Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise e Melhoria de Processo de Reengenharia de Software: Proposta de Instrumento para Gerenciamento de Riscos do Processo

Guilherme Mendonça de Moraes

Orientador
Prof. Dr. Edgard Costa Oliveira

Brasília
2021

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

MG956a

Mendonça de Moraes, Guilherme
Análise e Melhoria de Processo de Reengenharia de Software: Proposta de Instrumento para Gerenciamento de Riscos do Processo / Guilherme Mendonça de Moraes; orientador Edgard Costa Oliveira. -- Brasília, 2021.
124 p.

Dissertação (Mestrado - Mestrado Profissional em Computação Aplicada) -- Universidade de Brasília, 2021.

1. Gestão de riscos. 2. Reengenharia de software. 3. Melhoria de processo de software. I. Costa Oliveira, Edgard, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise e Melhoria de Processo de Reengenharia de Software: Proposta de Instrumento para Gerenciamento de Riscos do Processo

Guilherme Mendonça de Moraes

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

Prof. Dr. Edgard Costa Oliveira (Orientador)
PPCA/CIC e EPR/FT/UnB

Prof.a Dr. a Simone Borges Simão Monteiro
PPCA/CIC e EPR/FT/UnB

Prof. Dr. Altino Jose Mentzingen de Moraes
Engenharia de Produção/UNIP

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

Brasília, 28 de janeiro de 2021

Dedicatória

Primeiramente a Deus por sua grande misericórdia e graça sobre minha vida.

Ao meu avô/pai, José Matias Mendonça (in memorian) e ao meu irmão, Pedro Alexandre Mendonça do Nascimento (in memorian).

À minha esposa, Bruna Araújo da Silva Mendonça e aos meus filhos, Sophia Maria Araújo Mendonça, Kaleb José Araújo Mendonça e Benjamin Samuel Araújo Mendonça.

Agradecimentos

Agradeço imensamente ao Prof. Dr. Edgard Costa de Oliveira, por sua liderança nesse processo de pesquisa, todo seu apoio, paciência, compreensão e incentivo nessa caminhada e por ter acreditado em mim quando nem mesmo eu acreditava.

Agradeço à Prof.ª. Drª. Simone Borges Simão Monteiro, por ter me auxiliado em todo processo da pesquisa, desde quando eu ainda era um aluno especial e por toda parceria no processo das publicações e evolução dessa proposta de pesquisa.

Agradeço ao Prof. Dr. Altino Jose Mentzingen de Moraes, por ter contribuído para evolução dessa proposta de pesquisa.

Agradeço ao Prof. Dr. Marcelo Ladeira, pelo seu incentivo e por ter aberto essa grandiosa oportunidade de conhecer o Programa de Pós-graduação em Computação Aplicada.

Agradeço à Me. Ana Cristina Fernandes Lima por ter me apoiado na proposta inicial de pesquisa, ofertando seu tempo, conhecimento e dedicação.

Agradeço à Universidade de Brasília, por oferecer uma oportunidade tão valiosa de realizar um curso de altíssima qualidade e com reconhecimento mundial.

Agradeço a todos os professores do PPCA, por ofertarem o seu tempo para disseminação do conhecimento e crescimento intelectual/profissional dos alunos.

Agradeço ao Me. Eduardo Costa Cheng por todo incentivo e por ter acreditado do meu sonho de me tornar mestre e nessa proposta de pesquisa.

Agradeço ao Ricardo Rodrigues por ter me ajudado a evoluir como profissional e por ter me apresentado a Gestão de Riscos.

Agradeço ao Wellerson Aparecido da Costa por todo apoio técnico no desenvolvimento dessa pesquisa, seu profissionalismo e dedicação.

Agradeço ao Prof. Dr. Marcondes Gorgonho, Ph.D. por todo apoio profissional em minha caminhada e por ter endossado o meu conhecimento.

Agradeço à minha sogra, Lourdes Araújo Siqueira da Silva, por ter me apoiado no cuidado dos meus filhos em toda caminhada dessa pesquisa.

Agradeço à minha família, em especial a minha avó, Maria Gomes Mendonça, minha mãe, Rosemeri Gomes Mendonça e minha tia, Roselene Gomes Mendonça, por acreditarem no meu sonho e darem todo apoio de vida para que eu conseguisse chegar até aqui.

Agradeço principalmente à minha esposa, que sonhou, sofreu, chorou junto comigo e me apoiou nessa pesquisa de uma forma incrível. Sem ela, nada dessa pesquisa teria se tornado realidade.

Resumo

A reengenharia de software é responsável pela reconstrução de um sistema legado. Realizara gestão de riscos no processo de reengenharia de software é um desafio para as organizações, impactando no sucesso de projetos dessa natureza. Nesse contexto, essa proposta de pesquisa tem por objetivo geral propor um instrumento computacional para gerenciamento de riscos no processo de reengenharia de software. Como forma de atingir esse objetivo, visa-se realizar a modelagem do processo atual (AS-IS) de reengenharia de software, identificar os riscos relacionados a esse processo, redesenhar um processo melhorado (TO-BE) como forma de respostas aos riscos identificados e prototipar uma ferramenta computacional para a gestão de riscos no processo de reengenharia de software. Para isso, a natureza dessa pesquisa é a aplicada, com uma abordagem qualitativa de caráter exploratório utilizando a estratégia de estudo de caso *in-loco* numa instituição financeira de grande porte, ora denominada CredTec. Os resultados alcançados foram a modelagem do processo atual (AS-IS) de reengenharia de software, a identificação dos riscos relacionados a esse processo, o redesenho de um processo melhorado (TO-BE) como forma de resposta aos riscos e a construção de um instrumento computacional para análise e avaliação dos riscos do processo. Através disso conclui-se que é foi possível apoiar a CredTec no entendimento do seu processo atual de reengenharia de software, identificar, analisar e avaliar os riscos desse processo através de um instrumento computacional e remodelar um processo de reengenharia melhorado como forma de resposta aos riscos identificados.

Palavras-chave: Gestão de Riscos, reengenharia de software, melhoria de processo de software

Abstract

Software reengineering is responsible for the reconstruction of a legacy system. Performing risk management in the software reengineering process is a challenge for organizations, impacting the success of projects of this nature. In this context, this research proposal has the general objective of proposing a computational instrument for risk management in the software reengineering process. As a way to achieve this goal, we aim to model the current process (AS-IS) of software reengineering, identify the risks related to this process, redesign an improved process (TO-BE) as a way to respond to the identified risks and prototyping a computational tool for risk management in the software reengineering process. For this, the nature of this research is applied, with a qualitative approach of an exploratory character using the strategy of case study in place in a large financial institution, now called CredTec. The results achieved were the modeling of the current software reengineering process (AS-IS), the identification of risks related to this process, the redesign of an improved process (TO-BE) as a way of responding to risks and the construction of a computational instrument for analysis and assessment of process risks. This concludes that it was possible to support CredTec in understanding its current software reengineering process, identifying, analyzing and evaluating the risks of this process through a computational instrument and remodeling an improved reengineering process as a way to respond to risks identified.

Keywords: Risk management, software reengineering, software process improvement

Sumário

Lista de Figuras

Lista de Quadros

Lista de Abreviaturas e Siglas

ANSI American National Standards Institute.

API Application Programming Interface.

BPMN Business Process Model and Notation.

BPS Benefícios da Previdência Social.

