos ou subsistemas, sob a responsabilidade da CGSI, sejam relacionados ao desenvolvimento de *softwares* novos, ou sustentação nos sistemas existentes [31].

O gráfico exibido na Figura 4.2 mostra de forma consolidada a situação que se encontram os módulos dos sistemas da CGSI.

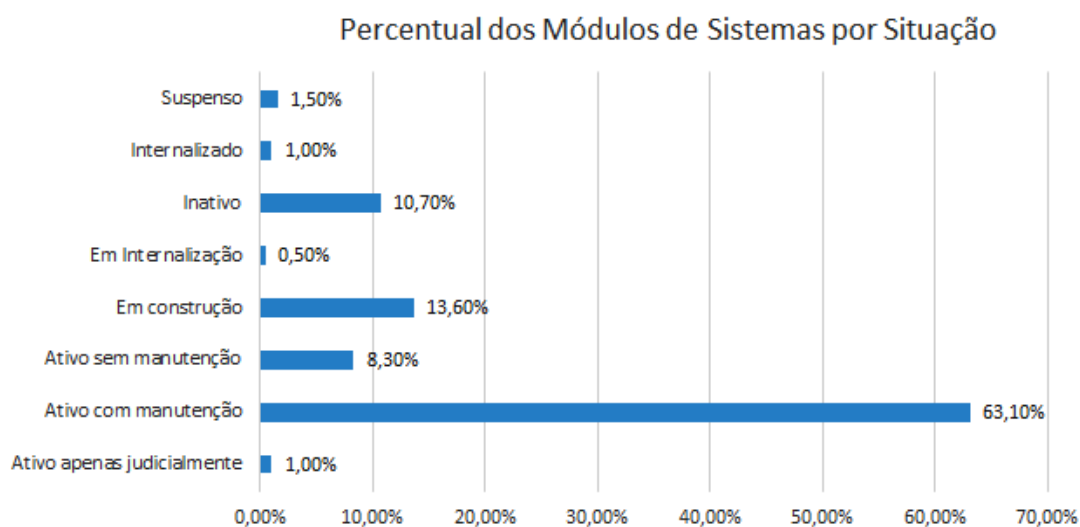


Figura 4.2: Percentual dos Módulos de Sistemas por Situação. Fonte: Elaboração própria [31].

Constatou-se que apenas 13,6% de todos os módulos dos sistemas correspondem a novos projetos, ao passo que 63% são módulos de sistemas que sofrem manutenções evolutivas ou adaptativas. Constatou-se ainda que cerca de 19% são módulos que estão ou inativos ou sem manutenção, ou seja, estão ativos somente para consulta. Os demais 4% dos módulos dos sistemas ou são internalizações, ou estão disponíveis apenas judicialmente ou se encontram suspensos. Ou seja, a força de trabalho se concentra em 63% em manutenções de sistemas, sendo que pouco mais de 13% se concentra no desenvolvimento de novos sistemas [31].

Com relação a categorias de negócio que estes sistemas se enquadram, no catálogo de sistemas, todos os sistemas se enquadram em uma das categorias a seguir:

- Exames e Avaliações - Trata-se de todos os sistemas que possuem o objetivo de inscrição de usuários para a realização de exames ou avaliações, cumprindo etapas como disponibilização de local de provas e resultado final.
- Censos - Trata-se de ambos os censos da educação básica e superior do país.
- Administrativos - Sistemas de uso interno do INEP, das mais variadas diretorias.
- Comunicação - Sistemas de comunicação, tais como: *Moodle*, Pergamun, Sistema de publicações e Prêmio Inovação.
- Não definido - Sistemas em sua maioria internalizados, que foram desenvolvidos fora do INEP, o qual não se pode enquadrar em uma das categorias acima.

Conforme a Figura 4.3, a maioria dos sistemas e seus respectivos módulos pertencem às categorias das áreas de negócio: ‘Exames e Avaliações’ e ‘Censos’, representando juntos mais de 70% de todos os sistemas e subsistemas do INEP, sendo que 17% dos sistemas estão relacionados à administração interna, estando ligados a setores como: Financeiro e Recursos Humanos ou controles internos, como gestão de contratos [31].

Percentual de Módulos de Sistemas por Área de Negócio

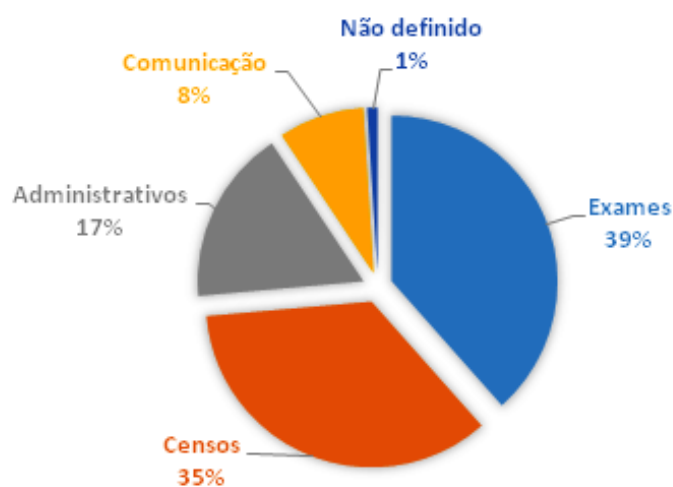


Figura 4.3: Percentual de Módulos por Área de Negócio. Fonte: Elaboração própria [31].

Sobre a tecnologia utilizada nos sistemas, o gráfico exibido na Figura 4.4 mostra o percentual de módulos de sistemas em cada tecnologia.

A maior parte dos sistemas são desenvolvidos em Java. Este dado pode ser explicado pelo fato de que dentre outros fatores, todos os sistemas cujo usuário seja externo ao INEP, devem ser construídos em Java, segundo regra interna da área de arquitetura [31].

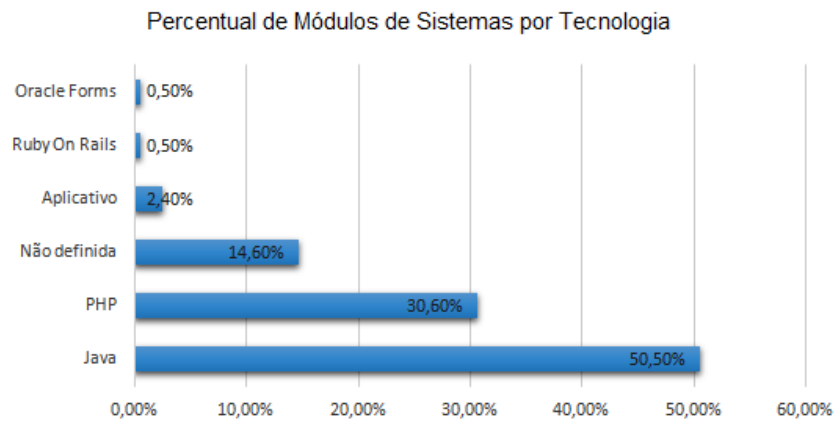


Figura 4.4: Percentual de Módulos por Tecnologia. Fonte: Elaboração própria [31].

Verificando a quantidade de módulos por diretoria do INEP, pode-se constatar que, segundo a Figura 4.5, a DEED demanda aproximadamente 30% de todos os sistemas, incluindo sistemas da área de negócio: ‘Censos’, sendo que a DGP e DAEB demandam percentuais semelhantes entre si [31].

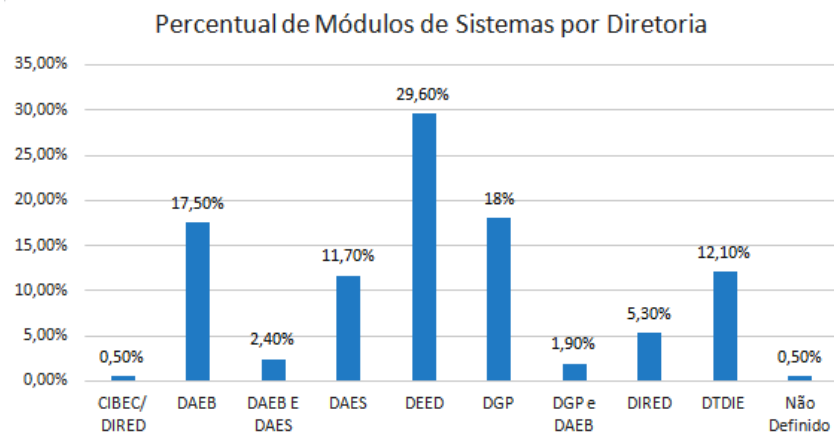


Figura 4.5: Percentual de Módulos de Sistemas por Diretoria. Fonte: Elaboração própria [31].

De forma geral a maioria dos sistemas possuem apenas um módulo, porém conforme a Figura 4.6, alguns sistemas possuem mais de 5 módulos, a citar: Censo Superior, Censo Básico, BNI, Robô, ENCCEJA, ENEM e EMEC, tornando-os complexos de gerenciar e demandando grande esforço dos gerentes.

Quantidade de Módulos por Sistema

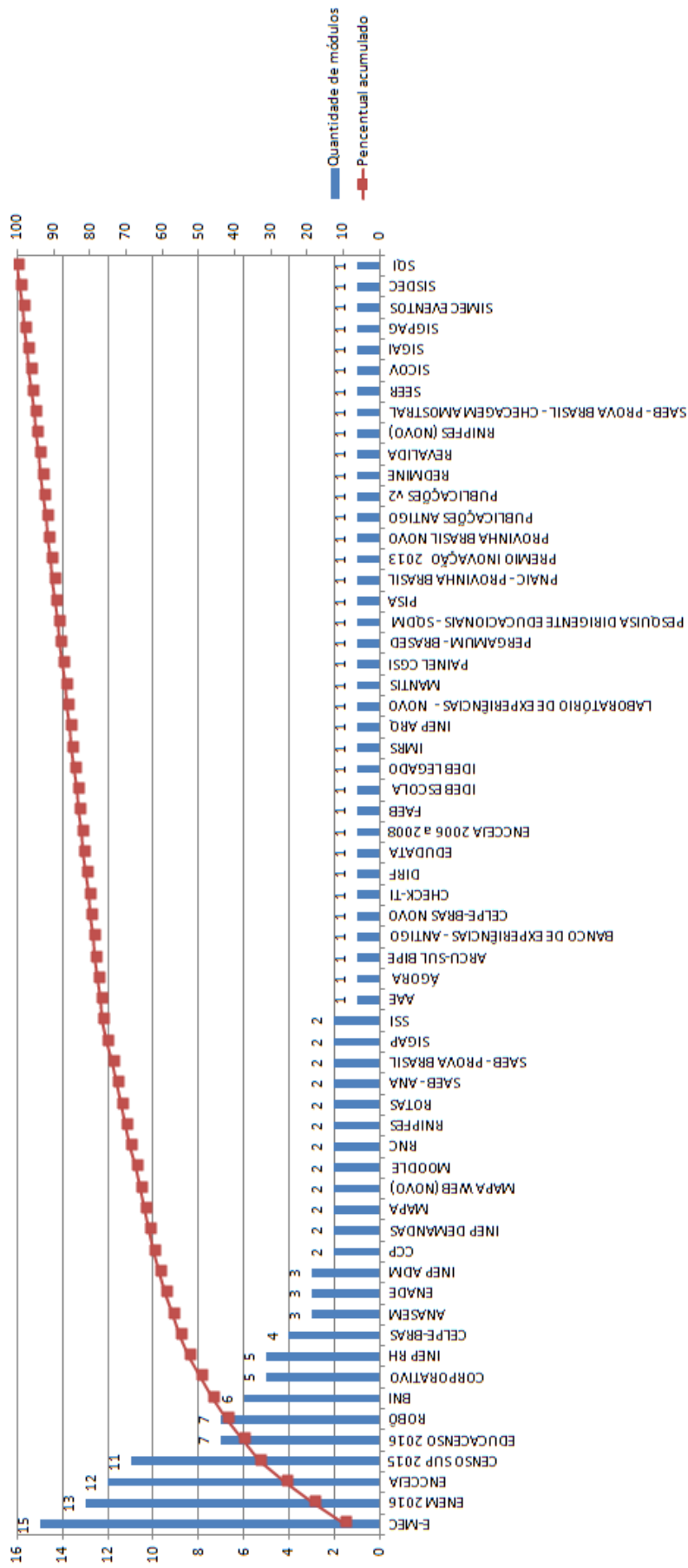


Figura 4.6: Quantidade de Módulos por Sistema. Fonte: Elaboração própria [31].

Nesta subseção foram analisados os percentuais de módulos de sistemas por situação, por área de negócio, por tecnologia, por diretoria e por projeto. Isso possibilita uma visão geral relacionada aos sistemas que são demandados por meio dos projetos de software. Conforme exibido nesta seção, aproximadamente 80% de todos os sistemas e subsistemas estão ativos e demandam esforço (seja por meio de ‘projetos novos’, ‘manutenções’, ‘internalizações’, ou na situação de ‘ativo apenas judicialmente’). Mesmo os sistemas e subsistemas que necessitam estar ativos porém sem manutenção (8.3%), necessitam de atenção, quanto a sua disponibilidade, exigindo em certo grau um esforço do gerente de projetos. Então pode-se verificar que cerca de 88% dos módulos dos sistemas demandam atenção.

4.3.2 Análise das Ordens de Serviços dos Projetos de Software

Para analisar a situação dos projetos, é necessário analisar as OSs dos projetos. Para isso, foram realizadas consultas à base de dados do sistema CCP. Esta análise conta com uma amostra de 1.085 OSs do atual contrato, ou seja todas as OSs que foram emitidas a uma mesma terceirizada, e que estão aceitas no sistema CCP no período de 01/08/2012 a 01/09/2015. Este contrato iniciou-se em 2012 e finda em 2017. Assim, a amostra de 3 anos de OSs corresponde a 60% de todas as OSs que foram emitidas neste contrato. É relevante mencionar que não necessitou de integração com outras bases, além da própria base de dados do CCP. Foram coletadas as 12 variáveis listadas abaixo:

1. Número da Ordem de Serviço - código sequencial - chave da tabela;
2. Emissão - Data da emissão da Ordem de serviço à contratada;
3. Diretoria - Diretoria demandante do serviço;
4. Projeto - Nome do projeto ao qual a OS foi vinculada;
5. Tipo de demanda - podem ser de 4 tipos: Iniciação, Manutenção, Desenvolvimento e Garantia;
6. Entrega Prevista - Data da entrega prevista e informada pela fábrica;
7. Primeira Entrega - Data da primeira entrega realizada;
8. Última Entrega - Data da última entrega. Em caso de entregas no prazo, poderá ser igual a data da primeira entrega e da entrega prevista. Porém, em caso de atrasos é divergente da data da entrega prevista.
9. Homologação realizada - Data em que a OS foi homologada no sistema;

10. Situação - Atributo derivado, para receber os valores: no prazo, atrasado e adiantado;
11. PF Estimado - Quantidade estimada de Pontos de Função calculados para a emissão da OS e
12. PF Realizado - Quantidade de Pontos de Função calculados com base no que foi realmente realizado pela fábrica, calculados pelo núcleo de métricas da CGSI, novamente após a entrega da OS. Normalmente há divergência entre ambas as variáveis, devido ao fato de se utilizar a contagem estimada para a emissão da OS e a contagem detalhada somente após a finalização do trabalho. Isso corrigindo para mais ou para menos as estimativas iniciais.

Tratamento dos dados

Nos dados obtidos, as datas foram transformadas em dias, por meio do cálculo dos intervalos de dias entre uma data e outra, para melhorar a forma de entendimento dos prazos. A data de homologação não satisfaz diretamente os critérios de sucesso do projeto, pois é somente a data de aceite do cliente, não agregando em termos da medição do sucesso do projeto no que diz respeito ao prazo estimado para entrega da OS.

Quanto à qualidade dos dados, sabe-se que as variáveis ‘PF Estimado’ e ‘PF Realizado’ possuem muitos dados faltantes (*missing values*), pois quando uma ordem de serviço é de manutenção com prioridade crítica e pontual, a contagem não é realizada, ficando vazia no banco de dados. Desta forma, ambas as variáveis não são analisadas neste trabalho.

A variável Número da Ordem de Serviço, por ser um atributo chave da tabela, também não será objeto de análise, pois identifica de forma única determinado registro, não sendo o objetivo da análise. Foi verificado ainda que alguns registros tinham a data de entrega prevista anterior à data de emissão, o que denotou problemas na qualidade dos registros e por isso foram expurgados. Após o tratamento de dados das 1085 OSs iniciais e retirada dos atributos (campos) que possuíam dados faltantes (*missing values*) ou eram *outliers*, identificados pelo método de intervalos interquartílicos, finalmente, a análise foi realizada com base em 964 OSs cujos dados estavam tratados e íntegros.

Analisando a Situação das OSs

Para a realização da análise, optou-se por utilizar o algoritmo da árvore de decisão disponibilizado pelo *IBM SPSS Statistics*, a fim de classificar as OSs pelo rótulo da variável Situação, mantendo o percentual referente a cada tipo de OS a cada nó. Cabe ressaltar que o algoritmo não foi utilizado para a tomada de decisão. O uso do software permitiu uma maior correção nos cálculos dos quase mil registros analisados, fator

preponderante, quando comparado a possíveis erros que poderiam ocorrer caso fossem calculados manualmente.

Neste trabalho, a análise foi embasada no número de ordens de serviços e segmentada em 3 classificações distintas (OSs: adiantadas, no prazo ou atrasadas).

Em uma primeira análise é possível verificar na Figura 4.7, que 42% das OSs são do tipo manutenção, 30% iniciação, 20.3% desenvolvimento e apenas 6.8% são de garantia. Tal análise das OSs mostrou que a maioria delas são do tipo ‘manutenção’. Assim, as manutenções de *software* ocorrem em maior número quando comparadas ao desenvolvimento de novos projetos ou funcionalidades, confirmando a informação que constava no catálogo de sistemas, onde a maior parte do esforço se concentrava na manutenção dos sistemas.

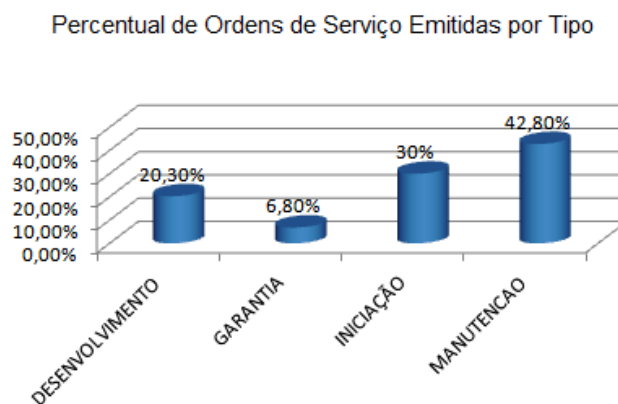


Figura 4.7: Percentual do tipo de Ordens de Serviço. Fonte: Elaboração própria.

Conforme análise realizada e exibida na Figura 4.8 verificou que a taxa de atrasos na entrega das OSs ‘aceitas’, chega a 47.9%, distribuído entre os projetos cadastrados no CCP. As OSs entregues adiantadas correspondem a apenas 7,2% e as entregues dentro do prazo correspondem a 44.9%. As OSs entregues adiantadas e dentro do prazo somam 52.1% do total de todas as OSs aceitas durante o intervalo de três anos analisados.

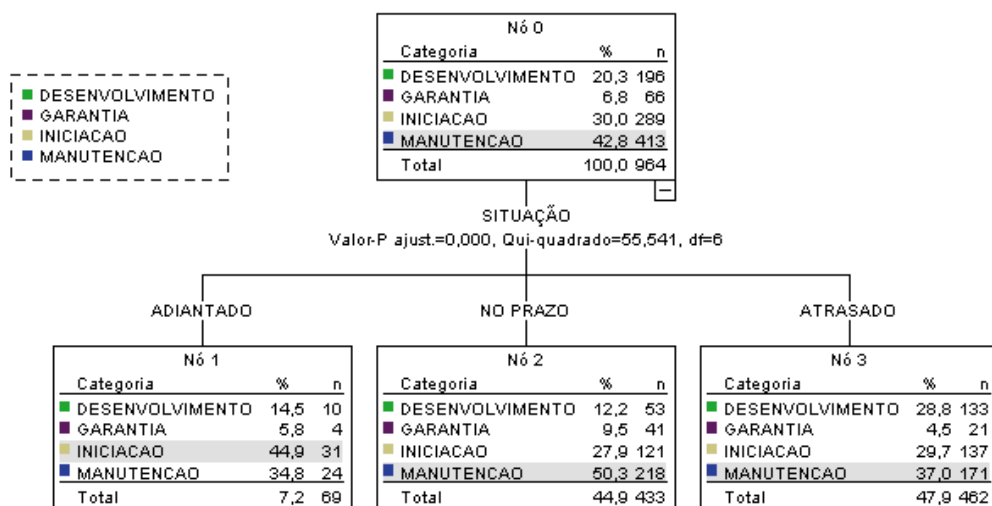


Figura 4.8: Situação das OSs entregues. Fonte: Elaboração própria.

Com base na Figura 4.8 e em seus percentuais quantitativos é possível chegar à seguinte conclusão: as ordens de serviço que mais atrasam são as relacionadas ao desenvolvimento de novos projetos, ou seja, são as OSs do tipo desenvolvimento que possuem o dobro da probabilidade de serem entregues em atraso (28,8%, correspondente ao nó 03) quando comparado com o percentual de entregas no prazo (12,2%, correspondente ao nó 02). Já as OSs de manutenção em sua maioria (218 OSs de 413 no total, exibidas nos nós 02 e 0 -nó principal) foram entregues no prazo. Esta análise mostra como novos projetos de softwares possuem mais que dobro de chance de sofrerem atrasos na entrega final (comparando os nós 2 e 3 na categoria desenvolvimento).

Neste contexto, há uma necessidade real de se elucidar o motivo dos atrasos em projetos de *software* somarem quase 48% de todas as entregas realizadas. Porém, tais atrasos podem ocorrer devido a uma série de riscos aos quais os projetos são expostos no decorrer de suas diversas fases do próprio desenvolvimento de *software*.

Diferenças entre o Prazo Estimado x Prazo Executado usando Cluster

A Tabela 4.1 mostra a diferença entre o prazo de entrega estimado das OSs e o prazo em que foram realmente entregues (prazo executado).

Trata-se de uma média dos prazos estimados e executados em dias, sendo que as OSs foram segregadas por meio de *Cluster*, onde a variável categórica: ‘tipo de OS’, foi utilizada para definir o número máximo de *clusters*, assim foi possível, ter um prazo médio correto para cada agrupamento. Nota-se que para a iniciação apesar de preverem em média uma semana para a entrega, o prazo em média realizado são de 15 dias, para

Tabela 4.1: Classificação de prazo estimado x executado por tipo de OS. Fonte: Elaboração própria

Cluster	1	2	3	4
Tamanho	43.5%	29.1%	20.4%	7.0%
Entradas	405	271	190	66
Tipo de Ordem de Serviço	Manutenção	Iniciação	Desenvolvimento	Garantia
Prazo Estimado	5.57	7.63	10.3	2.68
Prazo Executado	12.58	15.71	24.40	6.21

desenvolvimento 24 dias, para manutenção 12 dias e para garantia 6 dias. Esta média de prazo executado é calculado considerando desde a data de emissão da OS até a data da última entrega desta mesma OS.

Pode-se assumir que algo está causando o atraso das OSs, assim, a causa raiz destes atrasos necessita ser investigada.

4.3.3 Análise de Causa Raiz por meio da Análise de Árvore de Falhas (FTA)

Para tanto, foi realizada análise, por meio da ferramenta chamada Análise de Árvore de Falhas (FTA), que está prevista na norma ISO/IEC 31010:2012, sendo utilizada para auxiliar na descoberta das principais causas-raízes de determinado evento que esteja sendo estudado, chamado de evento topo da árvore. Neste caso, o evento topo da árvore no qual se deseja descobrir as causas raízes é a taxa de atrasos dos projetos da coordenação estudada. Fatores causais são identificados por dedução e organizados de maneira lógica e representados em um diagrama de árvore que descreve fatores causais e sua relação lógica com o evento de topo [3].

A árvore de falhas pode ser utilizada de forma qualitativa para identificar potenciais causas e os caminhos para uma falha ou quantitativamente para calcular as probabilidades do evento topo, dado o conhecimento das probabilidades de eventos causais [3]. Auxilia na identificação das falhas, aqui tratadas como ‘inconformidades’. Assim também é verificada a importância (a probabilidade) dos diferentes caminhos que mais contribuem para que o evento topo ocorra. A aplicação da abordagem *top-down* implícita na técnica foca a atenção nos efeitos da falha que estão diretamente relacionadas com o evento topo [3]. A análise de causa e efeito, como o diagrama *Ishikawa* ou espinha de peixe, é similar a uma árvore de falha na aparência. Entretanto, não pode ser quantificada para produzir a probabilidade do evento principal, uma vez que as causas são possíveis fatores contributivos com uma probabilidade de ocorrência conhecida. Por esta razão, foi utilizada a análise de árvore de falhas.

Para a obtenção dos percentuais, foram realizadas consultas diretamente no banco de dados do sistema CCP para se obter o resultado das avaliações de cada OS e assim permitir o cálculo do percentual do resultado obtido para cada tipo de avaliação. A título de explicação, quando cada OS é entregue pela fábrica no sistema CCP, o Líder de projeto necessita abrir avaliações no sistema INEP demandas, conforme o tipo da OS. A Figura 4.9 mostra quais tipos de avaliações as OSs de iniciação e execução estão sujeitas.

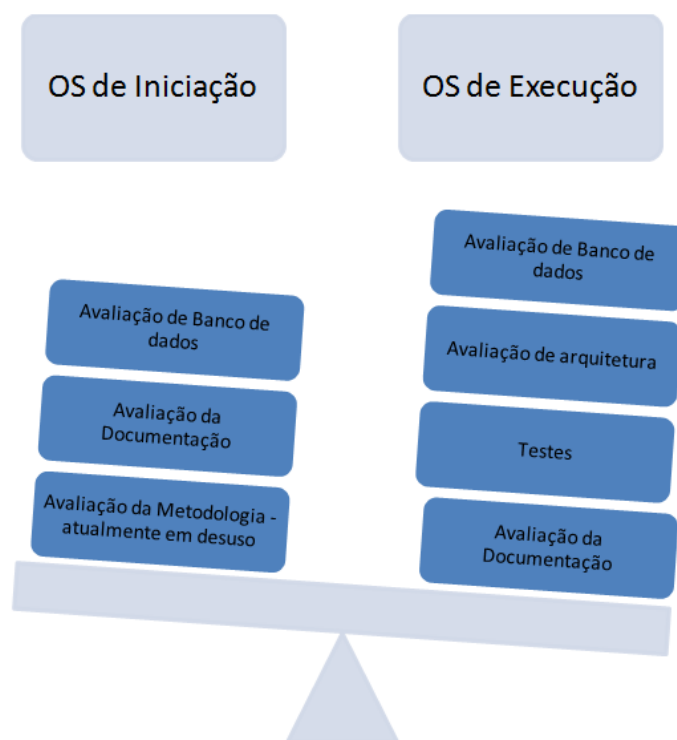


Figura 4.9: Tipos de avaliações a serem submetidas cada tipo de OS. Fonte: Elaboração própria.

Uma OS de iniciação poderá gerar demandas de avaliação de banco de dados, caso algum *script* de banco de dados seja entregue e avaliação da documentação para verificação dos casos de uso ou das histórias de usuário. A avaliação de metodologia, apesar de ter sido descontinuada pela CGSI, ainda apareceu nos registros retornados no banco de dados, pois era um item de validação utilizado no início do contrato. Para as OSs de execução, as avaliações são referentes à: banco de dados, arquitetura, testes, e documentação (caso tenha sido alterada em tempo de execução). Foram verificadas as taxas de inconformidades encontradas em cada tipo de avaliação citada acima, não importando se foram ‘não conforme com ressalvas’ ou ‘não conforme’. Esse percentual de inconformidades sinaliza aonde está o gargalo das avaliações, qual o tipo de avaliação que geralmente possui maior taxa de não conformidade.

Atualmente os atrasos são causados e evidenciados em um primeiro momento em dois grandes tipos:

1. Erro de mensuração do Prazo - Significa que a OS atrasou, não sendo entregue na data prevista no sistema CCP. Geralmente se deve pelo fato da funcionalidade ser mais complexa do que o prazo que havia sido estimado pela equipe do projeto, para a sua conclusão.
2. Falta de qualidade da entrega - Muitas OSs são entregues no prazo, porém após o primeiro ciclo de validação são encontradas inconformidades e essas OSs acabam sendo devolvidas para a correção, gerando retrabalho da equipe;

Como mostra a Figura 4.10, consegue-se verificar que o percentual de atraso na entrega das OSs, devido a mensuração incorreta do prazo de desenvolvimento do *software*, corresponde apenas a 28.7% de todos os atrasos. Ou seja, atrasos por erro na mensuração do esforço para o desenvolvimento da funcionalidade são pequenos se comparados ao percentual de OSs entregues no prazo porém com problemas que são encontrados, durante as avaliações de: arquitetura, banco de dados, documentação ou na etapa de testes.

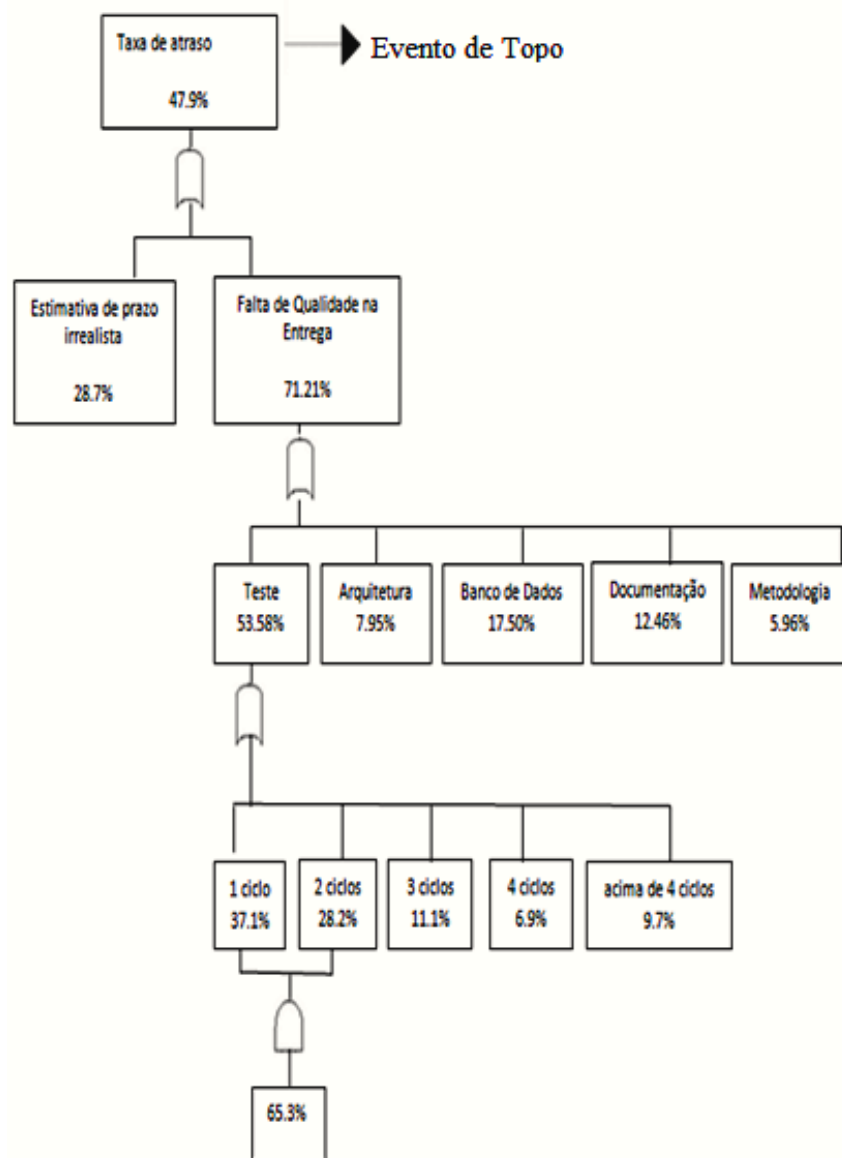


Figura 4.10: Análise da Árvore de Falhas - Analisa as causas dos atrasos das OSs. Fonte: Elaboração própria.

O percentual de atraso devido ao retrabalho da equipe gasto na correção da OS corresponde a mais de 70%. Indo mais além, verifica-se que a principal fonte de não conformidades encontradas durante a avaliação de testes, equivale a 53,58% de todas as inconformidades encontradas, sendo que 65,3% de todas as OSs em atraso passam por até dois ciclos de realização de testes, o que colabora em muito, para o atraso final do projeto.

O que se pode concluir é que 53,58% de todas as inconformidades encontradas, são descobertas durante a execução de testes. Este fato convida a se prestar atenção ao ciclo de desenvolvimento de software, por meio da gestão de riscos, para que os problemas

possam ser descobertos mais cedo.

4.3.4 Impacto do Retrabalho na Cadeia de Valor e Recomendações Finais

Conforme exibido na Figura 4.11, que ilustra o ciclo da ordem de serviço, cada vez que erros são encontrados nas avaliações solicitadas via demanda, ou durante o processo de homologação pelo cliente, a OS é retornada à fábrica, devendo ser corrigida em até 48 horas, segundo o contrato, gerando um retrabalho para a equipe e refletindo em atrasos nos projetos.

Após realizada a correção das não conformidades apontadas, a fábrica reentrega a OS, e o líder de projeto abre uma demanda para que a equipe que realizará a reavaliação verifique se os erros foram corrigidos na aplicação. Porém, esta demanda, ao ser aberta, vai para o final da fila do setor responsável e fica aguardando atendimento. Quando este ciclo ocorre mais de uma vez, como é o caso evidenciado na Figura 4.10, onde a maioria das OSs da CGSI passam por até 2 ciclos de testes, o atraso da OS excede facilmente mais de uma semana.

Esse tipo de atraso em cada OS de um mesmo projeto faz com que o projeto inteiro atrase. Com base nestes resultados, a gerência de riscos se faz necessária para que a CGSI consiga gerenciar os riscos dos projetos de *software* atacando a causa raiz dos problemas, ou seja, identificando, tratando e planejando ações de resposta a estes riscos a fim de mitigá-los, diminuindo o percentual de não conformidades encontradas e sucessivamente o de atraso.

Ou seja, todo o ciclo de retrabalho poderia ser minimizado, caso os riscos fossem geridos durante o processo de desenvolvimento do software. A Árvore de Falhas identificou inconformidades em todos os tipos de avaliação e identificou que algumas OSs passaram até por mais de 4 ciclos de testes, o que deixaria inviável qualquer prazo acordado com o cliente.