CCO Conta Corrente.

CMMI Capability Maturity Model® Integration.

COBIT Control Objectives for Information and Related Technologies.

COTS Commercial Off-The-Shelf.

CRM Controle de Registro de Mudança.

DDL Data Definition Language.

DOC Documento de Ordem de Crédito.

E.A Enterprise Architect.

EPC Event-driven Process Chain.

IBM Business Machines Corporation.

INSS Instituto Nacional do Seguro Social.

ITIL Information Technology Infrastructure Library.

LGPD Lei geral de proteção aos dados pessoais.

MER Modelo de Entidade e Relacionamento.

MPsBR Melhoria no processo de software no Brasil.

MVP Minimum Viable Product.

OMG Object Management Group.

P.O Product Owner.

PMBoK® Project Management Body of Knowledge.

RUP Rational Unified Process.

TED Transferência Eletrônica Disponível.

UML Unified Modeling Language.

UX User Experience.

XP Extreme Programming.

Capítulo 1

Introdução

Neste capítulo são apresentadas as informações principais que contextualizam o tema a ser estudado, a definição do problema, a justificativa, os objetivos gerais e específicos e a estrutura da pesquisa.

1.1 Contextualização

Com o avanço tecnológico e o crescimento das necessidades de mercado, é comum que as empresas optem por realizar a mudança em sua estrutura tecnológica atual, visando a compatibilidade com outras linguagens, banco de dados e plataformas, como também o aumento do desempenho, usabilidade, acessibilidade, segurança e qualidade [1].

O software é o elemento-chave na evolução de produtos e sistemas baseados em computador e uma das mais importantes tecnologias no cenário mundial [2].

Existe um grande interesse das empresas em manter os sistemas legados em funcionamento, pois são considerados como propriedades e agregam lógicas do negócio codificadas, investimentos, anos de desenvolvimento e teste, experiências, estratégias empresariais etc., [3].

É necessário evoluir esses sistemas legados aplicando a reengenharia de software para que eles continuem funcionais, gerando valor para o negócio e não se tornem obsoletos [2] [4].

Além da evolução dos softwares legados para novas tecnologias, existe também a necessidade de corrigir erros conhecidos em ambiente produtivo desses sistemas e adicionar itens de qualidade como: confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade [5].

A atividade de reengenharia de software consome muitos recursos no desenvolvimento, sendo necessário que a organização avalie o seu custo-benefício [6].

Além da reengenharia de software ser uma tarefa onerosa, existe um alto risco nesse processo de mudança da tecnologia [7], pois o sistema legado precisa continuar

funcionando sem que as mudanças causem um efeito negativo no mesmo em ambiente produtivo.

Juntamente com as mudanças tecnológicas existem os riscos que a reengenharia representa para o negócio da instituição [8], pois a necessidade é que toda a estrutura dos aplicativos seja alterada sem que o software em ambiente produtivo fique comprometido.

Nesse contexto, o objetivo da reengenharia de software relacionado às expectativas de qualidade e funcionamento correto em ambiente produtivo do sistema migrado. Todavia, existem muitos riscos que permeiam o processo.

Já o risco pode ser considerado o efeito das incertezas sobre os objetivos [9]. Em outras palavras, é algo que pode causar um desvio nas expectativas iniciais.

O sucesso dos objetivos de negócios pode estar ligado diretamente aos níveis de qualidade dos controles e aperfeiçoamento da evolução da governança tecnológica que uma organização procura estabelecer [10]. Para algumas empresas o ativo de tecnologia representa parcial ou totalmente o seu valor ou atuação de mercado.

A gestão de riscos deve estar alinhada à evolução tecnológica das organizações e os controles que ela exerce sobre seus ativos e seus processos. Ativo é tudo aquilo que apresenta valor para uma organização: Pessoas, Processos, Ambiente e Tecnologia [11].

Todas as atividades de uma organização sujeitas a riscos devem ser gerenciadas. O processo de gestão de riscos auxilia o tomador de decisão, levando em consideração as incertezas, possibilidade de circunstâncias ou eventos futuros (intencionais ou não intencionais), seus efeitos sobre os objetivos acordados [12].

O gerenciamento de riscos tem como propósito identificar problemas em potencial antes que estes ocorram, assim atividades de contorno podem ser planejadas e utilizadas à medida que for necessário.

Um estudo recente sobre falhas de software feito pela Tricentis, uma empresa de desenvolvimento de software austríaca, revelou o quanto de prejuízo falhas ou *bugs* podem causar. Cerca de 3,7 bilhões de pessoas em 2017 foram afetadas por falhas de software mal desenvolvidos no ano de 2018, gerando perdas de cerca de US $1,7 trilhão em ativos [13].

No caso das instituições financeiras, por exemplo, que possuem grande parte do seu *business* atuando com a tecnologia, caso venham a ter alguma falha em seu software, tal falha poderá causar um grande impacto em seus ativos, fazendo com que a mesma tenha perdas operacionais: falhas no negócio, falhas em sistemas, falhas em processo, fraudes

internas e externas, perda de imagem e espaço no mercado [14], gerando até mesmo um marketing negativo.

Dessa forma entende-se que apesar da grande necessidade de realizar a migração dos sistemas legados, existem muitos riscos envolvidos no processo de reengenharia de software a ausência de uma gestão de riscos proativa pode resultar na concretização desses riscos, impactando negativamente nos objetivos de negócio.

1.2 Definição do problema

Este trabalho foi desenvolvido na empresa CredTec, cujo nome foi substituído por sigilo, e os projetos selecionados foram os de reengenharia de software dos produtos Conta Corrente (CCO) e Benefícios da Previdência Social (BPS).

Da mesma forma que o cenário atual de tecnologia da informação tem evoluído constantemente, a CredTec tem buscado acompanhar esse movimento visando potencializar seus produtos, aumentar sua abrangência e proporcionar uma melhor experiência para seus clientes, colaboradores e fornecedores.

Assim como contextualizado, a CredTec tem constantemente buscado melhorar sua tecnologia visando potencializar seus produtos, aumentar sua abrangência e proporcionar uma melhor experiência para seus clientes, colaboradores e fornecedores.

Com isso, ela tem investido em processos de migração de seus aplicativos legados para novas plataformas de aplicação e banco de dados com objetivo de atender às demandas do mercado tecnológico, como também, às expectativas e necessidades de seus clientes. Todavia, com essas mudanças, existem ameaças em torno do processo de reengenharia de software.

Dentro desse cenário, a CredTec identificou que seus projetos de renovação de tecnologia têm demandado muito tempo para sua conclusão, gerando um grande retrabalho com isso aumento de custo, prazo, qualidade e diminuição da satisfação dos gestores dos projetos. Isso também pode impactar diretamente os seus clientes que realizam uso constante dos aplicativos para acesso a conta corrente, transações financeiras, emissão de cheque, Documento de Ordem de Crédito (DOC), Transferência Eletrônica Disponível (TED), etc.

Ressalta-se que uma falha em um desses sistemas pode levar não só à perda de futuros clientes, risco à imagem da organização e prejuízo financeiro.

Esses fatores mostram a necessidade de se realizar a gestão de riscos proativa no processo de reengenharia de software, visando a qualidade do produto desenvolvido. Por

tanto, algumas questões de pesquisa foram levantadas para buscar respostas neste trabalho.