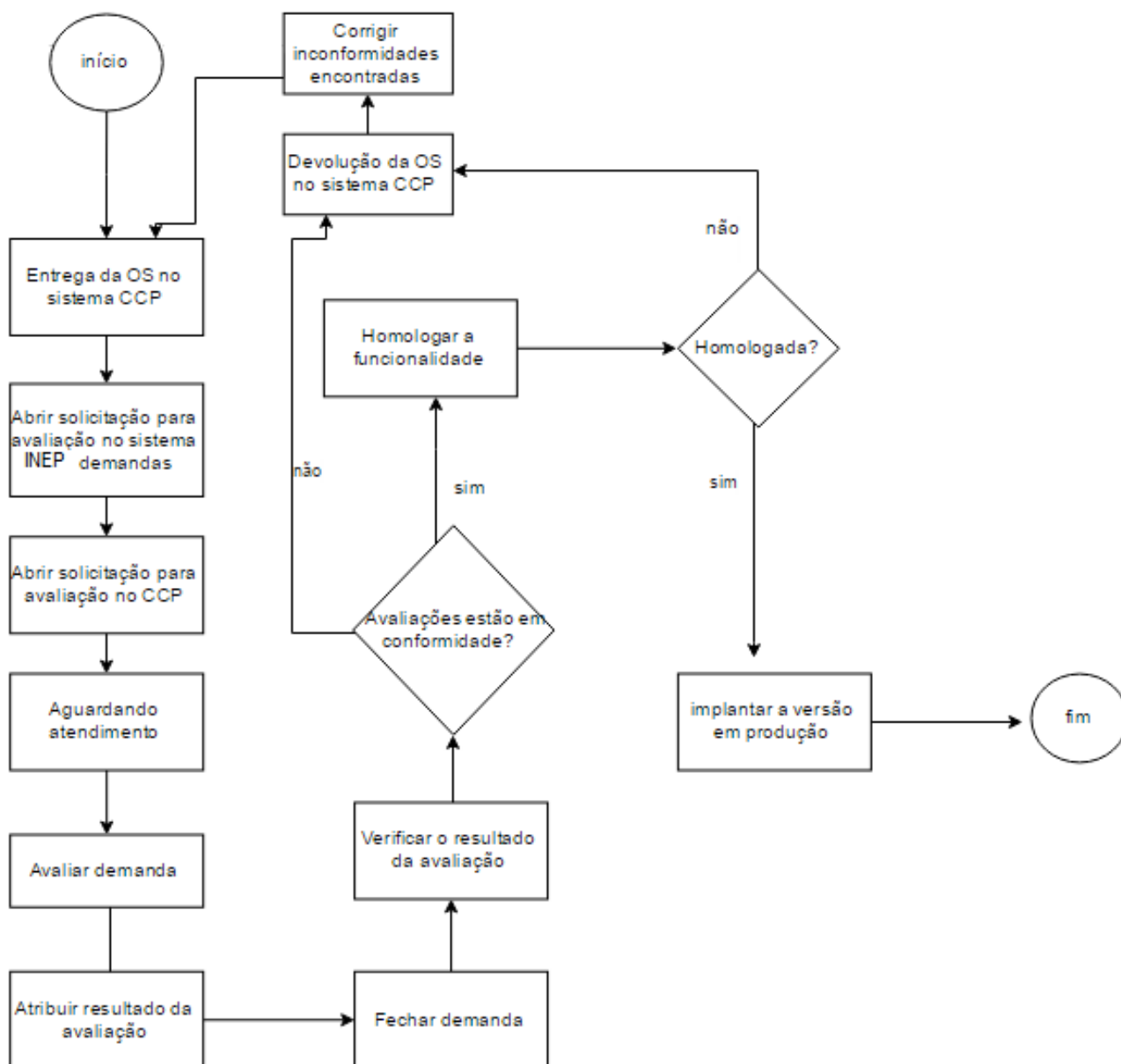


Figura 4.11: Ciclo da Ordem de Serviço. Fonte: Elaboração própria.

A Tabela 4.2, evidencia o atraso das OSs referentes aos sistemas solicitados por cada diretoria do INEP. Os prazos executados em média das OSs de cada diretoria equivalem a mais que o dobro da média em dias do que foi previsto, independente do tipo da OS.

Pode-se verificar que as diretorias DEED, DGP e DAEB juntas demandam mais de 70% dos serviços de TI. Coincidentemente, nestas diretorias estão vinculados os projetos dos censos da educação básica e superior e vários exames e avaliações, tais como: ENEM, ENCCEJA, SAEB, CENSOS, PROVINHA BRASIL, dentre outros.

É visível a forma como o retrabalho afeta os projetos propagando atrasos por toda

Tabela 4.2: Cluster de prazo estimado x executado por diretoria. Fonte: Elaboração própria

cluster	1	2	3	4	5	6
Tamanho	27,5% (257)	25,8%(241)	21,1%(197)	10,6%(99)	8,7%(81)	6,3%(58)
Diretoria	DEED	DGP	DAEB	DAES	DTDIE	DIRED
Prazo estimado	7,41	6,17	7,76	6,6	5,94	7,28
Prazo executado	15,54	12,64	18,86	15,11	12,81	20,41

a cadeia de valor. Por conseguinte, afeta as áreas de negócio, pois atrasa a chegada do software nos ambientes de homologação e produção.

Há necessidade de uma adoção da gestão de riscos nos projetos de TI que reflita todas as áreas do processo de desenvolvimento de *software* para minimizar os percentuais de inconformidades encontrados e exibidos na Figura 4.10. A elaboração da metodologia de gestão de riscos para a CGSI visa auxiliar os líderes de projetos de *software* a realizarem de forma padronizada a gestão de riscos em seus projetos, e os guiará, informando como poderá ser realizado em conjunto com o *Scrum*.

Com base no diagnóstico da situação atual baseado na análise da taxa de atrasos das OSs e nas causas raízes das inconformidades encontradas durante a aferição da qualidade, foi possível verificar que a maioria atrasa por inconformidades encontradas durante a execução dos testes, o que gera um retrabalho devido a correção de erros e defeitos, atrasando sua entrega. Tais inconformidades são oriundas de riscos que poderiam ser mitigados caso houvesse uma gestão de riscos durante o processo de desenvolvimento de software. Com base nesta situação, a próxima seção deste trabalho, apresentará a proposta de metodologia de gestão de riscos que explica como poderá ser realizado o processo da gestão de riscos dentro das fases do *Scrum* e quais ferramentas podem ser utilizadas.

Capítulo 5

Proposta de Metodologia de Gestão de Riscos da CGSI

5.1 Contextualização

A metodologia de gestão de riscos proposta neste trabalho se trata de uma integração da gestão de riscos tradicional ao processo ágil de desenvolvimento de software utilizado nesta coordenação. Esta iniciativa está alinhada com o PDTI (2016-2019), em seu objetivo estratégico 7: ‘Aprimorar a gestão de projetos’ e metas a seguir:

- Meta 3 - ‘Aumentar a efetividade das ações de governança de TI’.
- Meta 6 - ‘Aumentar as ações de modelagem, definição de processos e metodologias’.

A metodologia proposta é um conjunto de conceitos, processos, papéis, técnicas e ferramentas que tem por finalidade nortear a atuação dos envolvidos na condução da gestão de riscos em projetos de desenvolvimento ágil de software. Esta metodologia é compatível com a INC nº 1, de 10 de maio de 2016, publicada pela CGU e pelo MP, que dispõe sobre controles internos, gestão de riscos e governança no âmbito do Poder Executivo Federal. A INC CGU/MP nº 01/2016 tornou obrigatória aos órgãos e entidades do Poder Executivo Federal a adoção de medidas para a sistematização de práticas relacionadas à gestão de riscos, aos controles internos e à governança. A metodologia proposta também é compatível com as normas técnicas NBR ISO/IEC 31000:2009, 31010:2012 e NBR ISO/TR 31004:2015 que discorrem sobre a gestão de riscos, técnicas e ferramentas para a sua adoção e implementação da gestão de riscos, respectivamente [1], [2], [3]. Leva ainda em consideração, os padrões já adotados internamente, tais como:

- *Framework Scrum* (*framework* adotado para desenvolvimento dos sistemas)
- MGDS (Metodologia Geral de Sistemas de Informação) e

- MGP (Metodologia de Gestão de Projetos).

Todos fornecem elementos para a descrição da metodologia proposta para a CGSI. Porém as definições trazidas na INC CGU/MP n 01/2016, são soberanas em relação a qualquer definição assumida internamente que não esteja alinhada a ela. Esta metodologia poderá ainda ser apoiada por software computacional, porém é independente de qualquer ferramenta.

5.2 Conceitos

Para os efeitos desta metodologia, aplicam-se os seguintes termos e definições conforme NBR ISO/IEC 31000:

Gestão de Risco - Atividades coordenadas para dirigir e controlar uma organização no que se refere a riscos.

Estrutura da Gestão de Riscos - Conjunto de componentes que fornecem os fundamentos e os arranjos organizacionais para a concepção, implementação, monitoramento, análise crítica e melhoria contínua da gestão de riscos através de toda a organização [64].

- Os fundamentos incluem a política, objetivos, mandatos e comprometimento para gerenciar riscos;
- Os arranjos organizacionais incluem planos, relacionamentos, responsabilidades, recursos, processos e atividades;
- A estrutura da gestão de riscos está incorporada no âmbito das políticas e práticas estratégicas e operacionais de toda a organização.

Processo de gestão de riscos - Aplicação sistemática de políticas, procedimentos e práticas de gestão para as atividades de comunicação, consulta, estabelecimento do contexto, identificação, análise, avaliação, tratamento, monitoramento, e análise crítica dos riscos.

Proprietário/dono do risco - Pessoa ou entidade com a responsabilidade e a autoridade para gerenciar um risco.

Estabelecimento do contexto - Definição dos parâmetros externos e internos a serem levados em consideração ao gerenciar riscos, e estabelecimento do escopo e dos critérios de risco para a política de gestão de riscos.

Comunicação e consulta - Processos contínuos e iterativos que uma organização conduz para fornecer, compartilhar ou obter informações e se envolver no diálogo com as partes interessadas e outros, com relação a gerenciar riscos.

Identificação de riscos - Processo de busca, reconhecimento e descrição de riscos. A identificação de riscos envolve a identificação das fontes de risco, eventos, suas causas e suas consequências potenciais. Pode envolver dados históricos, análises teóricas, opiniões de pessoas informadas e especialistas, e as necessidades das partes interessadas.

Processo de Avaliação de Risco - Processo global de identificação de riscos, análise de riscos e avaliação de riscos.

Fonte de risco - Elemento que, individualmente ou combinado, tem o potencial intrínseco para dar origem ao risco. Uma fonte de risco pode ser tangível ou intangível.

Consequência - Resultado de um evento que afeta os objetivos;

- Um evento pode levar a uma série de consequências;
- Uma consequência pode ser certa ou incerta e pode ter efeitos positivos ou negativos sobre os objetivos;
- As consequências podem ser expressas qualitativa ou quantitativamente;
- As consequências iniciais podem desencadear reações em cadeia.

Probabilidade - Chance de algo acontecer. Na terminologia de gestão de riscos, a palavra "probabilidade" é utilizada para referir-se à chance de algo acontecer, não importando se definida, medida ou determinada objetiva ou subjetivamente, qualitativa ou quantitativamente, ou se descrita utilizando-se termos gerais ou matemáticos (tal como probabilidade ou frequência durante um determinado período de tempo).

Perfil de risco - Descrição de um conjunto qualquer de riscos. O conjunto de riscos pode conter riscos que dizem respeito a toda a organização, parte da organização, ou referente ao qual tiver sido definido.

Análise de riscos - Processo de compreender a natureza do risco e determinar o nível de risco.

- A análise de riscos fornece a base para a avaliação de riscos e para as decisões sobre o tratamento de riscos;
- A análise de riscos inclui a estimativa de riscos.

Nível de risco - Magnitude de um risco ou combinação de riscos, expressa em termos da combinação das consequências e de suas probabilidades.

Avaliação de riscos - Processo de comparar os resultados da análise de riscos com os critérios de risco para determinar se o risco e/ou sua magnitude é aceitável ou tolerável.

Critérios de risco - Termos de referência contra os quais a significância de um risco é avaliada;

- Os critérios de risco são baseados nos objetivos organizacionais e no contexto externo e contexto interno;
- Os critérios de risco podem ser derivados de normas, leis, políticas e outros requisitos.

Comunicação e consulta - Processos contínuos e iterativos que uma organização conduz para fornecer, compartilhar ou obter informações e se envolver no diálogo com as partes interessadas e outros, com relação a gerenciar riscos. As informações podem referir-se à existência, natureza, forma, probabilidade, significância, avaliação, aceitabilidade, tratamento ou outros aspectos da gestão de riscos. A consulta é um processo bidirecional de comunicação sistematizada entre uma organização e suas partes interessadas ou outros, antes de tomar uma decisão ou direcionar uma questão específica. A consulta é:

- Um processo que impacta uma decisão através da influência ao invés do poder e
- Uma entrada para o processo de tomada de decisão, e não uma tomada de decisão em conjunto.

Tratamento de riscos - Processo para modificar o risco. O tratamento de risco pode envolver:

- A ação de evitar o risco pela decisão de não iniciar ou descontinuar a atividade que dá origem ao risco;
- Assumir ou aumentar o risco, a fim de buscar uma oportunidade;
- A remoção da fonte de risco;
- A alteração da probabilidade;
- A alteração das consequências;
- O compartilhamento do risco com outra parte ou partes (incluindo contratos e financiamento do risco); e
- A retenção do risco por uma escolha consciente. Os tratamentos de riscos relativos às consequências negativas são muitas vezes referidos como ‘mitigação de riscos’, ‘eliminação de riscos’, ‘prevenção de riscos’ e ‘redução de riscos’. O tratamento de riscos pode criar novos riscos ou modificar riscos existentes.

Controle - Medida ou ação que está modificando o risco. Os controles incluem qualquer processo, política, dispositivo, prática ou outras ações que modificam o risco. Os controles nem sempre conseguem exercer o efeito de modificação pretendido ou presumido.

Risco residual - Risco remanescente após o tratamento do risco. O risco residual pode conter riscos não identificados.

Monitoramento

- Verificação, supervisão, observação crítica ou identificação da situação, executadas de forma contínua, a fim de identificar mudanças no nível de desempenho requerido ou esperado;
- O monitoramento pode ser aplicado à estrutura da gestão de riscos, ao processo de gestão de riscos, ao risco ou ao controle.

Análise crítica

- Atividade realizada para determinar a adequação, suficiência e eficácia do assunto em questão para atingir os objetivos estabelecidos;
- A análise crítica pode ser aplicada à estrutura da gestão de riscos, ao processo de gestão de riscos, ao risco ou ao controle.

5.3 Nível de Atuação da Metodologia

A metodologia de gestão de riscos proposta é uma integração das atividades da gestão de riscos com o *framework Scrum* de desenvolvimento de software ágil. Tal proposta se atém ao contexto interno da CGSI e aos seus projetos e sistemas. O contexto interno da CGSI foi explanado detalhadamente no Capítulo 4.

5.4 Objetivo

Os órgãos e entidades do Poder Executivo Federal devem implementar, manter, monitorar e revisar o processo de gestão de riscos, de forma a ser compatível com sua missão e seus objetivos estratégicos. Os controles internos da gestão propiciam o alcance de seus objetivos. Devem ser estruturados para oferecer segurança razoável de que os objetivos da organização serão alcançados [64]. Possui três objetivos principais:

1. Assegurar que os tomadores de decisão tenham acesso tempestivo a informações quanto ao risco aos quais a organização está exposta para determinar questões de delegação, se for o caso;
2. Aumentar a probabilidade de alcance dos objetivos estratégicos da organização, reduzindo os riscos a níveis aceitáveis; e

3. Agregar valor à organização por meio de melhoria dos processos de tomada de decisão e do tratamento adequado dos riscos e dos impactos negativos decorrentes de sua materialização [64].

Este trabalho atua em nível de coordenação, de forma mais específica, alinhando-se aos objetivos estratégicos citados no PDTI. Segundo a NBR ISO/IEC 31000:2009, a gestão de riscos pode ser aplicada a toda uma organização, em suas várias áreas e níveis, a qualquer momento, bem como a funções, atividades e projetos específicos [1]. Desta forma, é aplicada à CGSI repetindo de igual maneira os objetivos principais citados acima oriundos da INC MP/CGU N° 01/2016 [64].

5.5 Papéis

Cada risco mapeado e avaliado deverá estar associado a um agente responsável formalmente identificado, que é o dono do risco. O dono do risco é a pessoa com responsabilidade para gerenciar determinado risco [64]. Os gestores são os responsáveis pela avaliação dos riscos no âmbito das unidades, processos e atividades que lhe são afetos. Por esta razão, foram definidos papéis para que fosse possível a realização da associação entre risco e dono do risco. Abaixo, estão listados os papéis envolvidos na gestão de riscos dos projetos:

- Gestor de Risco - Gestor ligado à governança de TI com autoridade suficiente para orientar e acompanhar as ações de mapeamento, avaliação e mitigação do risco;
- Líder de Projeto - Gestor do projeto, ou o fiscal técnico, responsável por elaborar e manter o relatório de gestão e comunicação dos riscos atualizado durante as fases do projeto;
- Cliente - Representante da área demandante do projeto ou fiscal requisitante do projeto;
- Coordenador Geral - Fornece o apoio gerencial para a realização do projeto e assim da gestão de riscos sendo responsável junto ao líder de projetos pelo sucesso do mesmo;
- Coordenador das áreas técnicas envolvidas - Representante responsável por coordenar as equipes de arquitetura, testes e administração de dados, envolvidas no projeto, mas não diretamente alocadas nele;
- Preposto da Fábrica - Responsável por representar a contratada e pelo atendimento às questões contratuais;

- *Scrum Master* - Representante da fábrica de software contratada que deve realizar a ponte entre a equipe de desenvolvimento e o líder de projeto, principalmente garantindo que o processo *Scrum* seja seguido. Os riscos cuja responsabilidade de mitigação seja de sua equipe, são associados formalmente a ele.

Os papéis criados acima são sugestões e podem ser alterados ou complementados todas as vezes que se fizer necessário.

5.6 Proposta para processo de gestão de riscos integrada ao *Scrum*

Com base no modelo de desenvolvimento de software ágil adotado na coordenação, é proposto o modelo de gestão de riscos adaptado da Extensão de Software do PMI [76]. A Extensão de Software do PMI pretende fazer a ligação entre os processos mais estruturados de gestão de projetos, mencionados no PMBOK e os métodos ágeis de gestão de projetos. Por isso, o modelo ágil sugerido pelo PMI para gestão de riscos mostrou maior integração entre atividades da gestão de riscos dentro das cerimônias do *Scrum* em relação aos demais modelos investigados no Capítulo 2, item 2.8. A adaptação levou em conta a necessidade de adição explícita das atividades de tratamento e monitoramento dos riscos, assim como a necessidade de descrever as ferramentas e técnicas que poderiam ser utilizadas em cada atividade. Para isso o estudo realizado no Capítulo 2, item 2.7 foi de suma importância para a elaboração deste processo de gestão de riscos ágil. Baseado no processo de desenvolvimento de software do *framework Scrum*, a gestão de riscos é proposta para ser ágil e integrada às suas atividades e cerimônias.

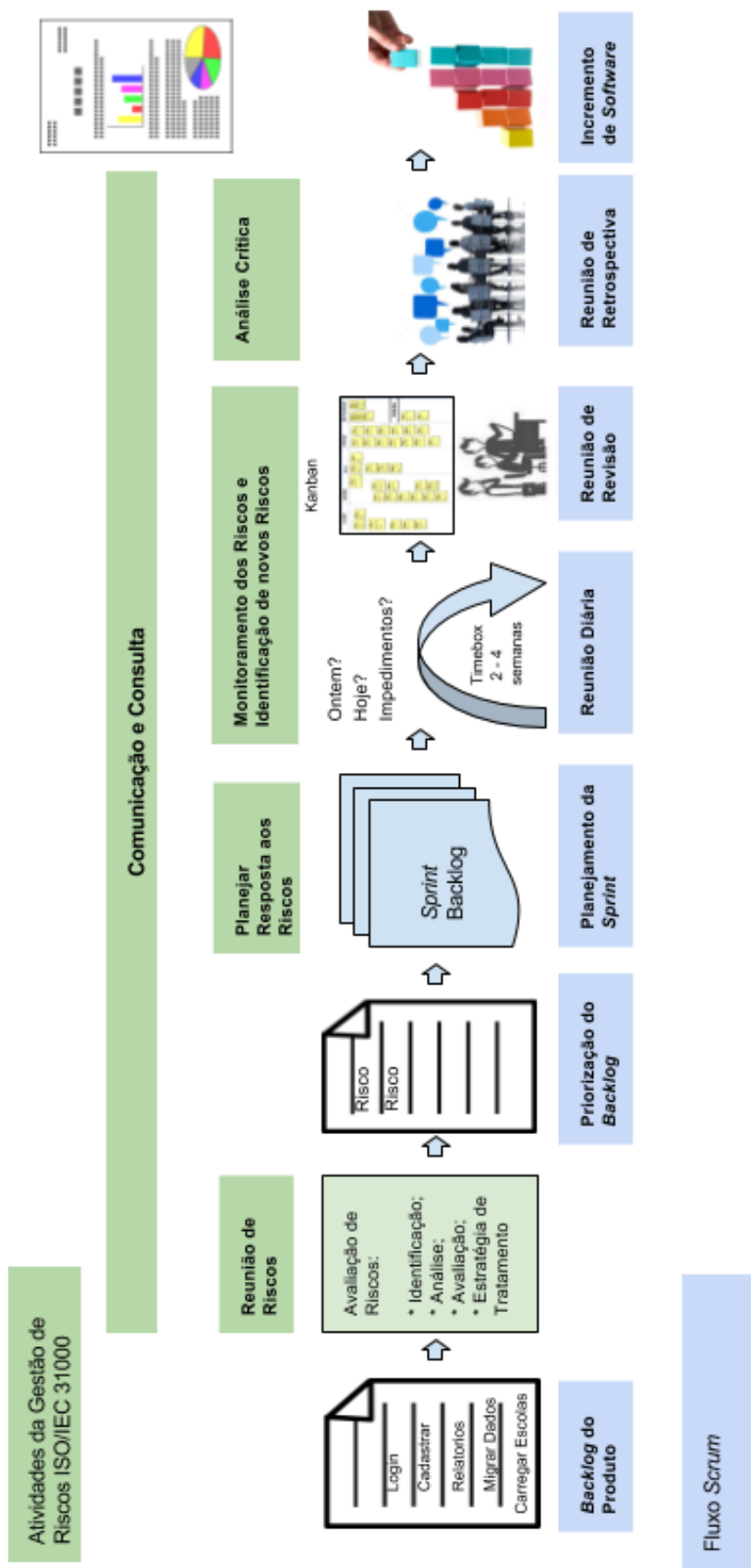


Figura 5.1: Proposta de processo de gestão de riscos integrada ao Scrum. Fonte: Elaboração Própria.

Conforme Figura 5.1, de posse do *backlog* do produto, ou seja, da lista de novas funcionalidades e manutenções adaptativas ou corretivas do produto, é realizada a **Reunião de Riscos**, que inclui a identificação dos riscos inerentes ao desenvolvimento do produto analisado. Nesta mesma reunião também é realizada a análise e a avaliação que determina quais funcionalidades possuem os riscos mais críticos e altos e, assim, estas são priorizadas no *backlog*. A definição da estratégia de tratamento dos riscos (aceitar, mitigar, transferir ou evitar) também é realizada nesta reunião.

Durante a **reunião de planejamento da (*Sprint*)**, o tratamento do risco e suas ações de resposta são definidas e descritas. Neste momento, a equipe do projeto estará reunida e de posse dos requisitos detalhados poderão analisar a avaliação inicial de priorização do *backlog* pelo nível de risco e definir uma ação de resposta para cada risco. Também é possível, neste momento, atribuir a cada risco o ‘dono do risco’.

Quando a iteração se inicia os riscos são monitorados diariamente na **reunião diária** do *Scrum*. A identificação de novos riscos também pode ocorrer durante a *Sprint*, a medida que o entendimento da equipe aumenta e possíveis impedimentos vão aparecendo.

Durante a **reunião de revisão da iteração**, quando a equipe apresenta o incremento de software ao cliente, caso sejam solicitadas alterações, novos riscos podem ser identificados, e assim os mesmos deverão compor o *backlog* que sofrerá repriorização.

Ao final da *sprint*, durante a **reunião de retrospectiva**, quando a equipe se reúne para levantamento das lições aprendidas, deverá haver uma análise sobre quais riscos inerentes ao desenvolvimento, poderão ser fechados e quais poderão ser levados para a próxima iteração.

A análise crítica poderá ser realizada tanto no momento da **reunião de retrospectiva** da *Sprint* quanto agendada de modo a levantar se as ações de resposta aos riscos estão sendo efetivas, se os riscos estão se manifestando e se o processo de gestão de riscos ainda está adequado ao contexto.

5.6.1 Mapeamento da Gestão de Riscos no *framework Scrum*

O Quadro 5.1, mostra o mapeamento da gestão de risco no *framework Scrum*.

A identificação dos riscos ocorre durante a maior parte das cerimônias definidas no *framework Scrum*. Podem ser utilizados os artefatos visão do produto, *backlog* e *roadmap* do produto para se identificar os riscos do projeto. Isto poderá ser realizado nas reuniões de planejamento, reuniões diárias e reuniões de revisão, pois a todo momento podem ser encontrados novos riscos na iteração, exceto na cerimônia da retrospectiva da *Sprint*, pois esta reunião indica o fim da iteração corrente.

As fases de análise e avaliação são comumente realizadas durante a reunião de planejamento da *sprint*, pois o ideal é que seja realizada juntamente com a equipe antes do início

Quadro 5.1: Mapeamento da Gestão de Riscos no *Scrum*. Fonte: Adaptado de Nyfjord, et al. (2008)[69]

Scrum / Gestão de Riscos	Identificação de Riscos	Análise e Avaliação de Riscos	Tratamento e Resposta aos Riscos	Monitora- mento	Comuni- cação e Consulta
Visão	x	x			x
Backlog e Roadmap	x	x	x		x
Planejamento da Sprint	x	x	x		x
Reunião Diária	x	x	x	x	x
Revisão	x			x	x
Retrospectiva					x

da iteração, assim como o tratamento e resposta aos riscos que também deverá ocorrer no planejamento da *Sprint*. Porém eventualmente, algum risco pode ser identificado, avaliado e tratado durante as reuniões diárias, por ser inerente à *sprint* corrente e conforme o entendimento dos requisitos for aumentando, novos riscos vão sendo reconhecidos pela equipe.

O monitoramento é frequentemente realizado durante as reuniões diárias, momento este em que todos os membros da equipe estão reunidos, dentre outras coisas, para identificação de possíveis impedimentos na implementação da *Sprint* corrente, assim os riscos já tratados são diariamente monitorados e novos riscos podem ser revelados.

Na revisão, onde o incremento de software funcional é exibido para cliente ao final da fase de construção, podem surgir novos riscos oriundos de novas solicitações realizadas pelo cliente. Porém, estes riscos deverão ser encaminhados para o *backlog* juntamente com as novas solicitações para aguardarem nova priorização. O monitoramento também é realizado nesta fase para verificar se algum risco se materializou.

A retrospectiva é realizada após a entrega do incremento do software e tem por objetivo também de fechar riscos e analisar se algum risco desta iteração tem a probabilidade de ocorrer na próxima iteração.

A análise crítica é utilizada para verificar até que ponto as respostas estão reduzindo a probabilidade ou o impacto dos riscos que foram identificados. O *feedback* da equipe passado na reunião de retrospectiva da *Sprint* poderá auxiliar o Gestor de Riscos ou o auditor na atividade de análise crítica, que analisará o processo da gestão de riscos, sua estrutura e os próprios riscos. A comunicação e consulta permeia toda a gestão tradicional de riscos e deverá permear todas as cerimônias do *framework Scrum* para que as partes envolvidas sejam consultadas quanto aos riscos inerentes ao projeto e também possam ser igualmente comunicadas sobre o resultado da gestão de riscos.

Note que a comunicação e consulta é uma atividade constante da gestão de riscos.

Ela ocorre paralelamente as demais atividades, devendo haver uma consulta às partes interessadas no processo para que a gestão de riscos seja efetiva e, de igual maneira estas mesmas partes interessadas necessitam ser comunicadas sobre os riscos. Esta comunicação é importante para a tomada de decisões, delegações de riscos, realização das ações mitigadoras pelo dono do risco e no geral, pela ciência de todos os envolvidos no resultado final de sua gestão.

5.7 Fluxograma das Atividades da Gestão de Riscos com Técnicas e Ferramentas a serem Utilizadas

O fluxograma da Figura 5.2 ilustra a proposta da gestão de riscos no âmbito da CGSI, informando sequencialmente os passos que devem ser seguidos para a gestão de riscos, assim como as ferramentas a serem utilizadas em cada etapa e os artefatos gerados.

Propõe-se que a gestão de riscos seja iniciada com a definição do projeto. Para a atividade de identificação dos riscos, a metodologia sugere que seja realizada em dois passos: riscos comuns em projetos de software e riscos específicos do projeto selecionado.

Para o levantamento dos riscos comuns em projetos de software, foi elaborado um *checklist* de riscos para uso dos projetos da CGSI, que auxiliará no levantamento de riscos em várias categorias, como: riscos de modelagem, de requisitos, de arquitetura, riscos na fase de realização dos testes, riscos na implantação, no contrato, dentre outros. Isso não significa que todos os riscos elencados poderão ocorrer no projeto analisado, por isso se faz necessária a colaboração dos membros da equipe para efetivamente verificar quais poderão de fato impactar nos planos do projeto. Portanto, a proposta é que seja utilizado o *brainstorming* para realizar esta identificação dos riscos juntamente com a equipe do projeto. Os riscos específicos do projeto podem ser identificados também baseando-se na documentação específica do projeto: *backlog* do produto, *roadmap* e histórias de usuário.

A análise e avaliação dos riscos é realizada por intermédio da matriz de probabilidade e impacto dos riscos. Também conta com a colaboração da equipe para, por meio de *brainstorming* realizado durante a reunião de planejamento da *Sprint*, serem analisados e avaliados os níveis de cada risco.

O tratamento dos riscos é definido para que os riscos que possuam maior nível de criticidade sejam priorizados em detrimento aos riscos que possuam níveis aceitáveis. O planejamento de resposta aos riscos se dá pela descrição de uma ou mais ações para minimizar, evitar ou transferir o risco identificado. Nesta etapa, é necessária a opinião de especialista para definição de ações efetivas de mitigação.

O monitoramento dos riscos é realizado durante as reuniões diárias do *Scrum*, quando são tratados os impedimentos da iteração corrente. Neste momento, a equipe possui um

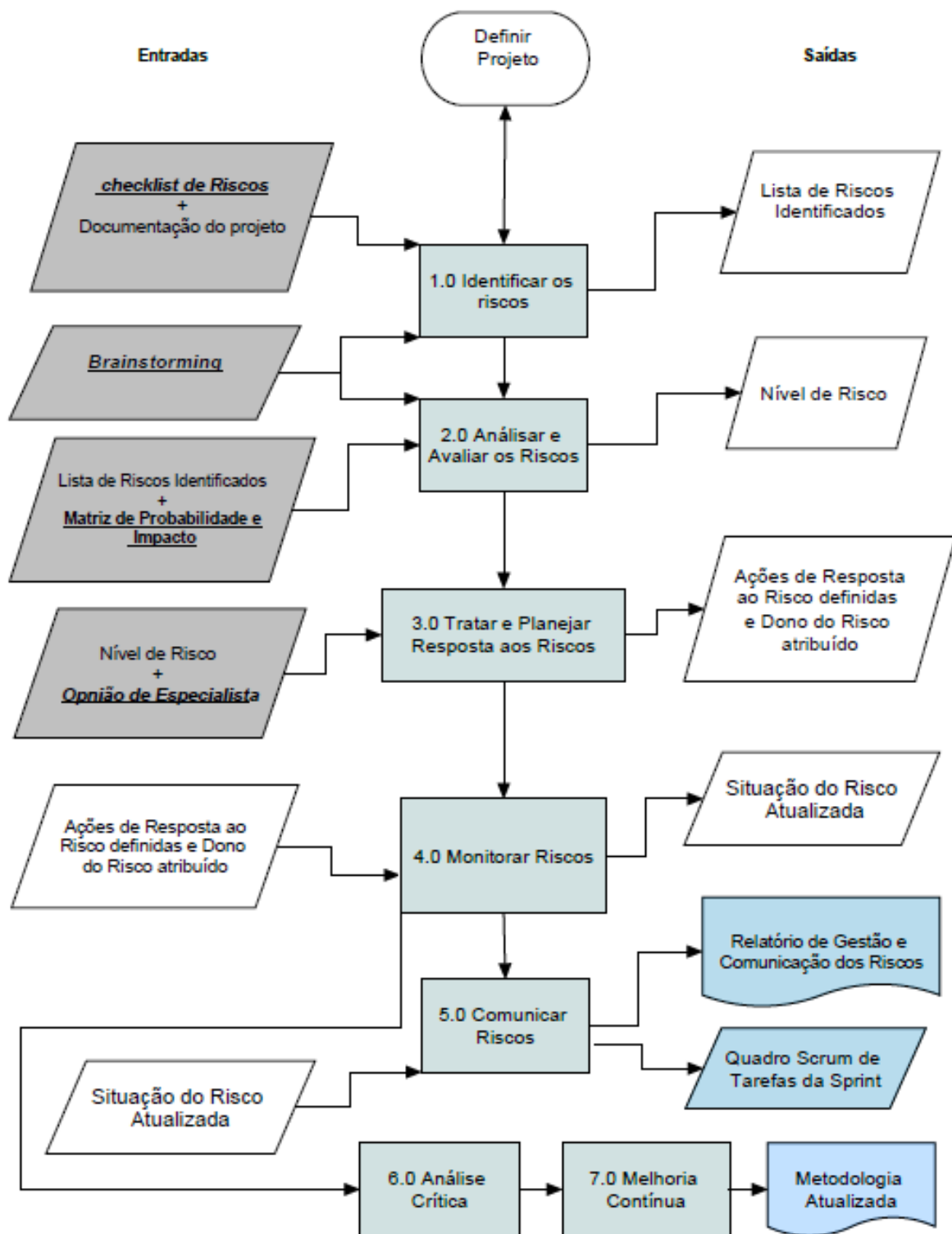


Figura 5.2: Fluxograma proposto para a Metodologia de Gestão de Riscos da CGSI. Fonte: Elaboração própria.

grau de entendimento maior sobre os itens da *Sprint* e assim terá condições de monitorar os riscos já tratados e identificar novos riscos.

A comunicação dos riscos é uma via de mão dupla: a comunicação recebe os riscos identificados e os reporta às partes interessadas do projeto. A comunicação ocorre paralelamente a todas as demais atividades da gestão de riscos, embora no fluxograma exibido na Figura 5.2, ele pareça ocorrer apenas ao final. É proposto um quadro de tarefas da *Sprint* que contempla os riscos da iteração corrente e um Relatório de Gestão e Comunicação dos Riscos que consolidará os riscos do projeto, ambos deverão ser atualizados no decorrer de cada *Sprint*. Este relatório servirá para reportar a alta gestão da DTDIE, incluindo a governança de TI, sobre os riscos nos projetos de software da CGSI.

A análise crítica é realizada como evento agendado tal qual uma auditoria no processo da gestão de riscos adotado. É também alimentada por *feedbacks* durante o monitoramento dos riscos. A análise crítica é o início da melhoria contínua que tem como finalidade as adaptações e evoluções de melhoria na metodologia proposta, que deverá ser atualizada sempre que se fizer necessário.

Para efeito de informação, o *checklist* de riscos da CGSI e o modelo do Relatório de Comunicação e Gestão de Riscos farão parte dos apêndices A e B deste trabalho, respectivamente.

O detalhamento de cada etapa deste fluxograma proposto para a Gestão de Riscos da CGSI é apresentado a seguir.

5.7.1 Método de Identificação de Riscos

A Identificação de Riscos é uma tentativa de identificar ameaças ao plano do projeto (cronograma, custo, recursos, qualidade, comunicação etc.), por isso, necessita ser realizada na fase de planejamento do projeto.

Existem dois tipos de riscos:

1. Riscos comuns, típicos ou genéricos - que são uma ameaça em potencial a todo projeto de software e
2. Riscos específicos do produto - podem ser identificados apenas por aqueles que têm uma visão clara da tecnologia, dos requisitos funcionais e não funcionais e dos usuários para os quais o software está sendo desenvolvido [79].

Elaboração do Checklist de Identificação de Riscos Comuns nos Projetos da CGSI

Uma boa prática para se identificar riscos comuns de um projeto de software é criar um checklist de itens de risco [79] [50] [28] [63]. A identificação dos riscos, conforme exibida

no macro processo da gestão de riscos, sugere que seja utilizada uma lista ou *checklist* de riscos de projetos de software. Esta lista tem por função auxiliar na identificação de riscos em diversas áreas e fases do desenvolvimento de software, servindo como guia, no processo de identificação de riscos típicos ou comuns [79]. Mais do que isto, ela é um perfil de riscos inerentes aos projetos de software da CGSI. A lista ou *checklist* de riscos foi criada tendo por base as fraquezas listadas na análise SWOT do PDTI, pois estão focadas nos riscos inerentes ao ambiente interno [66] [86]. Assim como nos 16 riscos em se utilizar metodologias ágeis, citados no relatório do TCU [94] e em riscos comuns de projetos de software, identificados por meio de revisão bibliográfica [95]. A validação da lista foi realizada da seguinte forma: Todos os riscos foram separados por categorias, e encaminhados via e-mail aos profissionais representantes das diversas áreas participantes do processo de desenvolvimento de software (contratadas, arquitetura, administração de dados, modelagem de processos, teste, desenvolvimento, gerência de projeto e etc). Assim, foi possível que realizassem a validação dos riscos já evidenciados e a inclusão de novos riscos. É importante que a lista de riscos seja atualizada sempre que necessário, para que a mesma se torne uma base histórica de riscos alinhada com a realidade dos projetos. A lista de riscos não tem por finalidade o objetivo de esgotar todos os riscos, pois ela não identifica os riscos específicos do produto. Para isso, além da lista de riscos, deverão ser utilizadas outras técnicas e ferramentas de identificação de riscos em conjunto, para que todos os riscos inerentes a determinado projeto possam ser devidamente identificados. Conforme tratado no Capítulo 2 item 2.7, outras ferramentas de identificação de riscos bastante citadas e que podem ser utilizadas durante as reuniões dos projetos, como a entrevista semi-estruturada, e o *Brainstorming* com a equipe envolvida. Enquanto a entrevista é realizada de maneira mais formal e estruturada o *Brainstorming* é realizado de forma simples e com a participação de toda a equipe, incluindo o cliente que, no *Scrum*, integra a equipe. A visão de diversos profissionais agiliza o processo de identificação de riscos quando comparado a entrevistas realizadas individualmente ou que seguem um roteiro definido. Por este motivo, a técnica de *brainstorming* será utilizada para a identificar os riscos específicos do produto.