1. Como auxiliar a CredTec a melhorar seu processo de reengenharia de software?
2. Quais são os riscos envolvidos nesse processo?
3. Quais são os controles necessários aplicados às vulnerabilidades?
4. Como gerir riscos no processo de engenharia reversa pode auxiliar a área gestora e a instituição na tomada de decisões?

Na próxima seção será abordada a justificativa.

1.3 Justificativa

Os processos de reengenharia de software do produto CCO e BPS da CredTec envolvem tanto as células de equipes de trabalho alocadas no projeto, como também outros parceiros de trabalho, sejam eles externos ou internos à organização.

1.4 Objetivo Geral

Desenvolver um instrumento computacional para gerenciamento de riscos de processo de reengenharia de software do projeto de Renovação de Software do Produto Conta Corrente e Benefícios da Previdência Social da Instituição Financeira CredTec.

1.4 Objetivo Geral

Desenvolver um instrumento computacional para gerenciamento de riscos de processo de reengenharia de software do projeto de Renovação de Software do Produto Conta Corrente e Benefícios da Previdência Social da Instituição Financeira CredTec.

1.4.1. Objetivos Específicos

Para realização do objetivo geral, essa pesquisa busca os seguintes objetivos específicos:

1. Desenhar o processo AS-IS e realizar a melhoria do processo de reengenharia de software (TO-BE).
2. Identificar e aplicar modelos de gestão de riscos e controles aplicados ao processo atual (AS-IS) de reengenharia de software.

3. Prototipar uma ferramenta para gestão de riscos no processo de reengenharia de software.

Na próxima seção será apresentada a estrutura da pesquisa.

1.5 Estrutura da Pesquisa

A divisão geral dos capítulos dessa pesquisa foi estruturada da seguinte forma:

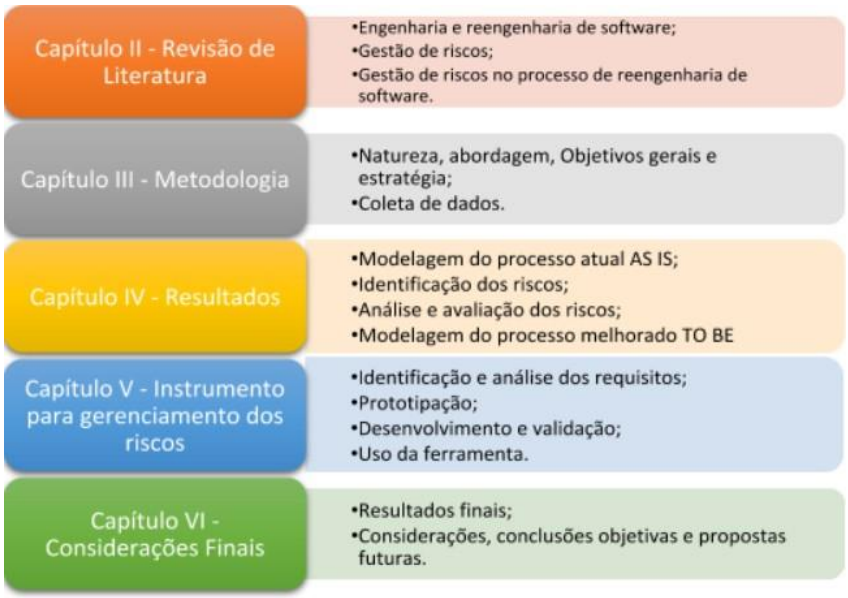


Figura 1: Estrutura da Pesquisa. Fonte: Próprio Autor.

A seguir será apresentada a revisão de literatura da pesquisa.

Capítulo 2

Revisão da Literatura

Neste capítulo abordamos o referencial teórico da pesquisa em questão, assim como os conceitos fundamentais que serviram como referência principal para o desenvolvimento do projeto.

Na revisão de literatura em questão buscou-se identificar os estudos sobre as temáticas de "Engenharia de Software", "Reengenharia de Software", "Gestão de Riscos" e a inter-relação entre eles.

Utilizando descritores como "*Software Reengineering*" OR "*Software Migration*" OR "*Software Reverse*" AND "*Risk*", como também "Engenharia de Software", "Reengenharia de Software" e "Gestão de Riscos", foram mapeados os principais trabalhos sobre Reengenharia de Software e Gestão de Riscos.

As bases utilizadas foram Web Of Sciencee o Google Acadêmico entre um período de 2009 a 2020.Além disso foram identificados os principais autores sobre essas temáticas, que em alguns casos estão fora do período temporal informado, porém possuem uma alta relevância para a pesquisa.

2.1 Reengenharia de Software

A reengenharia de software está presente na maioria das instituições que fazem algum tipo de uso de ferramentas tecnológicas. Sendo que algumas optam pelo uso de métodos tradicionais e outras por um desenvolvimento mais ágil.

De acordo com Sommerville [2], os sistemas de software são abstratos e intangíveis, não se limitando às propriedades materiais, leis físicas ou processos de produção. Todavia, justamente por não haver tais limitações, o software se torna caro e sua manutenção complexa.

Existem vários tipos de sistemas como: aplicativos de smartphones com a função de calcular, softwares embarcados em veículos, games digitais, softwares aplicados à saúde

ou até mesmo sistemas bancários de grande porte com impacto e uso no âmbito nacional e internacional.

Existem vários métodos ou técnicas universais para a reengenharia de software, pois diferentes tipos de software exigem abordagens igualmente diferentes [2], de acordo coma necessidade de cada organização.

O termo Engenharia de software foi criado na década de 1960 e oficialmente institucionalizado na NATO Science Committee, que na ocasião discutia sobre a crise do software, na tentativa de organizar os desenvolvimentos complexos de software com o uso da engenharia [15].

Durante as décadas de 70 e 80 nos EUA, foram desenvolvidas técnicas de engenharia de software que serviriam como base para a engenharia de software atual, como por exemplo: programação estruturada, desenvolvimento orientado a objetos e ocultação da informação [2].

A engenharia de software é conhecida como a disciplina da engenharia que se preocupa com os aspectos de produção do software, desde sua concepção até sua operação e manutenção [2], utilizando aplicações de abordagem sistemáticas associadas ao desenvolvimento, operação e manutenção de sistemas [16].

A engenharia de software está relacionada ao desenvolvimento profissional do software, sendo o software utilizado para um determinado propósito ou necessidade [4].

É importante ressaltar, que na visão de engenharia de softwares, um software em si é apenas um programa de computador, mas toda documentação, bibliotecas, websites de apoio e dados de configuração, que são elementos necessários para o funcionamento do software em si [2].

Um dos aspectos importantes na engenharia de software é a preocupação com a qualidade que o produto de software irá proporcionar, sendo a qualidade uma das áreas da engenharia de software mais notória.

2.1.1 Qualidade de Software

Um software pode ser utilizado para vários fins, desde os mais simples como realizar o cálculo automatizado de uma função de segundo grau através de parâmetros informados pelo usuário, até um software para uma bomba de insulina ou sistema de tráfego aéreo.

Quando se fala sobre a qualidade do software, temos que considerar que o software é utilizado e modificado por outras pessoas além dos seus desenvolvedores. Portanto, a

qualidade do software não está relacionada apenas àquilo que ele faz, mas como também aspectos voltados à sua manutenção.

A ISO 25000 [5] propõe um conjunto de especificações de atributos que pode se esperar de um software de qualidade, compreendendo oito características de qualidade relacionadas na figura 2:

Figura 2 Qualidade de Software. Fonte: Próprio autor, adaptado de ISO25010 [5].