A Figura 5.3 mostra de forma consolidada o detalhamento do subprocesso Identificar Riscos do fluxograma exibido na Figura 5.2.

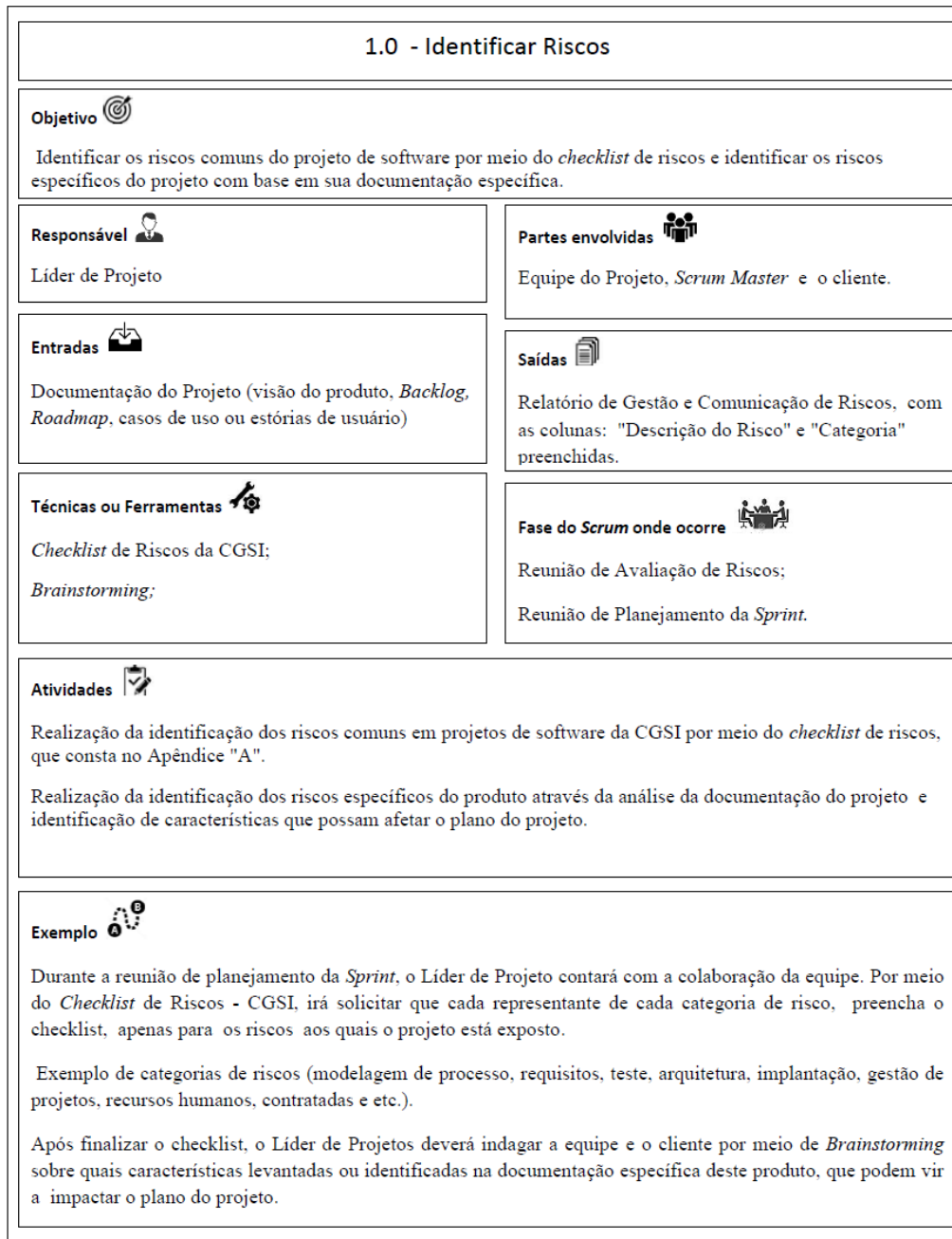


Figura 5.3: Subprocesso Identificar Riscos. Fonte: Elaboração própria.

A identificação de riscos é realizada inicialmente em reunião específica de avaliação de riscos que inclui a identificação, análise e avaliação dos riscos com a equipe técnica e o cliente, logo após a definição dos itens do *backlog*. A equipe se reunirá para identificar os riscos comuns e específicos do projeto.

Para levantar os riscos comuns do projeto, é sugerido que se utilize o *checklist* de riscos, porém para levantar os riscos específicos do projeto, uma boa forma de identificá-los é

fazendo a seguinte pergunta à equipe [79]:

- "Quais são as características especiais existentes nas funcionalidades que podem ameaçar o plano do projeto?"

Os requisitos funcionais fazem parte do escopo do projeto, e os não funcionais dizem respeito às suas restrições. Os requisitos não funcionais são tão importantes quanto os funcionais. Por exemplo:

- O software necessitará possuir recursos de acessibilidade?
- Será compatível com quais *browsers*?
- Necessitará de *captcha* de segurança?
- Deverá seguir quais padrões?
- Possui prazo legal para a sua disponibilização?
- Qual o tempo de resposta, em processamentos específicos, é aceitável?

A identificação de riscos deverá abranger tanto os requisitos funcionais quanto os não funcionais, pois ambos são requisitos do projeto. Por isso, uma avaliação da documentação do projeto necessita ser realizada. Convém que antes da reunião de avaliação de riscos, a equipe leia a documentação do projeto (*Visão do produto*, *Backlog* e *Roadmap*).

Durante o planejamento inicial do projeto, um risco pode ser especificado de forma genérica. À medida que o tempo passa e se conhece mais sobre o projeto e o risco, talvez seja necessário refinar o risco em um conjunto de riscos mais detalhados, cada um deles, será controlado, monitorado e gerenciado mais facilmente [79]. No *Scrum*, o software é construído de forma iterativa e incremental, por isso é comum que a equipe tenha um detalhamento maior apenas dos requisitos que comporão a *Sprint* a ser desenvolvida. Desta forma, a cada iteração no *Scrum*, durante a reunião de planejamento da *sprint*, poderá ser realizada a validação dos riscos identificados anteriormente com base nos itens de *backlog* e a inclusão de novos riscos específicos da *sprint* que será desenvolvida, já que neste momento existe um detalhamento maior sobre os itens que comporão a *Sprint* e a equipe poderá reconhecer riscos antes não detectados.

5.7.2 Método de Análise e Avaliação de Riscos

A análise de riscos é o processo de determinação do nível de risco calculado por meio de sua probabilidade e consequência. A avaliação de risco compara o resultado desta análise para verificar se o risco é tolerável ou aceitável. A Figura 5.4 apresenta de forma consolidada o detalhamento do subprocesso Analisar e Avaliar Riscos do fluxograma exibido na Figura 5.2.

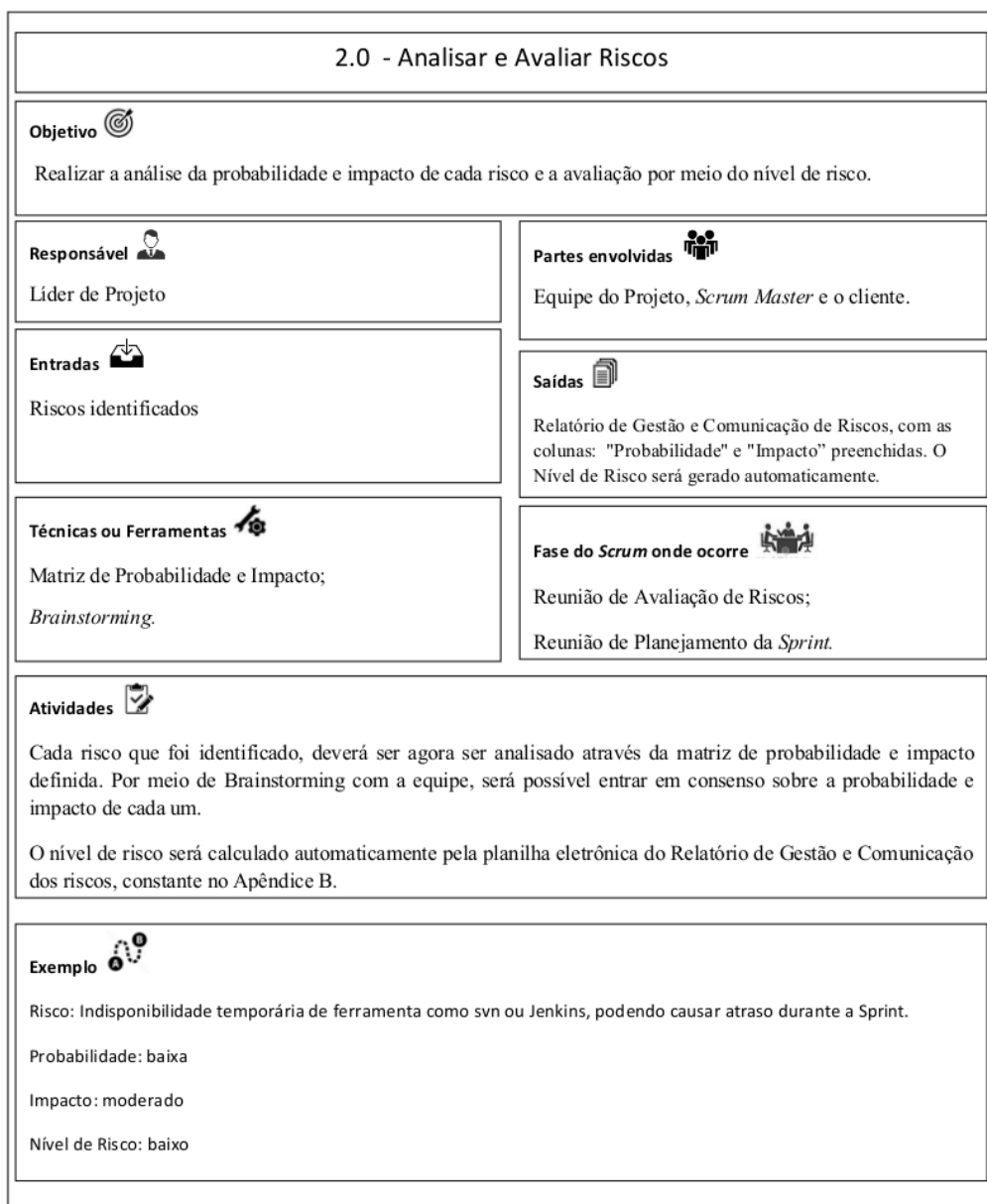


Figura 5.4: Subprocesso Analisar e Avaliar Riscos. Fonte: Elaboração própria.

Segundo a INC MP/CGU Nº 01/2016, a avaliação de riscos deverá ser realizada sob a perspectiva de probabilidade e impacto. Deverá ainda contar com análises quantitativas, qualitativas ou a combinação de ambas. Também deverão ser avaliados quanto as condições de riscos: inerentes (riscos aos quais a CGSI está exposta) e residuais. A probabilidade de cada risco poderá ser calculada com estimativas individuais e depois determinada com um valor de consenso da equipe.

Embasado na MGR-SISP, o Quadro 5.2 foi criado, para classificar o nível de riscos.

Quadro 5.2: Critério de classificação de Nível de riscos. Fonte: Adaptado da MGR-SISP [61]

	Probabilidade	Muito Baixa	Baixa	Moderada	Alta	Muito Alta
Impacto	Muito Baixo	Muito Baixa	Muito Baixa	Baixa	Baixa	Moderada
	Baixo	Muito Baixa	Baixa	Baixa	Moderada	Alta
	Moderado	Baixa	Baixa	Moderada	Alta	Alta
	Alto	Baixa	Moderada	Alta	Alta	Muito Alta
	Muito Alto	Moderada	Alta	Alta	Muito Alta	Muito Alta

As probabilidades podem ser expressas de forma qualitativa [61]:

- Muito Baixa: é altamente improvável que o evento ocorra;
- Baixa: é improvável que o evento ocorra;
- Moderada: é provável que o risco ocorra;
- Alta: é altamente provável que o risco ocorra;
- Muito Alta: é quase certo que o risco irá ocorrer.

As probabilidades também podem ser expressas de forma quantitativa [61]:

- Muito Baixa: entre 10% e 20%;
- Baixa: entre 30% e 40%;
- Moderada: 50%;
- Alta: entre 60% e 70%;
- Muito Alta: entre 80% e 90%.

Um risco com 100% de probabilidade de ocorrência, é uma restrição do projeto [79].

Avaliar o impacto do risco é avaliar as possíveis consequências originadas pela materialização dele. O escopo do risco está relacionado com sua gravidade, com a distribuição do impacto, como por exemplo: quantos usuários serão afetados, quanto do projeto será afetado e por quanto tempo este impacto será sentido [79].

Nenhuma equipe de software tem recursos para resolver todos os riscos possíveis com o mesmo rigor. Por isso, a priorização dos riscos é importante, para que se aloque recursos onde os riscos terão maior impacto [79]. O impacto do risco, poderá ser avaliado conforme critérios abaixo:

- Muito Baixo: falha ao atingir o requisito não traria impacto significativo ao projeto;
- Baixo: falha ao atingir o requisito traria impacto aceitável ao projeto;
- Moderado: falha ao atingir o requisito traria prejuízo moderado ao projeto;
- Alto: falha ao atingir o requisito traria grande prejuízo ao projeto;
- Muito Alto: falha ao atingir o requisito impactaria na continuidade do projeto;

Para um grande projeto, de 30 a 40 riscos em média podem ser identificados [79]. Se para cada um deles forem identificados de 3 a sete passos da gestão de riscos, esta gestão pode se tornar um projeto em si mesma. Por isso pode-se adaptar a regra 80-20 de Pareto [79]. A experiência indica que 80% dos riscos gerais de um projeto podem ser responsáveis por apenas 20% dos riscos identificados. Durante a atividade da análise de riscos, ela ajudará a descobrir quais dos riscos estão incluídos neste 20% críticos (os riscos mais altos do projeto).

Conforme o Quadro 5.2, a avaliação dos riscos se dará por meio do nível de risco, conforme abaixo:

- Muito Baixo: O risco deve ser apenas registrado. Risco aceito;
- Baixo: O risco deve ser registrado e apenas monitorado eventualmente;
- Moderado: O risco deve ser registrado, tratado a médio prazo e monitorado provisoriamente;
- Alto: O risco deve ser registrado, tratado a curto prazo e monitorado constantemente e
- Muito Alto: O risco deve ser registrado, tratado imediatamente e monitorado frequentemente.

No *Scrum*, a análise e avaliação dos riscos é realizada a cada iteração durante a reunião de planejamento da *Sprint*. O Quadro 5.2 pode ser impresso num papel de tamanho 'A0' e afixado na sala de reunião de planejamento da *Sprint*. À medida que os riscos forem sendo identificados e analisados quanto à sua probabilidade e impacto, irão sendo colocados "*post-its*" no quadro até que seja possível visualmente se identificar quais riscos devem ser tratados ou não, de acordo com o seu nível de risco. A equipe deve propor ações de mitigação ou evitação prioritariamente, para os riscos mais críticos e referentes a iteração que irá iniciar.

5.7.3 Tratamento e Planejamento de Resposta aos Riscos

A CGSI deve indicar qual estratégia seguir: evitar, transferir, aceitar ou mitigar (reduzir ou minimizar), em relação aos riscos identificados e avaliados. O tratamento dos riscos envolve a tomada de decisão sobre uma ou mais opções para modificar o risco. Estas opções são descritas a seguir:

- Aceitar o Risco: é a retenção do risco por uma escolha consciente;
- Mitigar o Risco: ocorre por meio da remoção da fonte de risco, ou da alteração da sua probabilidade ou ainda alteração das suas consequências;
- Evitar o Risco: é a ação de evitar o risco pela decisão de não iniciar ou descontinuar a atividade que dá origem ao risco;
- Transferir o Risco: é o compartilhamento do risco com outra parte ou partes (incluindo contratos e financiamento do risco).

Os tratamentos de riscos relativos às consequências negativas são muitas vezes referidos como ‘mitigação de riscos’, ‘eliminação de riscos’, ‘prevenção de riscos’ e ‘redução de riscos’. O tratamento de riscos pode criar novos riscos ou modificar riscos existentes.

A Figura 5.5 mostra de forma consolidada o detalhamento do subprocesso Tratar e Planejar Resposta aos Riscos do fluxograma exibido na Figura 5.2.