Cada um dos itens citados pela ISO 25010 (2011) [5] pode ser compreendido da seguinte maneira:

- Adequação funcional:
    - **Completude**: grau em que o conjunto de funcionalidades cobre todas as tarefas e objetivos do usuário especificados.
    - **Correção**: grau em que as funcionalidades fornecem os resultados corretos com o grau de precisão necessário.
    - **Propriedade**: grau em que as funcionalidades facilitam a realização de tarefas e objetivos especificados.
- Eficiência e Desempenho:
    - **Tempo de resposta**: grau em que o tempo de resposta e processamento e taxas de rendimento de um produto ou sistema, quando desempenhar suas funções, atender aos requisitos.
    - **Utilização de recursos**: grau em que as quantidades e tipos de recursos utilizados por um produto ou sistema, ao realizar suas funções e atender aos requisitos.
    - **Capacidade**: grau em que os limites máximos do produto ou sistema, parâmetro atendem aos requisitos.

- Compatibilidade:
    - **Coexistência**: grau em que um produto pode executar suas funções necessárias de forma eficiente, enquanto compartilha um ambiente e recursos com outros produtos, sem impacto prejudicial em qualquer outro produto.
    - **Interoperabilidade**: grau em que dois ou mais sistemas, produtos ou componentes podem trocar informações e usar as informações que foram trocadas.
- Usabilidade:
    - **Reconhecimento da propriedade**: grau em que os usuários podem reconhecer se um produto ou sistema é apropriado para suas necessidades.
    - **Aprendizagem**: grau em que um produto ou sistema permite ao usuário aprender como usá-lo com eficácia e eficiência em emergenciais.
    - **Operabilidade**: grau em que um produto ou sistema é fácil de operar, controlar e é apropriado para usar.
    - **Proteção ao erro do usuário**: grau em que um produto ou sistema protege os usuários contra cometer erros.
    - **Estética**: grau em que uma interface de usuário permite uma interação agradável e satisfatória para o usuário.
    - **Acessibilidade**: grau em que um produto ou sistema pode ser usado por pessoas com a mais ampla gama de características e recursos para atingir um objetivo específico em um contexto de uso específico.
- Confiabilidade:
    - **Maturidade**: grau em que um sistema, produto ou componente atende às necessidades de confiabilidade em operação normal.
    - **Disponibilidade**: grau em que um produto ou sistema está operacional e acessível quando necessário para uso.
    - **Tolerância a falhas**: grau em que um sistema, produto ou componente opera conforme pretendido, apesar da presença de hardware ou falhas de software.
    - **Recuperabilidade**: grau em que, em caso de interrupção ou falha, um produto ou sistema pode recuperar os dados afetados diretamente e restabelecer estado desejado do sistema.
- Segurança:

    - **Confidencialidade**: grau em que um sistema garante que os dados são acessíveis apenas para aqueles autorizados a ter acesso.
    - **Integridade**: grau em que um sistema, produto ou componente impede o acesso não autorizado a, ou modificação de, programas de computador ou dados.
    - **Não-repúdio**: grau em que ações ou eventos podem ser provados ter ocorrido, de modo que os eventos ou ações não podem ser repudiados mais tarde.
    - **Rastreabilidade**: grau em que as ações de uma entidade podem ser rastreadas exclusivamente para a entidade.
    - **Autenticidade**: grau em que a identidade de um sujeito ou recurso pode ser comprovada como sendo o reivindicado.
- Manutenção:
    - **Modularidade**: grau em que um sistema ou programa de computador é composto de componentes discretos, de modo que uma mudança para um componente tem impacto mínimo em outros componentes.
    - **Reusabilidade**: grau em que um ativo pode ser usado em mais de um sistema ou na construção de outros ativos.
    - **Analisabilidade**: grau de eficácia e eficiência com o qual é possível avaliar o impacto sobre um produto ou sistema de uma alteração pretendida em uma ou mais de suas partes, ou para diagnosticar um produto quanto a deficiências ou causas de falhas, ou para identificar peças a serem modificadas.
    - **Modificabilidade**: grau em que um produto ou sistema pode ser modificado de forma eficaz e eficiente sem introduzir defeitos ou degradação da qualidade do produto existente.
    - **Testabilidade**: grau de eficácia e eficiência com o qual os critérios de teste podem ser estabelecidos para um sistema, produto ou componente e testes podem ser realizados para determinar se esses critérios foram atendidos.
- Portabilidade:
    - Adaptabilidade: grau em que um produto ou sistema pode ser eficaz e eficientemente adaptado para diferentes ou em evolução hardware, software ou outros ambientes operacionais ou de uso.

- Instabilidade: grau de eficácia e eficiência em que um produto ou sistema pode ser instalado com sucesso e / ou desinstalado em um ambiente especificado.
- Substituibilidade: grau em que um produto pode substituir outro produto de software especificado para a mesma finalidade no mesmo ambiente.

Para o nosso estudo de caso deste trabalho, por exemplo, o software utilizado é um sistema bancário de conta corrente e pagamento de benefícios da previdência social, envolvendo a movimentações grandes de forma macro e que impactam diretamente os clientes da instituição.

Dessa forma, entende-se que a qualidade é um fator indispensável num software profissional.

2.1.2 Processo de Reengenharia de Software

Processo, em sua definição clássica, é uma série de atividades executadas sequencialmente para produzir um produto ou serviço. Pode ter os mais variados propósitos: criar, inventar, projetar, transformar, produzir, controlar, manter e usar produtos e sistemas [17].

Um processo básico pode ser determinado pela figura 2.2: Um processo básico pode ser determinado pela figura 2.2:

Figura 3 - Estrutura de Processo. Fonte: Próprio autor, adaptado de Machado [17]

Sommerville [2] diz que um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software.

Essas atividades podem envolver o desenvolvimento de software a partir do zero em uma linguagem padrão de programação ou modificação de aplicativo existente, ou seja, a reengenharia de software.

Para que um software seja desenvolvido, os profissionais envolvidos nesse processo utilizam de teorias, métodos e ferramentas de forma apropriada e de acordo com a realidade de cada empresa [2], obedecendo os limites financeiros e organizacionais, o que incluem processos de engenharia de software.

Vale ressaltar que a engenharia de software não se preocupa apenas com os itens técnicos relacionados ao desenvolvimento de software. Todavia, irá abranger também métodos, processos e atividades para o gerenciamento do projeto, teste do software e sua manutenção em ambiente produtivo [18].

Existem muitos processos de engenharia de software diferentes, mas todos devem incluir, de alguma forma, quatro atividades fundamentais para a engenharia de software [17]:

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

A engenharia de software engloba processos, métodos e ferramentas que possibilitam construção de sistemas complexos baseados em computador dentro do cronograma definido, restrição de orçamento e qualidade esperada [19].

Sommerville [2] destaca que um dos grandes desafios da engenharia ou reengenharia de software é a construção de sistemas de forma mais ágil, mais confiável e com maior qualidade.

Existem vários processos de desenvolvimento de software conhecidos no ramo de desenvolvimento, entre eles: modelo em cascata, modelo incremental, modelo em espirale engenharia de reuso, que é a utilização de componentes existentes e conhecidos para compor desenvolver ou alterar um outro sistema [20] [2] [6] [17].