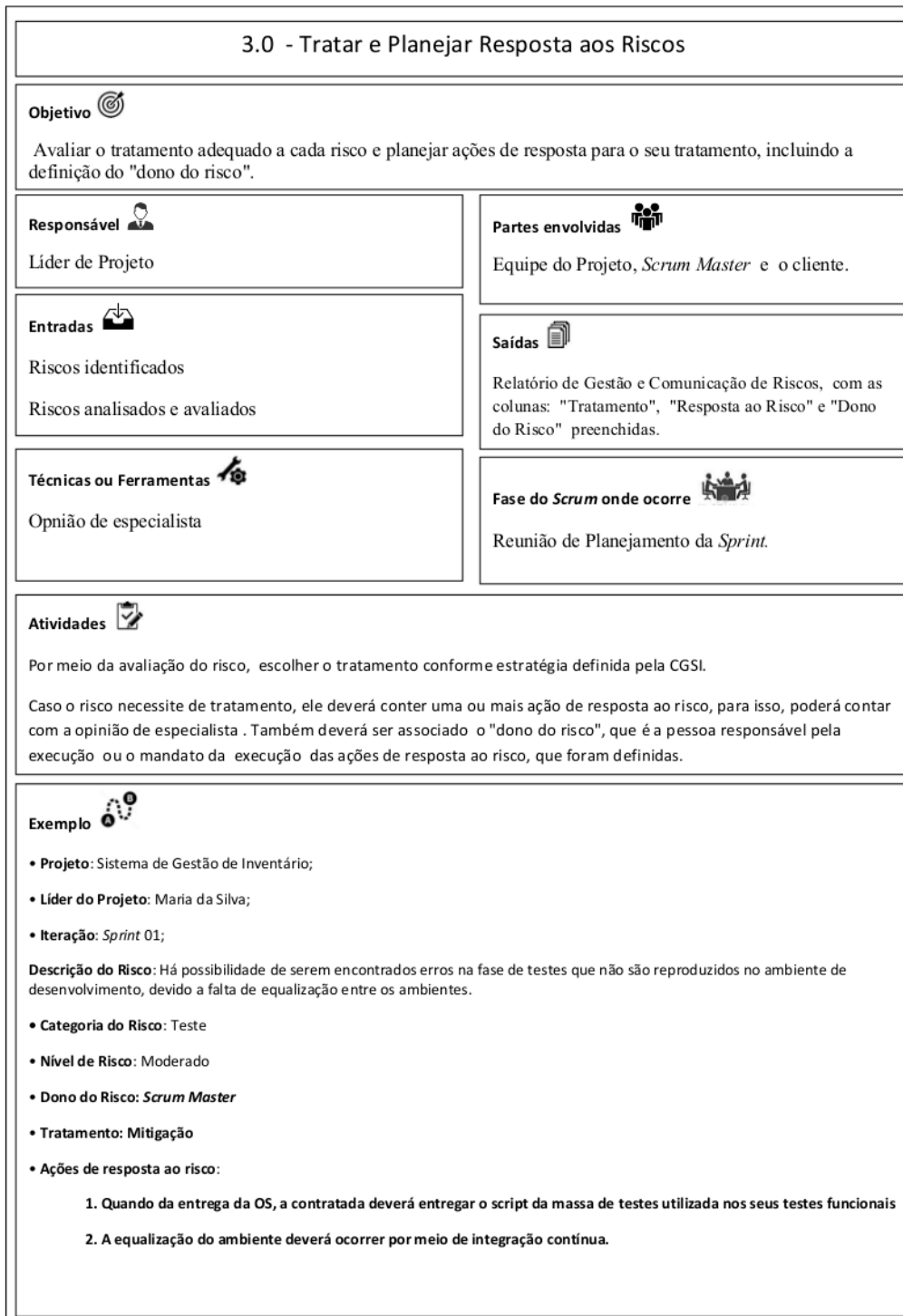


Figura 5.5: Subprocesso Tratar e Planejar Resposta aos Riscos. Fonte: Elaboração própria.

O planejamento de resposta aos riscos envolve a determinação dos riscos a serem gerenciados. Planos de ação para os riscos que estão sob controle e planos de contingência

para os que estão além das capacidades de mitigação. As definições das ações podem ser tanto ações de evitação, mitigação quanto de transferência dos riscos [67]. Os riscos mais críticos devem ter suas ações planejadas inicialmente, enquanto os demais devem ter o custo de mitigação comparados ao seu impacto para avaliação de custo e benefício [67]. Esta é uma parte crucial do processo de gestão de riscos, entender quais riscos devem ser aceitos e quais devem ser tratados, para que a gestão de riscos não se torne demasiadamente dispendiosa quanto ao tempo e ao custo de sua realização. Geralmente para descrever estas ações de resposta aos riscos, dependendo da categoria do risco, é necessário contar com a opinião de especialista, ou seja, um profissional que trabalhe na área relacionada à fonte do risco, para que as ações de tratamento descritas por ele possam ser mais efetivas.

5.7.4 Monitoramento do Risco

O monitoramento de risco é uma atividade de acompanhamento de projeto com três objetivos primários:

1. Avaliar se os riscos previstos vão ocorrer de fato;
2. Assegurar que as etapas de mitigação ao risco estejam sendo aplicadas adequadamente e
3. Coletar informações que possam ser usadas para futuras análises de riscos.

As seguintes perguntas ajudarão a realizar um eficiente monitoramento dos riscos:

- Como saber se o risco está se materializando ou já se materializou?
- Como detectá-lo?

A Figura 5.6 mostra de forma consolidada o detalhamento do subprocesso Monitorar Riscos do fluxograma exibido na Figura 5.2.

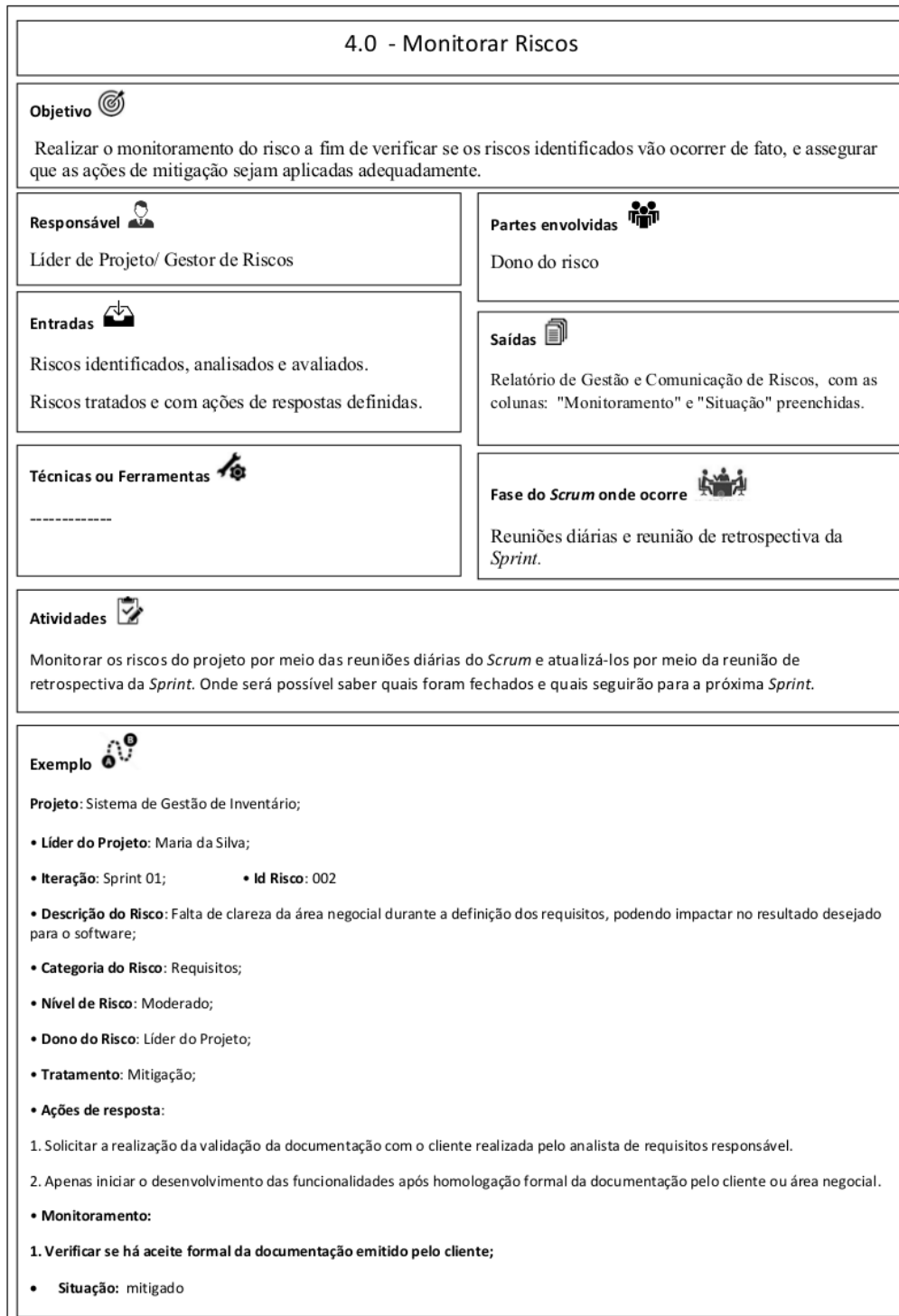


Figura 5.6: Subprocesso Monitorar Riscos. Fonte: Elaboração própria.

Para monitorar qualquer evento, é necessário um acompanhamento, no *Scrum*, este monitoramento deve ser realizado durante as reuniões diárias.

As ações de resposta ao risco podem ser utilizadas como auxiliares na criação da detecção do risco. Para responder às perguntas acima, será utilizado o exemplo abaixo, considere o risco seguinte:

- Risco: A falta de homologação da documentação dos requisitos do projeto, por parte do cliente, impede que a *Sprint* seja desenvolvida.
- Ação de Mitigação: Obter o aceite do cliente, para prosseguir a execução da *Sprint*.

Neste exemplo, o mecanismo usado para detecção é o aceite ou a homologação do cliente. O monitoramento é realizado nas reuniões diárias, onde o líder de projetos verifica junto ao analista de requisitos se ele já conseguiu realizar a validação do requisito com o cliente. Esta ação garante que o cliente homologue estando ciente do que leu e entendeu o documento, por isso a presença do analista de requisitos se faz necessária. Já a detecção pode ser realizada por meio da verificação do aceite formal do cliente. Se foi emitido este aceite, a mitigação do risco foi realizada. Caso não tenha sido emitido o aceite pelo cliente, o risco se materializou e, portanto, é necessária uma ação de contingência.

Usando ainda este exemplo hipotético, para a contingência deste risco, poderia se tomar a ação a seguir:

- Ação de Contingência: Suspensão temporária do projeto até que seja obtido o aceite referente a documentação da primeira entrega.

Esta ação de contingência evita gastos públicos no desenvolvimento de solução que pode não estar em consonância com a necessidade do requisitante, uma vez que o mesmo não emitiu o aceite da documentação.

5.7.5 Comunicação dos Riscos

A comunicação das informações produzidas deve atingir todos os níveis por meio de canais claros e abertos que permitam que a informação flua em todos os sentidos. O processo ‘Comunicar Riscos’, exibido na Figura 5.7, discorre sobre as formas de comunicação ágil.

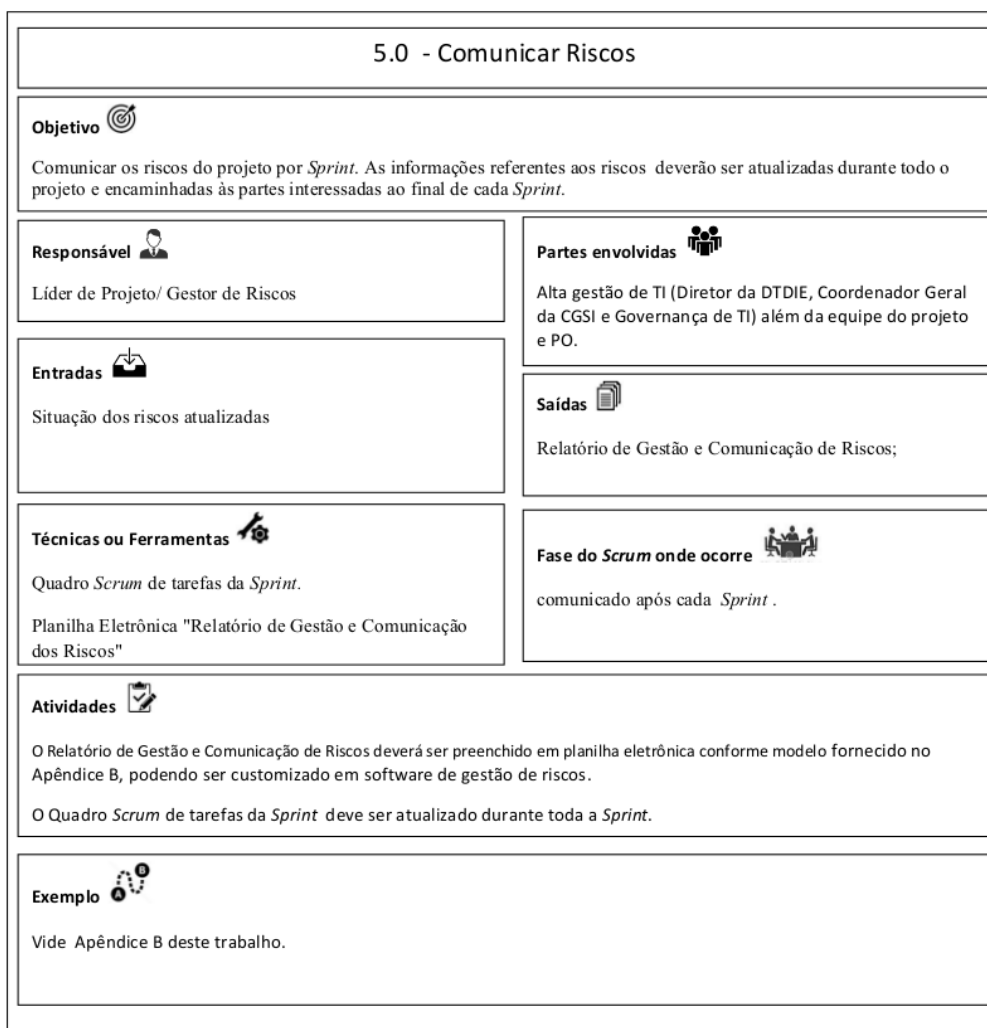


Figura 5.7: Subprocesso Comunicar Riscos. Fonte: Elaboração própria.

Para que não haja assimetria de informações dentro da equipe do projeto e para que os riscos sejam mais facilmente monitorados, durante as reuniões diárias, foi proposta a adição da coluna ‘riscos’ no quadro *Scrum*, exibido no exemplo da Figura 5.8.

Conforme o pilar do *Scrum* que diz respeito à transparência e baseando-se em um dos seus artefatos utilizados, com este quadro de tarefas da *Sprint* também é possível manter os riscos visíveis a todas as partes interessadas do projeto.

O *Scrum* foca na entrega de software funcional para o cliente. Esta é a medida primordial de progresso de seus projetos. Apesar da documentação ser importante, o *Scrum* é baseado no manifesto ágil que preza por leveza, agilidade. Assim, a documentação proposta nesta metodologia necessita ter apenas o detalhamento necessário ao entendimento das partes interessadas e cobrir na íntegra o processo da gestão de riscos, levando em consideração também o grau de maturidade em gestão de riscos da coordenação, a adaptação

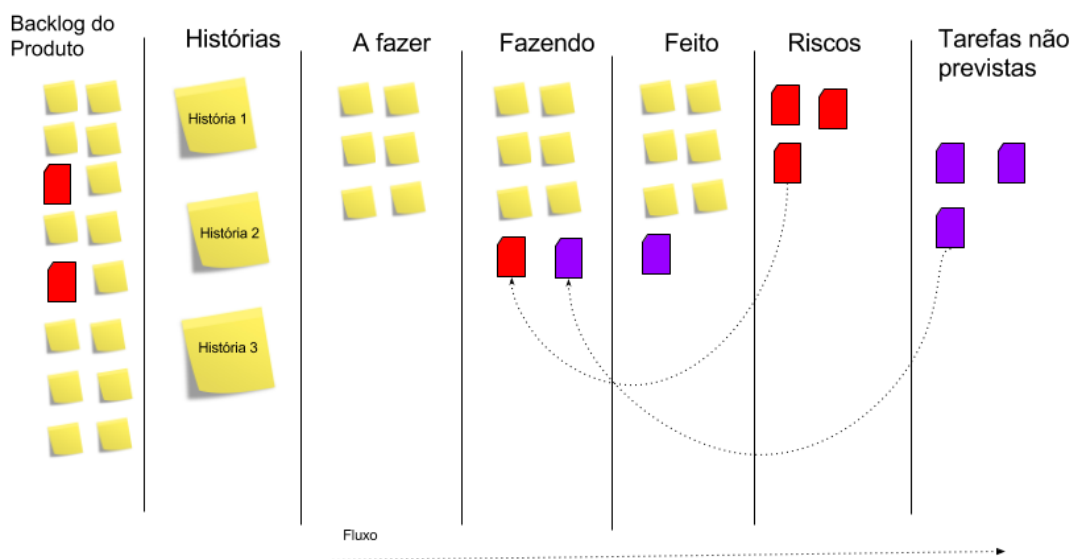


Figura 5.8: Quadro *Scrum* com controle dos Riscos da *Sprint*. Fonte: Elaboração própria.

ao processo e o princípio da economicidade, que discorre sobre a promoção de resultados esperados com o menor custo possível na prestação do serviço ou no trato com os bens públicos.

É importante que seja definida a matriz de comunicação para que níveis de acesso a determinadas informações possam ser mapeados. O dono do risco deverá conhecer a ação de resposta sob sua responsabilidade. Definições de acesso às informações dos riscos dos projetos por parte dos clientes ou demais diretorias devem ser acordadas com a alta administração e governança de TI [2]. E ainda, há necessidade de definição de quais informações são sigilosas, quais atendem às necessidades de *compliance* e qual o período de retenção dos registros.

Elaboração do *Template* do Relatório de Gestão e Comunicação dos Riscos

O Relatório de Gestão e Comunicação de Riscos, que consta no Apêndice 'B', é mantido em planilha eletrônica, onde, a partir da lista de riscos do projeto, é realizada a análise e avaliação dos riscos que, por meio do nível de risco, será sinalizada a possibilidade de tratamento (aceitar, evitar, mitigar ou transferir). Da mesma forma, também são definidas ações de resposta ao risco por meio da planilha e podem indicar a exposição aos riscos durante as iterações do projeto. Enfim, é um relatório simples que consolida todas as fases da gestão de riscos e que pode ser facilmente preenchido durante uma reunião com a equipe do projeto, sem demandar um grande esforço de manutenção como os planos de gerenciamento de riscos ao estilo de gerenciamento tradicional de projetos.

É necessário que o progresso em relação ao plano de gerenciamento de riscos seja reportado para a alta direção periodicamente, ao final de cada *Sprint*, variando para mais ou para menos, conforme o tamanho da *Sprint* [2].

5.7.6 Análise Crítica

A análise crítica, conforme Figura 5.9, deve ser realizada de forma objetiva e imparcial, incluindo um exame de toda a estrutura da gestão de riscos, seus processos e as possíveis mudanças no ambiente.