Além dos métodos tradicionais, existem outros métodos mais ágeis para o desenvolvimento de software como: *Extreme Programming* (XP) e o *Scrum Framework*, por exemplo [21].

É importante ressaltar que cada software ou organização possui sua particularidade no processo de engenharia ou reengenharia de software.

Exemplo: um sistema de bomba de insulina ou de uma aeronave, necessariamente precisa ser totalmente elicitado, documentado e desenvolvido para que possa ser entregue ao usuário final [2].

O que será diferente de um *e-commerce* que necessariamente pode ser entregue por módulos ou funcionalidades.

2.1.3 Modelos tradicionais de processo de software

Ao longo dos anos, logo após a década de 70, foram surgindo uma série de modelos de processo de software que foram aprimorados, gerando os que conhecemos até hoje [17].

Um processo de software, às vezes chamado de ciclo de vida do desenvolvimento de software, é apenas uma representação simplificada de um processo de software [2].

Cada um desses modelos possui suas particularidades e exibem o processo de desenvolvimento de software de uma ótica diferente, uns considerando o enfoque apenas nos macroprocessos e outros destrinchando por atividades desses do processo em si.

Os modelos tradicionais de processo de software identificados são: modelo em cascata, desenvolvimento incremental e reuso [7].

Esses modelos genéricos são descrições mais gerais e abstratas dos processos de software, e podem ser utilizados para explicar as diferentes abordagens ao desenvolvimento de software.

Modelo Cascata

O primeiro modelo de processo de software desenvolvido a ser publicado foi o modelo cascata de Royce de 1970 [4].

Conhecido também como ciclo de vida clássico ou tradicional do software, o modelo cascata representa as atividades fundamentais do processo de engenharia ou reengenharia de software como: especificação, desenvolvimento, validação e evolução.

Uma forma de utilizar o modelo cascata é quando se fazem necessárias adaptações ou aperfeiçoamentos em um sistema já existente [20]. Por exemplo, quando existe um sistema já pronto é necessário fazer uma adequação ou reengenharia.

Também é possível utilizar o modelo cascata quando um software necessita de uma nova funcionalidade e os requisitos estão bem definidos e são estáveis.

Este modelo sugere uma abordagem sequencial e sistemática para o desenvolvimento de software, conforme a figura 4:

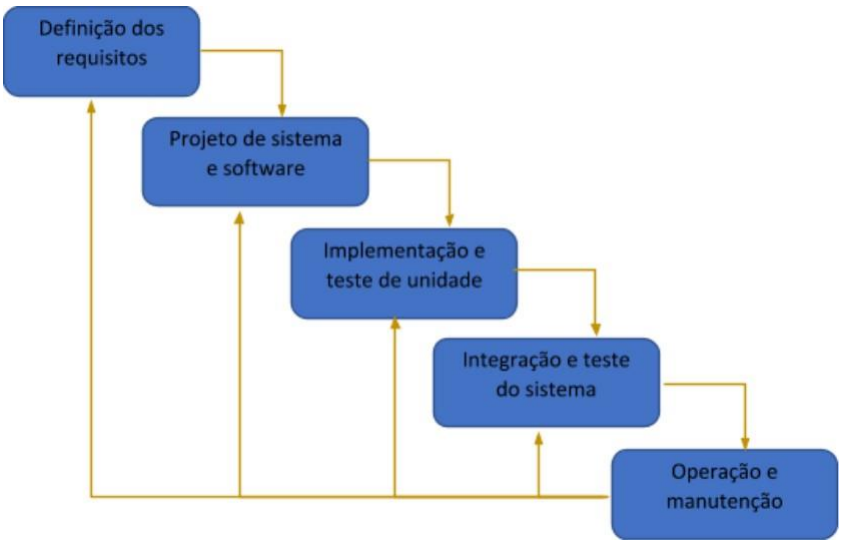


Figura 4 - Modelo Cascata. Fonte: Próprio autor, adaptado de Sommerville [2]

Nesse modelo, inicia-se com o levantamento de requisitos ou necessidades junto ao cliente, depois inicia-se a fase de planejamento onde é definida as estimativas, cronograma acompanhamento [2].

Posteriormente é dado início a modelagem onde é feita a análise e o projeto, seguidos da construção, onde é realizada a codificação e testes, passamos para a implantação ou emprego onde efetuamos a entrega, suporte e feedback do software concluído.

De forma mais detalhadas, as atividades são descritas da seguinte forma:

- **Análise e definição dos requisitos:** os serviços, as restrições e as metas do sistema são estabelecidas por meio de consulta aos usuários. Depois eles são definidos em detalhes e servem como uma especificação de sistemas.
- **Projeto do sistema e do software:** O processo de projeto do sistema reparte os requisitos entre requisitos de sistemas de hardware e de software, e estabelece uma arquitetura global do sistema. O projeto de software envolve a identificação e a descrição das abstrações fundamentais do sistema de software e seus relacionamentos.
- **Implementação e teste de unidade:** Durante essa etapa, o projeto do software é realizado como um conjunto de programas ou unidades de programa. O teste de unidade envolve a verificação de cada unidade, conferindo se satisfazem a sua especificação.

- **Integração e teste de sistema:** As unidades de programa ou os programas são integrados e testados como um sistema completo a fim de garantir que os requisitos.
- de software tenham sido cumpridos. Após os testes, o sistema de software é entregue ao cliente.

Desenvolvimento Incremental

O desenvolvimento incremental se baseia na ideia de desenvolver uma implementação inicial, obter feedback dos usuários ou terceiros e fazer o software evoluir através de várias versões, até alcançar o sistema necessário.

As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de separadas, com feedback rápido ao longo de todas elas, conforme figura 5:

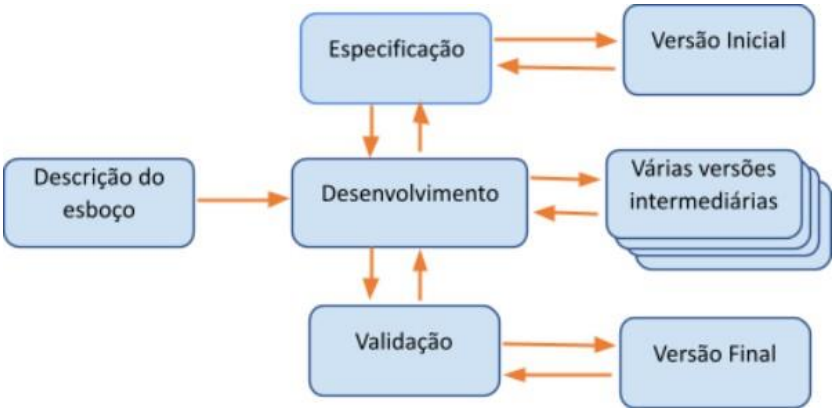


Figura 5 Modelo Incremental. Fonte: Adaptação de Sommerville [2]

De acordo com Machado [17], a engenharia incremental é melhor do que uma abordagem em cascata para a maioria dos sistemas de negócios, e-commerce e sistemas pessoais, pois ele reflete a maneira como os problemas são resolvidos, tendo em vista comumente uma solução de determinado problema não elaborada de uma só vez e sim forma incremental.

Cada incremento ou versão do sistema incorpora alguma funcionalidade necessária para o cliente.

Frequentemente, os incrementos iniciais incluem a funcionalidade mais importante ou mais urgente. Isso significa que o cliente pode avaliar o sistema em um estágio relativa- mente inicial do desenvolvimento para ver se ele oferece o que foi requisitado.

Em caso negativo, só o incremento que estiver em desenvolvimento no momento precisará ser alterado e, possivelmente, nova funcionalidade deverá ser definida para incrementos posteriores.

Modelo de reengenharia orientado ao reuso

Já a reengenharia de software ou reversa de software é o processo de mudança da tecnologia dos sistemas legados aprimorando o mesmo, como por exemplo o processo de engenharia orientada a reuso onde há alguma reutilização de software.

Isso acontece muitas vezes informalmente, quando as pessoas envolvidas no projeto sabem de projetos ou códigos semelhantes ao que é exigido. Elas os buscam, fazem as modificações necessárias e incorporam-nas a seus sistemas.

Esse reuso informal ocorre independentemente do processo de desenvolvimento que se use. No entanto, no século XXI, processos de desenvolvimento de software com foco no reuso de software existente tornaram-se amplamente usados [17].

Abordagens orientadas a reuso dependem de uma ampla base de componentes reusáveis de software e de um framework de integração para a composição desses componentes [2]. Em alguns casos, esses componentes são sistemas completos (COTS ou de prateleira), capazes de fornecer uma funcionalidade específica, como processamento de texto ou planilha, vide figura 6:

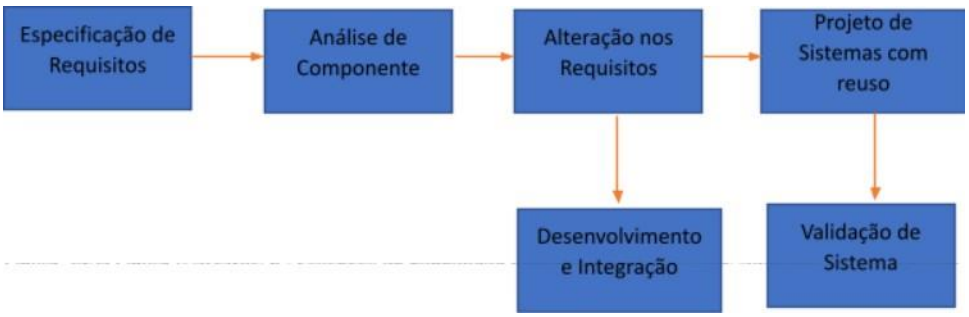


Figura 6 Modelo Reuso. Fonte: Adaptado de Sommerville [2]

Embora o estágio de especificação de requisitos iniciais e o estágio de validação sejam comparáveis a outros processos de software, os estágios intermediários em um processo orientado a reuso são diferentes.

Esses estágios são:

- **Análise de componentes**: Dada a especificação de requisitos, é feita uma busca por componentes para implementar essa especificação. Em geral, não há corres-

pendência exata, e os componentes que podem ser usados apenas fornecem alguma funcionalidade necessária.

- **Modificação de requisitos**: Durante esse estágio, os requisitos são analisados usando-se informações sobre os componentes que foram descobertos. Em seguida, estes serão modificados para refletir os componentes disponíveis. No caso de modificações impossíveis, a atividade de análise dos componentes pode ser inserida na busca por soluções alternativas.
- **Projeto do sistema com reuso**: Durante esse estágio, o framework do sistema é projetado ou algo existente é reusado. Os projetistas têm em mente os componentes que serão reusados e organizam o framework para reuso. Alguns softwares novos podem ser necessários, se componentes reusáveis não estiverem disponíveis.
- **Desenvolvimento e integração**: Softwares que não podem ser adquiridos externamente são desenvolvidos, e os componentes e sistemas Commercial Off-The-Shelf (COTS) são integrados para criar o novo sistema. A integração de sistemas, nesse modelo, pode ser parte do processo de desenvolvimento, em vez de uma atividade separada.

2.1.4 Métodos tradicionais de desenvolvimento de software

Como citado anteriormente, existem vários modelos genéricos de processo de software, porém, cada organização, ao longo dos anos, adota os processos ou atividades de processos que melhor se adequem com sua realidade.

Em meados de 1990, a Rational Software Corporation criou o Rational Unified Process (RUP) ou Processo Unificado da Rational [22], que buscou unificar o processo de engenharia ou reengenharia de software.

Em 2003 a Rational foi comprada pela Business Machines Corporation (IBM) e o processo RUP começou a ser difundido em grande escala, sendo esse utilizada até hoje em grande escala [4], como é o caso do governo federal.

Em comparação aos métodos apresentados anteriormente, para a época o RUP era considerado um modelo mais ágil de desenvolvimento de software, tendo em vista que focava em entregas incrementais, por iterações que duravam cerca de 3 meses.

Dessa forma, ao invés de documentar todo o sistema para posteriormente iniciar o desenvolvimento, testes e entregar ao cliente, os ciclos eram divididos e as entregas mais curtas [17].

O RUP reúne os elementos de todos os modelos de processos genéricos de software e apoia a entrega incremental do software.

A proposta do RUP era que todo o fluxo de especificação, desenvolvimento, teste e entrega, acontecesse por essas iterações de três meses [2].

A abordagem RUP é a de orientação a objetos em sua concepção e é projetado e documentado utilizando a notação Unified Modeling Language (UML) para ilustrar os processos em ação [23]. Utiliza técnicas e práticas aprovadas comercialmente, conforme exemplo da figura 7:

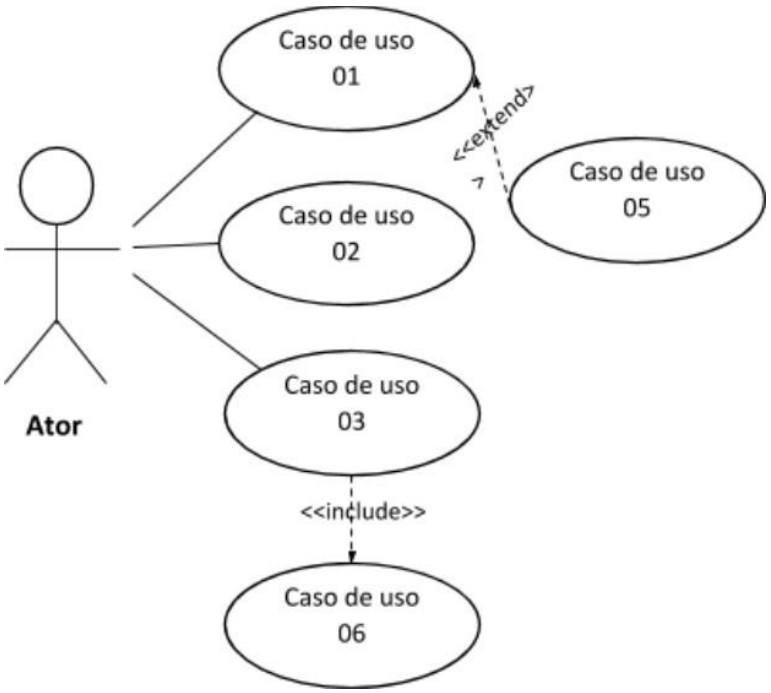


Figura 7 Diagrama de Caso de Uso. Fonte: Próprio autor, adaptado de OMG® [24]

Conforme apresentado na figura 7, este é um exemplo de diagrama de caso de uso, que mostra o relacionamento entre o ator, que seria um usuário de sistema [25], com as respectivas funcionalidades, descritas aqui como “caso de uso”.