6.0 - Análise Crítica	
<p>Objetivo </p> <p>A análise crítica deve ser realizada incluindo o exame da estrutura da gestão de riscos, seus processos, os próprios riscos e as possíveis mudanças no ambiente. Também serve para identificar lições aprendidas e oportunidades de melhoria.</p>	
<p>Responsável </p> <p>Auditor</p>	<p>Partes envolvidas </p> <p>Equipe do Projeto, <i>Scrum Master</i> e o cliente.</p>
<p>Entradas </p> <p>Resultado do Processo de Monitoramento dos Riscos</p>	<p>Saídas </p> <p>Oportunidades de Melhoria Lições Aprendidas</p>
<p>Técnicas ou Ferramentas </p> <p>-----</p>	<p>Fase do <i>Scrum</i> onde ocorre </p> <p>Reunião de Retrospectiva da Sprint</p>
<p>Atividades </p> <p>Atividade realizada para determinar a adequação, suficiência e eficácia do assunto em questão para atingir os objetivos estabelecidos.</p> <p>A análise crítica pode ser aplicada à estrutura da gestão de riscos, ao processo de gestão de riscos, ao risco ou ao controle.</p>	
<p>Exemplo </p> <p>Na reunião de retrospectiva, lições aprendidas poderão ser capturadas com a equipe e com ela oportunidades de melhorias no processo da gestão dos riscos, irão surgir.</p>	

Figura 5.9: Subprocesso Análise Crítica. Fonte: Elaboração própria.

A análise crítica também serve para identificar lições aprendidas e oportunidades de melhoria. As questões significativas resultantes devem ser reportadas para aqueles que são responsabilizáveis [2]. Convém que as análises críticas sejam programadas e que os auditores foquem na melhoria do sistema e no tratamento das causas dos problemas. Também convém que os auditores sejam independentes da atividade a ser analisada, tanto quanto possível, para que possam atuar de forma livre de viés e conflitos de interesse, mantendo a objetividade durante o processo de análise crítica, para assegurar que as constatações e conclusões sejam baseadas apenas em evidências.

Os executores de fato do processo, precisam também analisar criticamente, de forma regular as suas experiências, saídas e resultados para identificar oportunidades de melhoria.

5.7.7 Melhoria Contínua

Convém que o processo da gestão de riscos seja analisado criticamente avaliando a concepção e se a sua implementação está de fato agregando o valor pretendido à organização. Conforme Figura 5.10, caso estes resultados mostrem pontos de melhoria, convém que sejam rotineiramente monitorados até que sejam implementados [2].

Há alguns gatilhos para a melhoria contínua. São eles:

1. Resultado de monitoramento de rotina, ou análise crítica que demonstre oportunidades de melhoria;
2. Novos conhecimentos se tornaram disponíveis;
3. Grandes mudanças no contexto externo e interno da organização;

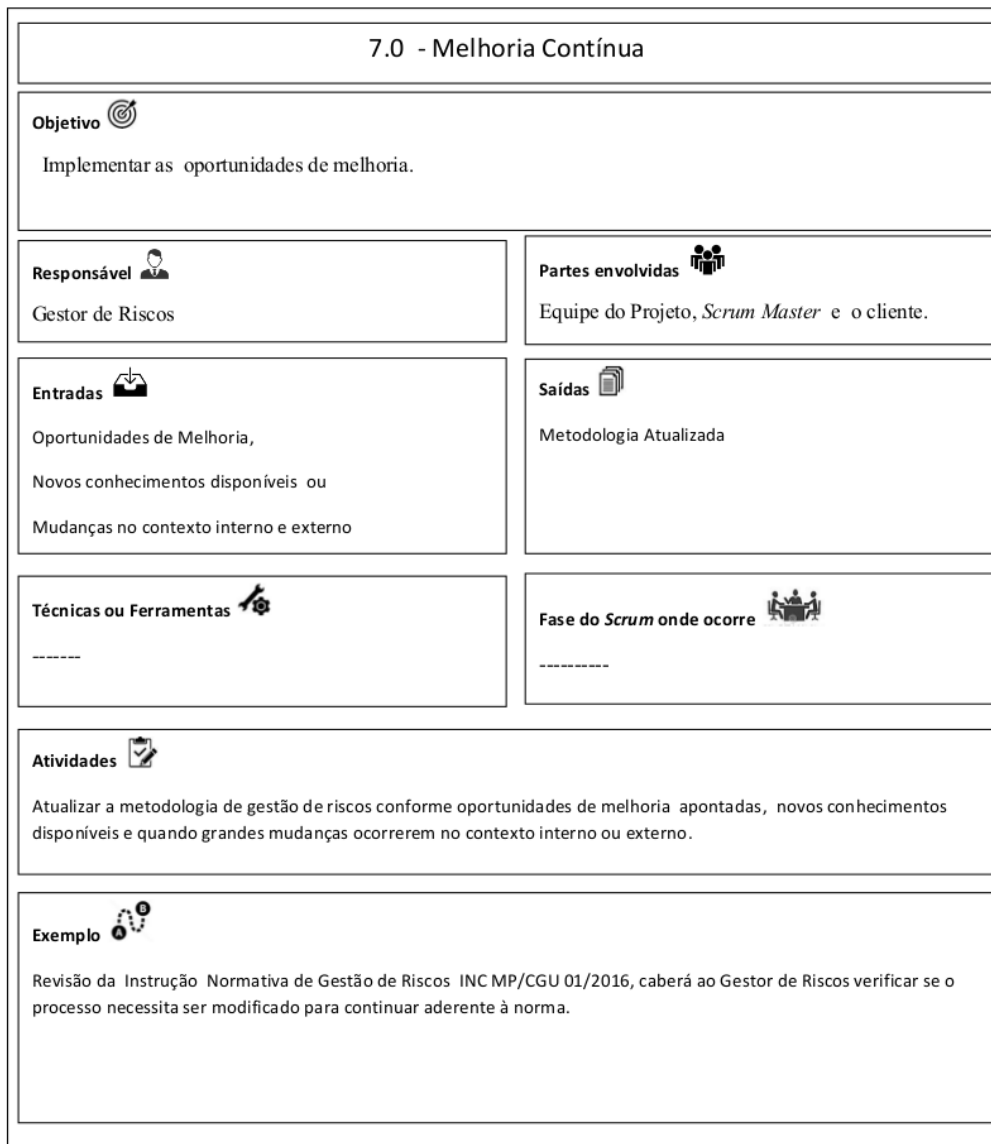


Figura 5.10: Subprocesso Melhoria Contínua. Fonte: Elaboração própria.

Esta metodologia foi elaborada para atender ao contexto atual, devendo ser submetida à análise crítica após a sua adoção e havendo oportunidades de melhoria as mesmas deverão ser implementadas. As questões significativas resultantes devem ser reportadas para aqueles que são responsabilizáveis [2].

O modelo de gestão de riscos proposto teve por objetivo mostrar como a gestão de riscos pode ser realizada dentro das fases do *Scrum* e quais as técnicas e ferramentas que podem ser utilizadas para uma gestão de riscos que não comprometa a agilidade preconizada na filosofia ágil.

Capítulo 6

Conclusão

O estudo bibliográfico revelou que a implementação de uma gestão de risco eficiente é importante na redução de perdas e no sucesso dos projetos de software. Os riscos encontram-se presentes em todos os projetos de desenvolvimento de software, sendo uma necessidade a adoção de uma gestão estruturada e eficiente para responder aos impactos que poderão causar no projeto. Vários são os projetos que face a uma incorreta gestão dos riscos são interrompidos e acabam por causar perdas significativas para a organização.

A metodologia ágil consiste em diversas técnicas que podem ser utilizadas para efeitos de desenvolvimento de um produto. Entre todas, o *Scrum* é um *framework* amplamente aceito, focado na gestão descentralizada, processo empírico de controle e alta adaptabilidade. Porém, as abordagens ágeis por si só não são uma medida de minimização dos riscos. Assim, há a necessidade da realização do processo de gestão de riscos explicitamente nos projetos de software. Constatou-se que existem poucos trabalhos relacionados à aplicação da gestão de riscos em metodologias ágeis e que muitos modelos de integração são flexíveis demais a ponto de não terem sido sugeridas técnicas que possam ser utilizadas em cada etapa da gestão de riscos.

Apesar disto, o trabalho propôs uma metodologia de gestão de riscos que pretende ser alinhada ao *Scrum*, às normas: NBR ISO/IEC 31000, manifesto ágil, MGR-SISP e a INC MP/CGU nº 01/2016. Esta metodologia foi desenvolvida e moldada à realidade da coordenação, buscando a simplicidade, leveza e agilidade preconizada pelo manifesto ágil e levando em considerações fatores culturais como a maturidade na APF no processo de gestão de riscos, observada no relatório do TCU (2014) [93]. Este relatório apontou que apenas 9% das organizações públicas implementavam integralmente a gestão de riscos nos seus processos de gestão, denotando que este processo, apesar de sua importância, ainda precisa ser mais discutido e analisado para que seja efetivamente amadurecido, customizado e que venha a ser executado nas organizações.

A metodologia proposta é uma fusão do processo de gestão de riscos proposto pela

ISO/IEC 31000 com o *Scrum*. Ela explica como as atividades da gestão de riscos poderão ser realizadas em cada momento do *Scrum*, sugerindo as ferramentas e técnicas mais utilizadas em modelos internacionais de integração entre métodos ágeis e gestão de riscos. Para a identificação dos riscos também foi elaborada uma lista de riscos comuns nos projetos de software da coordenação estudada e ainda desenvolvido um *template* de relatório de gestão e comunicação dos riscos.

De forma estruturada e integrada, a metodologia nasce com intuito de ser simples, pois entende que a gestão deverá ser mais uma atividade corriqueira do projeto e não se tornar complexa, correndo risco de não se ter recursos para sua execução ou ainda consumir muito tempo do projeto.

Espera-se que a adoção desta metodologia possa agregar valor ao resultado dos projetos, para que os riscos comecem a ser identificados com maior facilidade por meio da lista de riscos, e possam ser devidamente tratados quanto ao seu impacto, propondo ações de mitigação e sendo monitorados quanto a sua eficácia. A intenção é que os riscos sejam identificados e tratados o mais cedo possível no processo de desenvolvimento de software para minimizar atrasos e danos a imagem e reputação da organização. Esta metodologia vai ao encontro de novas tendências da legislação brasileira quanto à adoção da gestão de riscos nos órgãos da APF, a fim de assegurar que estes atinjam os seus objetivos finalísticos.

A melhoria dos processos internos ocorrem naturalmente quando da adoção da gestão de riscos, pois muitas ações de resposta aos riscos, demandam melhorias em seus processos internos [2]. Por isso, há grandes chances da adoção da gestão de riscos, aumentar a taxa de sucesso dos projetos e o atingimento das metas estratégicas do PDTI.

Esta metodologia pode contribuir com outras organizações pois diferentemente da MGR-SISP voltada completamente para segurança da informação e comunicação, a metodologia proposta está voltada aos projetos de desenvolvimento ágil de software e tem potencial para ser customizada em outras organizações que também tenham adotado o *Scrum*. Outra contribuição seria estar em alinhamento com recomendação de órgãos de controle como TCU e CGU. Por meio da sua adoção seria possível dar transparência às partes interessadas sobre os riscos, seus responsáveis, suas ações de respostas e o monitoramento.

O trabalho realizado apresentou limitações como: ausência de validação da metodologia por especialistas e a aplicação da metodologia proposta. Por este motivo, com relação a trabalhos futuros, sugere-se a validação da metodologia por especialistas da área de gestão de riscos e ainda sua aplicação nos projetos da CGSI.

Referências

- [1] ABNT NBR ISO/IEC 31000. Gestão de riscos — princípios e diretrizes, Primeira Edição; 2009. 3, 31, 38, 91, 96
- [2] ABNT ISO/TR 31004. Gestão de riscos - guia para a implementação da abnt nbr iso 31000, Primeira Edição; 2015. 1, 4, 34, 91, 116, 117, 118, 119, 121
- [3] ABNT NBR ISO/IEC 31010. Gestão de riscos - técnicas para o processo de avaliação de riscos, Primeira Edição; 2012. 0, 31, 32, 33, 84, 91
- [4] Cultura Agil. Gerenciamento de riscos em projetos ágeis. <http://www.culturaagil.com.br/gerenciamento-de-riscos-em-projetos-ageis/>, 22/05/2016. xi, 23, 24, 60
- [5] A. Agrawala, M. A. Atiqb, e L. S. Mauryac. Current study on the limitations of agile methods in industry using secure google forms. *Procedia Computer Science*, 78:291–297, 2016. 35
- [6] A. Albadarneh, I. Albadarneh, e A. Qusef. Risk management in agile software development: a comparative study. *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, 2015. 0, 36
- [7] A. Albadarneh, I. Albadarneh, e A. Qusef. Risk management in agile software development: a comparative study. *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6, 2015. 39
- [8] H. Andrat e S. Jaswal. An alternative approach for risk assessment in scrum. *Conference on Computing and Network Communications*, 41:16–19, 2015. xi, 0, 41, 42
- [9] H. Andrat e S. Jaswal. An alternative approach for risk assessment in scrum. *International Conference on Computing and Network Communications*, 54:535 – 539, 2015. 39
- [10] R. P. Anita e N. Chauhan. A mapping model for transforming traditional software development methods to agile methodology. *International Journal of Software Engineering Applications (IJSEA)*, 4(4), 2013. 43, 44
- [11] P. L Bannerman. Risk and risk management in software projects: A reassessment. *The Journal of Systems and Software*, 81:2118–2133, 2008. 4, 37, 38

- [12] M. O. Barros, C. M. L. Werner, e G. H. Travassos. Supporting risks in software project management. *The Journal of Systems and Software*, 70:21–35, 2004. 27
- [13] J. M. Bass. Artefacts and agile method tailoring in large-scale offshore software development programmes. *Information and Software Technology*, 75:1–16, 2016. 49
- [14] K. Beck. Agile manifesto. <http://www.manifestoagil.com.br/>, 26/05/2016. 13, 14, 15
- [15] P. H. S. Bermejo, A. L. Zambaldea, A. O. Tonellib, S. A. Souza, L. A. Zuppoa, e P. L. Rosa. Agile principles and achievement of success in software development: A quantitative study in brazilian organizations. *CENTERIS 2014 - Conference on ENTERprise Information Systems / ProjMAN 2014 - International Conference on Project MANagement / HCIST 2014 - International Conference on Health and Social Care Information Systems and Technologies*, 16, 2014. 15
- [16] E. Bjarnason, K. Wnuk, e B. Regnell. Are you biting off more than you can chew? a case study on causes and effects of overscoping in large-scale software engineering. *Information and Software Technology*, 54:1107–1124, 2012. 39
- [17] B. W. Boehm. A spiral model of software development and enhancement. *IEEE Computer*, 21:61–72, 1988. 38
- [18] B. W. Boehm. Software risk management: Principles and practices. *IEEE Software*, 8:32–41, 1991. 11, 37, 38
- [19] Z. Bougroun, A. Zeaaraoui, e T. Bouchentouf. the projection of the specific practices of the third level of cmmi in agile methods: Scrum, xp e kanban. *33rd Int. Spring Seminar on Electronics Technology*, pages 174–179, 2014. 39
- [20] K.M. Bumbarly. Using velocity, acceleration, and jerk to manage agile schedule risk. *International Conference on Information Systems Engineering*, pages 73–80, 2016. 39
- [21] A. S. Campanelli e F. S. Parreiras. Agile methods tailoring - a systematic literature review. *The Journal of Systems and Software*, 10:85–100, 2015. 15
- [22] A.L. Cervo, P.A. Bervian, e R. Da Silva. *Metodologia científica*. Pearson Education, 2009. 57, 58
- [23] P. Clarke e R.V. OConnor. The situational factors that affect the software development process: Towards a comprehensive reference framework. *Information and Software Technology*, 54:433–447, 2012. 39
- [24] P. Clutterbuck. A risk management investigation of sme adoption of agile method information system development. *3rd European Conference on Information Management and Evaluation Local: IT Univ, Gothenburg, SWEDEN*, pages 107–115, 2009. 39
- [25] K. Conboy, S. Coyle, X. Wang, e et al. People over process: key challenges in agile development. *IEEE Software*, pages 48–57, 2011. 26

- [26] H. R. Costa, M. O. Barros, e G. H. Travassos. Evaluating software project portfolio risks. *The Journal of Systems and Software*, 80:16–31, 2007. 27
- [27] M. A. F. Costa e M. F. B. Costa. *Metodologia da pesquisa- conceitos e técnicas*. Rio de Janeiro: Editora Interciência, 2001. 58
- [28] R. Cunha, C.S. Perreira, e J.A. Pinto. Agile software project: Proposal of a model to manage risks. *8th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–5, 2013. xi, 2, 39, 44, 45, 73, 103
- [29] A. Danait. Agile offshore techniques – a case study. *Proc. Agile Development Conf.*, page 214–217, 2005. 26
- [30] Governança de TI INEP. Plano diretor de tecnologia da informação 2016-2019, 14/03/2016. 0, 69, 70
- [31] Governança de TI INEP. Catalogo de sistemas. <http://portal.inep.gov.br>, 16/05/2016. xi, xii, 75, 76, 77, 78, 79
- [32] T. Dingsoyr, G. k. Hanssen, T. Dyba, G. Anker, e J. O. Nygaard. Developing software with scrum in a small cross-organizational project. *Software Process Improvement*, 4257:5–15, 2006. 16
- [33] T. Dingsoyr, S. Nerur, V. Balijepally, e N.B. Moe. A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software*, 85:1213–1221, 2012. 15
- [34] V. Dorca, S. Popescu, R. Munteanu, A. Chioreanu, e C. Peleskei. Agile approach with kanban in information security risk management. *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, pages 1–6, 2016. 39
- [35] A. J. Dorofee, J. A. Walker, C. J. Alberts, R. P. Higuera, R. L. Murphy, e R. C. Williams. *Continuous Risk Management Guidebook*. Carnegie Mellon University, SEI, 1996. 38
- [36] T. Dyba e T. Dingsoyr. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50:833–859, 2008. 18
- [37] A.A. Fernandes e V.F. Abreu. *Implantando a governança de TI: da estratégia à gestão de processos e serviços*. BRASPORT, 2012. 29, 30
- [38] V. Franqueira, Z. Bakalova, T. Tun, e M. Daneva. Towards agile security risk management in re and beyond. *33rd Int. Spring Seminar on Electronics Technology*, pages 33–36, 2011. 39
- [39] Gartner. Ti bimodal. <http://www.gartner.com/smarterwithgartner/busting-bimodal-myths/>, 29/07/2017. 24
- [40] S. Ghobadi e L. Mathiassen. Perceived barriers to effective knowledge sharing in agile software team. *Information Systems Journal*, 26:95–125, 2016. 39