Além da utilização da notação gráfica UML para desenho de vários diagramas, como forma de apoio às atividades de processo, sua dinâmica propõe quatro fases divididas em nove disciplinas [25] conforme a figura 8:

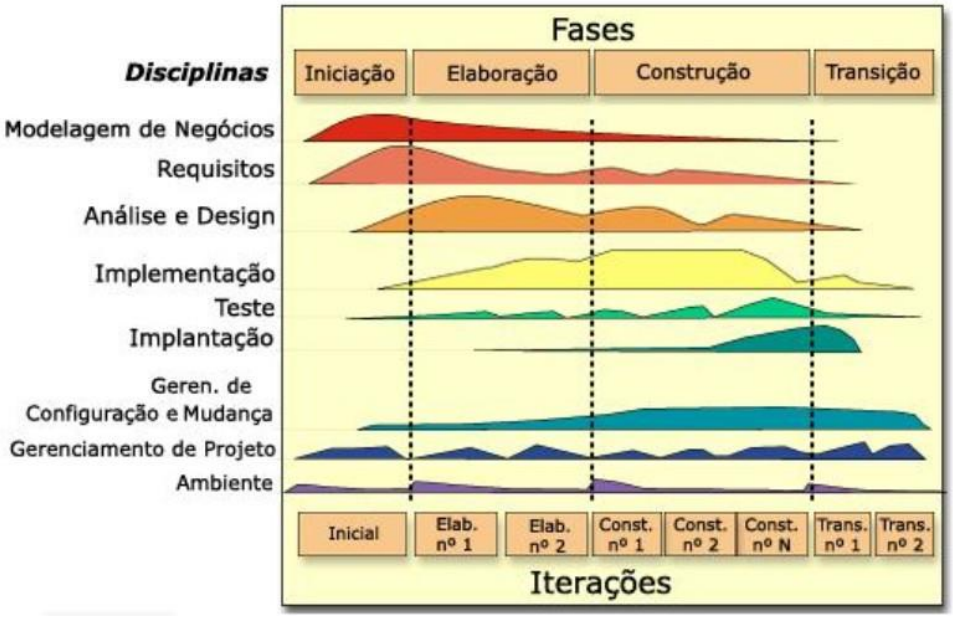


Figura 8 Disciplinas do RUP. Fonte: IBM [22]

O RUP organiza o desenvolvimento em 4 fases bem direcionadas, contendo em cada uma delas no mínimo uma iteração, ou seja, um ciclo de vida, são nessas iterações que são mostradas ao cliente o andamento da produção para que ele possa validar e assim liberar a continuação do desenvolvimento [22]. São elas:

- **Concepção:** define o escopo do software. É uma fase preliminar, é nessa etapa que se concentra o levantamento de requisitos, define preços e prazos da entrega do sistema e onde se avalia os possíveis riscos.
- **Elaboração:** plano do projeto, especificação de características e arquitetura. Aqui todas as análises de riscos são aprofundadas, como também os custos.
- **Construção:** ocorre a codificação do software.
- **Transição:** implantação do software, assegurando que ele esteja disponível aos usuários finais. Nesta fase estão incluídos os testes e o treinamento dos usuários.

As disciplinas do RUP podem ser separadas da seguinte forma:

- Modelagem de negócio: tem por objetivo estabelecer uma melhor compreensão e canal de comunicação entre engenharia de negócios e engenharia de software:
    - Compreender a estrutura e a dinâmica da empresa e os atuais problemas na organização;

    - Identificar possíveis melhorias no processo atual;
    - Garantir o entendimento comum entre os clientes e os desenvolvedores;
    - Descrever como o sistema será implantado e usá-la como uma base para descrever os processos, papéis e responsabilidades.
- **Requisitos**: disciplina destinada à identificação e análise de requisitos das partes interessadas (*stakeholders*) e como detalhá-los de forma que seja possível desenvolver os sistemas a partir desses [25].
    - Os requisitos de um sistema são as descrições dos serviços que o Sistema deve prestar e as restrições a sua operação. Esses requisitos refletem as necessidades dos clientes de um Sistema que atende a um determinado propósito". Exemplo: controlar um dispositivo, fazer um pedido de comida, encontrar informações de clientes [2].
    - A disciplina de requisitos está no atendimento de três atividades minimamente:
        - Elicitar e analisar: Identificar/descobrir e analisar, por meio de interações com as partes interessadas/usuários (*stakeholders*) do sistema, suas necessidades. (podendo envolver a análise de processos, sistemas, documentação pré-existente etc.).
        - Especificar: Registrar/documentar a conversão de todos os requisitos de uma forma padrão.
        - Validar: Verificação junto ao cliente se os requisitos estão corretos.
    - Os requisitos identificados seguirão o fluxo do projeto como um todo, desde requisitos macros, como objetivos de negócio, até o detalhamento de cada um deles em forma de caso de uso, por exemplo.
    - Esses requisitos podem ser divididos como:
        - Requisitos de negócio: são declarações de mais alto nível de objetivos, metas ou necessidades da organização. Eles descrevem as razões pelas quais um projeto foi iniciado, as metas que o projeto deve atingir e as métricas que serão utilizadas para aferir o seu sucesso.
        - Restrições ou premissas: não são consideradas requisitos funcionais, porém, impactam diretamente neles, traçando (de certa forma) um caminho ao qual o projeto deve ou não seguir. Essas

restrições podem ser, por exemplo: de projeto, de software, arquitetura e da própria organização.

- Requisitos de usuário: são requisitos em alto nível e de forma abstrata que descrevem a necessidade da parte interessada. Esse tipo de requisito poderá se tornar um ou vários requisitos de sistema.
- Requisitos de sistema: São descrições mais detalhadas das funções, dos serviços e das restrições operacionais do sistema de software. O documento de requisitos de sistema (chamado às vezes de especificação funcional) deve definir exatamente o que deve ser implementado. Pode ser descrito também como "requisitos funcionais".
- Requisitos não funcionais: são restrições ou exigências sobre os serviços ou funções oferecidas pelo sistema. Podem ser incluídas: restrições de tempo, exigências sobre o processo de desenvolvimento e imposta por padrões.

- Análise e Projeto ("*Design*"): O objetivo da análise e projeto é mostrar como o sistema vai ser realizado.
  - Exemplo, é necessário construir um sistema que:
    - Execute, em um ambiente específico, as tarefas e funções especificadas nas descrições de casos de uso.
    - Cumpra todas as suas necessidades.
    - Seja fácil de manter quando ocorrerem mudanças de requisitos funcionais
  - Resultados de projeto em um modelo de análise e projeto tem, opcionalmente, um modelo de análise.
  - O modelo de design serve como uma abstração do código-fonte, isto é, o projeto atua como um modelo de "gabarito "de como o código-fonte é estruturado e escrito.
  - O modelo de projeto consiste em classes de design estruturado em pacotes e subsistemas com interfaces bem definidas, representando o que irá se tornar componentes da aplicação.
  - Ele também contém descrições de como os objetos dessas classes colaboram para desempenhar casos de uso do projeto.