- [41] B. Gold e C. Vassell. Using risk management to balance agile methods: A study of the scrum process. *IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 49–54, 2015. 39
- [42] The Standish Group. Chaos manifesto 2013. 37
- [43] T. Hayata e J. Han. A hybrid model for it project with scrum. *IEEE Int. Conf. Service Operations, Logistics, and Informatics*, pages 285–290, 2011. 26
- [44] S. J. Huang e W. M. Han. Exploring the relationship between software project duration and risk exposure: A cluster analysis. *Information Management*, 45:175–182, 2008. 37
- [45] INEP. Decreto n° 6.317,de 20 de dezembro de 2007. <http://www.planalto.gov.br/ccivil03/ato2007-2010/2007/Decreto/D6317.htm>, 14/03/2016.2
- [46] INEP. Portal do inep. <http://portal.inep.gov.br>, 16/06/2015. vi, vii, xi, 3, 62, 63, 65, 66, 67, 68
- [47] INEP. Metodologia de gestão e desenvolvimento de sistemas mgds 2.1, 2012. 3
- [48] A. Kaczorowska. Traditional and agile project management in public sector and ic. *Proceedings of the Federated Conference on Computer Science and Information Systems*, page 1521–1531, 2015. 39
- [49] M. Keil, P. E. Cule, K. Lyytinen, e R.C Schmidt. A framework for identifying software project risks. *COMMUNICATIONS OF THE ACM*, 41(11), 1998. 38
- [50] A. Khatavakhotan, N. Hashemitaba, e O. Siew Hock. From identification to budget allocation: A novel it risk management model for iterative agile projects. *International Conference on Computer Design and Engineering (ICCDE 2011) Local: Kuala Lumpur, MALAYSIA*, 2011. 39, 73, 103
- [51] S.K Khatri, K. Bahri, e P. Johri. Best practices for managing risk in adaptive agile process. *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization*, 2014. xi, 39, 45, 46
- [52] Y. H. Kwaka e J. Stoddard. Project risk management: lessons learned from software development environment. *Technovation*, 24:915–920, 2004. 37
- [53] S. T. Lai. Maintainability enhancement procedure for reducing agile software development risk. *International Journal of Software Engineering Applications (IJSEA)*, 6(4), 2015. 34, 38
- [54] M. Larusdottira, J. Gulliksenb, e A. Cajanderc. Alicense to kill–improving ucscd in agile development. *The JournalofSystemsandSoftware*, 000:1–9, 2016. 35
- [55] Y. Leau, W. Loo, W. Tham, e S. Tan. Software development life cycle agile vs traditional approaches. *Int. Conf. Information and Network Technology, Singapore*, 2012. 26

- [56] J. Livari e N. Livari. The relationship between organizational culture and the deployment of agile methods. *Information and Software Technology*, 53:509–520, 2011. 35
- [57] L. L. Lobato, P. Neto, e I. Machado. A study on risk management for software engineering. *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, pages 47–51, 2012. 39
- [58] L. L. Lobato, P. Neto, e I. Machado. Agile development as a change management approach in software projects: Applied case study. *2nd International Conference on Information Management (ICIM)*, 54:100–104, 2016. 39
- [59] M. A. Marconi e E. M. Lakatos. *Fundamentos de Metodologia Científica*. São Paulo: Editora Atlas, 2010. 57
- [60] A. Martakis e M. Daneva. Handling requirements dependencies in agile projects: A focus group with agile software development practitioners. *IEEE 7th International Conference on Research Challenges in Information Science (RCIS)*, pages 1–11, 2013. 39
- [61] DESENVOLVIMENTO E GESTÃO SECRETARIA DE TECNOLOGIA DA INFORMAÇÃO MINISTÉRIO DO PLANEJAMENTO. Metodologia de gestão de riscos de segurança da informação e comunicações do sistema de administração de recursos da tecnologia da informação – sisp do poder executivo federal (mgr-sisp) v2.0. <https://www.governoeletronico.gov.br/documentos-e-arquivos/MGR-SISP-V260816.pdf>, 16/11/2016. xi, 0, 54, 55, 108
- [62] S. Misra, V. Kumar, e U. Kumar. Identifying some critical changes required in adopting agile practices in traditional software development projects. *Int. J. Quality Reliab. Manage*, page 451–474, 2010. 26
- [63] M. Molhanec. Agile project management framework. *33rd Int. Spring Seminar on Electronics Technology*, pages 525 – 530, 2010. 39, 73, 103
- [64] MP/CGU. Instrução normativa conjunta mp/cgu nº 01 de 10/05/2016, 2016. vi, vii, 2, 4, 53, 92, 95, 96
- [65] V. Mudumba e O. Lee. A new perspective on gdsd risk management. *International Conference on Global Software Engineering*, pages 219–227, 2010. 39
- [66] V. Mudumba e O. K. Lee. A new perspective on gdsd risk management: Agile risk management. *Global Software Engineering (ICGSE)*, 219:227, 2010. 49, 73, 104
- [67] M. Nogueira. *Engenharia de Software - Um framework para a Gestão de Riscos em Projetos de Software*. Ciência Moderna, 2009. 112
- [68] J Nyfjord. Towards integrating agile development and risk management. tese de doutorado, Universidade de Estocolmo, 2008. 3, 49

- [69] J. Nyfjord e M. Kajko-Mattsson. Outlining a model integrating risk management and agile software development. *34th Euromicro Conference Software Engineering and Advanced Applications*, pages 476 – 483, 2008. 0, 39, 100
- [70] J. Oliveira, V. Margarida, M. Nogueira, P. Ribeiro, e R. J. Machado. Is scrum useful to mitigate project’s risks in real business contexts? *Computational Science and Its Applications – ICCSA 2016*, 9790:422–437, 2016. 39
- [71] X. Peng. Coordination in large agile projects. *Rev. Bus. Inf. Syst.*, pages 29–44, 2009. 26
- [72] M. Perkusich, G. Soares, H. Almeida, e A. Perkusich. A procedure to detect problems of processes in software development projects using bayesian networks. *Expert Systems With Applications*, 42:437–450, 2015. 39
- [73] A. PHAM e P. V. PHAM. *Scrum em ação: Gerenciamento e Desenvolvimento Ágil de Projetos de Software*. São Paulo: Novatec, Reading, Mass., 2014. 16, 24
- [74] M. J. Pikkarainen, O. Haikara, P. Salo, e J. S. Abrahamsson. The impact of agile practices on communication in software development. *Empirical Software Engineering*, 13:303–337, 2008. 18
- [75] PMI. *Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK) Quinta Edição*. PMI Project Management Institute, 2013. 25, 27
- [76] PMI e IEEE Computer Society. *Software Extension to the PMBOK Guide fifth Edition*. IEEE Computer Society and PMI Global Standard, 2013. 25, 43, 97
- [77] E. R. Poort. Driving agile architecting with cost and risk. *THE PRAGMATIC ARCHITECT*, 27(2), 2010. 34, 35
- [78] PPCA. Portal do ppca - unb. <http://ppca.unb.br>, 02/12/2016. 1
- [79] S.R. Pressman e R. B. Maxin. *Engenharia de Software - UMA ABORDAGEM PROFISSIONAL*. bookman, 2016. xi, 1, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 20, 21, 22, 27, 28, 73, 103, 104, 106, 108, 109
- [80] L. Pries-Heje e J. Pries-Heje. Why scrum works: a case study from an agile distributed project in denmark and india. *Agile Conf.*, pages 20–28, 2011. 26
- [81] P.J. Rech. *Gerenciamento de Riscos em Projetos de Desenvolvimento de Software com Scrum*. Porto Alegre: Editora?, 2013. xi, 17
- [82] H. J. Rekha e R. Parvathi. Survey on software project risks and big data analytics. *Procedia Computer Science*, 50:295–300, 2015. 27
- [83] J. Ropponen e k. Lyytinen. Components of software development risk:how to address them?a project manager survey. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 26(2):98–112, 2000. 38
- [84] M. Schniederjans, A. Schniederjans, e D. Schniederjans. Outsourcing and insourcing in an international context. *M.E. Sharpe*, 2005. 26

- [85] K. Schwaber e J. Sutherland. Um guia definitivo para o scrum: As regras do jogo. <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>, 02/07/2015. 17, 18
- [86] L. Siddique e B. A. Hussein. Practical insight about risk management process in agile software projects in norway. *34th Euromicro Conference Software Engineering and Advanced Applications*, pages 1–4, 2014. 39, 73, 104
- [87] I. Sommerville. *Engenharia de Software, 8º edição*. Pearson, Addison Wesley, 2007. 8, 9, 10, 11, 12
- [88] S. Srikrishnan, B. Marath, e K.V. Pramod. Case study on risk management practice in large offshore-outsourced agile software projects. *IET Software*, 2014. 0, 26
- [89] S. Sundararajan, M. Bhasi, e P. K. Vijayaraghavan. Case study on risk management practice in large offshore-outsourced agile software projects. *Published by the IET*, 08:245–257, 2014. 39
- [90] J. Sutherland e K. Schwaber. Scrum and the perfect storm. <http://www.controlchaos.com/my-articles>, 26/05/2016. 16
- [91] H. S. Sverrisdottira, H. T. Ingasonb, e H. I. Jonassonc. The role of the product owner in scrum-comparison between theory and practices. *Sciences*, 119:257–267, 2014. 16, 18
- [92] H. Takeuchi e I. Nonaka. The new new product development game. *HARVARD BUSINESS REVIEW*, 64(01):137–146, 1986. 16
- [93] TCU. Avaliação da governança de tecnologia da informação na administração pública federal, tc n. 003.732/2014-2. vi, vii, 3, 4, 49, 50, 120
- [94] TCU. Conhecimento acerca da utilização de métodos ágeis nas contratações para desenvolvimento de software pela administração pública federal, tc 010.663/2013-4, 27/07/2016. 22, 23, 24, 50, 52, 61, 73, 74, 104
- [95] R. Turner e B. Boehm. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*, pages 30–39, 2015. 26, 104
- [96] E. Uy e N. Ioannou. Growing and sustaining an offshore scrum engagement. *Agile 2008 Conf.*, page 345–350, 2008. 26
- [97] Suprika V., Shrivastava, e U. Rathod. Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58:373–387, 2015. 35, 39, 49
- [98] VersionOne. State of agile report, the 10th. 19
- [99] VersionOne. State of agile survey, the 9th. <https://www.versionone.com/about/press-releases/versionone-releases-9th-annual-state-of-agile-survey-results/>, 20/04/2016. xi, 19

- [100] M. Wanderley, J. Menezes, C. Gusmão, e F. Lima. Proposal of risk management metrics for multiple project software development. *Procedia Computer Science*, 64:1001–1009, 2015. 37
- [101] S. Yacoub e H. Ammar. A methodology for architecture-level reliability risk analysis. *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 28(6):529–547, 2002. 38

Apêndice A

Lista/*checklist* de Riscos - CGSI

Checklist de riscos comuns aos Projetos de software da CGSI

Id	Categoria	Descrição	Fontes	Checklist
1	MODELAGEM	Falta de clareza dos gestores e atores envolvidos em cada etapa do processo.	TCU[92]	<input type="checkbox"/>
2	MODELAGEM	Indisponibilidade do Gestor para reuniões de modelagem do sistema.	TCU[92]	<input type="checkbox"/>
3	MODELAGEM	Indefinição ou inexistência de processos de trabalho interno na CGSI.	Fraquezas da SWOT	<input type="checkbox"/>
4	MODELAGEM	Indefinição de processo para manutenções em sistemas legados.	Fraquezas da SWOT	<input type="checkbox"/>
5	MODELAGEM	Falta de ferramentas, técnicas e padrões para subsidiar a execução dos processos.	Observação Participante	<input type="checkbox"/>
6	REQUISITOS	Incertezas das áreas de negócio durante a definição de requisitos.	TCU[92]; SWOT	<input type="checkbox"/>
7	REQUISITOS	Falta de levantamento de requisitos não funcionais, para a implementação e teste adequado do sistema.	Fraquezas da SWOT	<input type="checkbox"/>
8	REQUISITOS	Documentação ambígua ou incompleta.	TCU[92];[8]	<input type="checkbox"/>
9	REQUISITOS	Falta de rastreabilidade dos requisitos por meio de matriz de rastreabilidade	Observação Participante	<input type="checkbox"/>
10	REQUISITOS	Mudança na visão do sistema por mudança do Gestor da área de negócios.	TCU[92]	<input type="checkbox"/>
12	ANALISE E DESIGN	Utilização de novas tecnologias, sem a realização de prova de conceito .	Observação Participante	<input type="checkbox"/>
13	ANALISE E DESIGN	Mudança frequente dos frameworks gerando impacto nas aplicações.	Observação Participante	<input type="checkbox"/>
14	ANALISE E DESIGN	Dificuldade de integração com módulos ou outros sistemas.	Observação Participante	<input type="checkbox"/>
15	ANALISE E DESIGN	Dependência na utilização de serviço mantido por outros órgãos do governo - WEBSERVICES do MEC, ou mantidos pelo SERPRO e RFB por exemplo.	Observação Participante	<input type="checkbox"/>
16	ANALISE E DESIGN	Falta de conformidade da entrega com o guia de arquitetura adotado no Inep.	Observação Participante	<input type="checkbox"/>
17	TESTE	Falta de ferramentas para execução de testes automatizados	Observação Participante	<input type="checkbox"/>
18	IMPLANTACAO	Falta de equalização dos ambientes em relação a banco, código, variáveis de ambiente e configurações em geral como: ssi, cron e outras.	Fraquezas da SWOT	<input type="checkbox"/>
19	IMPLANTACAO	Entregas incompletas, faltando scripts, documentação e demais configurações necessárias para o ambiente que se deseja implantar.	TCU[92]	<input type="checkbox"/>
20	IMPLANTACAO	Indisponibilidade temporária da ferramenta Jenkins ou do repositório svn para a geração de build.	Observação Participante	<input type="checkbox"/>
21	IMPLANTACAO	Falta de definição de responsáveis pelo preenchimento das informações do documento de implantação.	TCU[92]	<input type="checkbox"/>
22	GERENC. CONF E MUDANÇA	Ações individuais de configuração e mudanças realizadas sem planejamento, comunicação ou avaliação de impacto sobre outros sistemas.	Fraquezas da SWOT	<input type="checkbox"/>
23	GESTÃO DE PROJETOS	Comunicação ineficiente na equipe (PO, Scrum Master , requisitos e analistas) ou entre demais áreas.	Fraquezas da SWOT	<input type="checkbox"/>
24	GESTÃO DE PROJETOS	Pouca autoridade do Gerente de Projeto com relação a priorização das atividades junto às demais áreas da TI.	Observação Participante	<input type="checkbox"/>
25	GESTÃO DE PROJETOS	Falta de um gerenciamento de riscos (identificação, tratamento e monitoramento) adequado.	[8]	<input type="checkbox"/>
26	GESTÃO DE PROJETOS	Falta de suporte ou treinamento em ferramenta para gerenciamento adequado de projetos.	TCU[92]; SWOT	<input type="checkbox"/>
27	GESTÃO DE PROJETOS	Falta de planejamento adequado do software a ser construído	TCU[92]	<input type="checkbox"/>
28	RECURSOS HUMANOS	Falta de expertise na utilização das metodologias ágeis, como o Scrum.	[8]	<input type="checkbox"/>
29	RECURSOS HUMANOS	Ineficácia no dimensionamento de equipes e recursos.	Fraquezas da SWOT	<input type="checkbox"/>
30	RECURSOS HUMANOS	Mudança de pessoa chave no projeto (cliente (PO), time de desenvolvimento, Scrum Master ou Gerente de Projeto).	[8]; TCU[92]	<input type="checkbox"/>
31	CONTRATADAS	Dificuldade de se estimar prazos realistas de entregas .	Observação Participante	<input type="checkbox"/>
32	CONTRATADAS	Dificuldade de comunicação entre Inep e FSW devido a trabalho executado remotamente.	Observação Participante	<input type="checkbox"/>
33	CONTRATADAS	Incapacidade das fábricas/contratadas em reter profissionais, causando alta rotatividade.	Observação Participante	<input type="checkbox"/>
34	CONTRATADAS	Inexecução dos contratos, causando interrupção dos projetos.	Observação Participante	<input type="checkbox"/>
35	REGULAMENTAÇÃO	Não atendimento de prazos legais: definidos em decretos, leis e portarias.	Observação Participante	<input type="checkbox"/>

Obs: Esta lista não contempla os riscos específicos de cada projeto. É um guia para identificação de riscos comuns que podem ocorrer nos projetos de software desta coordenação.

Apêndice B

Relatório de Gestão e Comunicação de Riscos

Relatório de Gestão e Comunicação de Riscos

Projeto: Projeto X

Data de Início: 19/01/2017

Líder de Projeto: Fulano de tal

Data da Atualização: 02/02/2017

Cód.	Nível de Risco	Descrição do risco	Probabilidade	Impacto	Categoria	Tratamento	Resposta ao Risco - Descrição	Dono do Risco	Monitoramento	Situação
1	2	Indisponibilidade temporária de ferramentas como svn e Jenkins	2-Baixa	1-Muito baixo	Técnico	Assumir	-----	-----	-----	-----
2	9	comunicação ineficiente entre equipe e LP	3-Média	3-Médio	Gestão do projeto	Mitigar	participação da equipe e Líder de Projetos nas Reuniões diárias, encaminhar emails a todos da equipe	Beltrano	diariamente durante as <i>daily scrums</i>	em mitigação
3	16	falta de ferramentas para execução de testes automatizados	4-Alta	4-Alto	Técnico	Prevenir	buscar ferramentas de testes automatizados	Fulano	verificar se já foi realizada a aquisição	ativo
4	25	baixa qualidade de testes realizados pela fabrica, causando reteste pelo Inep	5-Muito Alta	5-Muito Alto	Técnico	Transferir	solicitar evidencias de testes, plano de testes, definir criterios de aceitação no planejamento da	Fulano	ao receber a OS entregue no CCP, deverá ser verificado	fechado

Sprint nº	Identificados	Aceitos	Mitigados	Evitados	Transferidos	Fechados	Residuais
1	15	5	5	3	2	5	0
2	4	1	1	1	1	1	
3							
4							
5							