- Teste: O RUP propõe uma abordagem iterativa, o que significa que se deve testar todo o projeto. Isto permite encontrar defeitos tão cedo quanto possível, o que reduz radicalmente o custo de reparar o defeito. Os objetivos dos testes são:
    - Verificar a interação entre objetos.
    - Verificar a integração adequada de todos os componentes do software.
    - Verificar se todos os requisitos foram corretamente implementados
    - Identificar e garantir que os defeitos são abordados antes da implantação do software.
- Garantir que todos os defeitos sejam corrigidos, analisados e fechados. Disciplina de Implantação: O objetivo da implantação é o de produzir com sucesso lançamentos de produtos e entregar o software para seus usuários finais. Ele cobre uma vasta gama de atividades, incluindo a produção de releases externas do software, a embalagem do software e aplicativos de negócios, distribuição do software, instalação do software e prestação de ajuda e assistência aos usuários.
- Disciplina de Ambiente: O ambiente enfoca as atividades necessárias para configurar o processo para um projeto. Ele descreve as atividades necessárias para desenvolver as diretrizes de apoio a um projeto. A proposta das atividades de ambiente é prover à organização de desenvolvimento de software os processos e as ferramentas que darão suporte à equipe de desenvolvimento.
- Disciplina de Configuração e Gerência de Mudança: A disciplina de Gestão de Mudança em negócios com RUP abrange três gerenciamentos específicos: de configuração, de solicitações de mudança, e de status e medição.
    - Gerenciamento de configuração: A gestão de configuração é responsável pela estruturação sistemática dos produtos. Artefatos, como documentos e modelos, precisam estar sob controle de versão e essas alterações devem ser visíveis. Ele também mantém o controle de dependências entre artefatos para que todos os artigos relacionados sejam atualizados quando são feitas alterações.
    - Gerenciamento de solicitações de mudança: Durante o processo de desenvolvimento de sistemas com muitos artefatos existem diversas versões. O Controle de Registro de Mudança (CRM) mantém o controle das propostas de mudança.
    - Gerenciamento de status e medição: Os pedidos de mudança têm os estados: novo, conectado, aprovado, cedido e completo. A solicitação de

mudança também tem atributos como a causa raiz, ou a natureza (como o defeito e valorização), prioridade, etc.

- Disciplina de Gerência de Projeto: Esta disciplina concentra-se principalmente sobre os aspectos importantes de um processo de desenvolvimento iterativo: Gestão de riscos; Planejamento de um projeto iterativo através do ciclo de vida e para uma iteração particular; E o processo de acompanhamento de um projeto iterativo, métricas. No entanto, esta disciplina do RUP não tenta cobrir todos os aspectos do gerenciamento de projetos. Abrange questões como:
    - Gestão de Pessoas: contratação, treinamento, etc.
    - Orçamento Geral: definição, alocação, etc.
    - Gestão de Contratos: com fornecedores, clientes, etc.

E por fim, o RUP procura focar nos quatro "P´s" [17] que servem como guia do processo como um todo que são: Pessoas, projeto, produto e processo.

2.1.5 Métodos ágeis de engenharia de software

O software está presente em praticamente todo segmento de negócio [7] e cada organização possui uma particularidade quando se trata do processo de desenvolvimento e a entrega do software em si.

Existem empresas que trabalham com software de grande escala e que possuem a necessidade de estarem 100% completos para serem utilizados em produção [2], como é o caso de sistemas bancários para saque bancários.

Outros softwares são mais simples ou possuem a particularidade de poderem ser utilizados ou a organização detentora da necessidade, deseja apenas validar hipóteses de uso de suas ideias [26], disponibilizando então esses incrementos de software em *Minimum Viable Product* (MVP) ou Mínimo Produto Viável.

Alguns autores associam uma entrega de software mais rápida com a qualidade do produto. Todavia, a ideia da agilidade em sua grande maioria é de entregar incrementos de software em iterações menores [26], fazendo com que o cliente consiga utilizar mais rápido e validar mais rápido a hipótese do software em si.

Entre os anos de 1999 e 2001 começaram a surgir as primeiras ideias de métodos ágeis para desenvolvimento, como por exemplo, o Extreme Programming (XP) ou programação extrema de Beck [27], que por exemplo, propunha uma escrita mais enxuta para requisitos funcionais em forma de histórias de usuário ou o próprio Framework

Scrum com entregas incrementais de um incremento de software em prazos mais curtos de 14 dias.

A ideia da agilidade ganhou força com a divulgação do "Manifesto Ágil" em 2001 por um grupo de "agilistas" da época [28].

Esse manifesto foi separado em 4 valores a saber, conforme a figura 9:

Figura 9 Valores do Ágil. Fonte: Próprio Autor, adaptado de Manifesto Ágil [28]

Conforme figura 9, a ideia principal atrás da agilidade é tornar os integrantes mais próximos e colaborativos. Além disso, foca na disponibilização de software em produção, acima de documentação abrangente.

É possível observar que os valores do ágil também estão relacionados à flexibilidade do projeto para mudanças no que for necessário, além da disseminação da comunicação desburocratizada das partes envolvidas.

Esse manifesto foi separado em doze princípios a saber [28]:

1. A maior prioridade é satisfazer o 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, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
4. Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto por todo o projeto.
5. Construa projetos em torno de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer 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 aumenta a agilidade.
10. Simplicidade a arte de maximizar a quantidade de trabalho 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.

Ao contrário do foco em processos, o manifesto ágil apresenta doze princípios que estão focados no cliente, e na entrega de valor ao mesmo através de entregas contínuas tendo como centro a solução proposta [28] [26] [29].

É importante ressaltar que ser ágil não é entregar algo mal feito, mas sim focar no trabalho em equipe, em que todos sejam participantes do processo construtivo de um produto de qualidade para o cliente.

Uma pesquisa de 2020 feita pela State of Agile [30] mostra o uso dos métodos ágeis e como estão presentes nas organizações atuais, conforme figura 10:

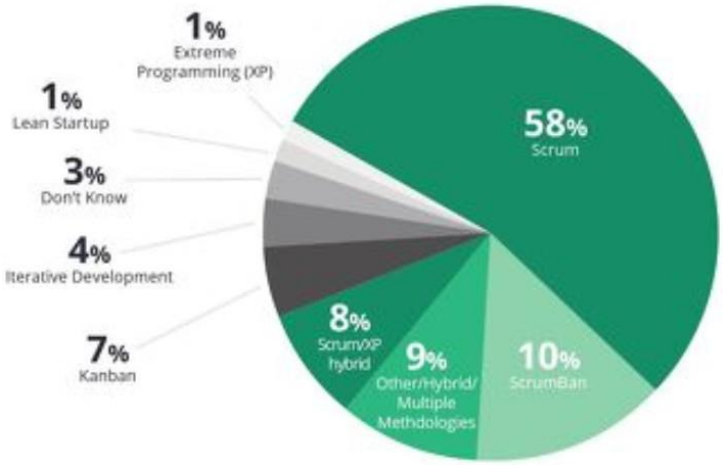


Figura 10 Dados sobre o uso da agilidade. Fonte: State of Agile [30]

A figura 10 quantifica em forma de porcentagem os métodos mais utilizados dentro das organizações que adotam algum tipo de metodologia ágil, como por exemplo o Framework Scrum que aparece em 58% delas ou a identificação de requisitos na concepção de projetos com o uso do Lean Inception de Caroli [26].

Como a pesquisa realizada apresenta o maior foco de concentração de métodos ágeis, a utilização do Scrum framework, iremos abordar um tópico sobre essa metodologia.

Framework Scrum esquematizado

O Scrum foi criado por Jeff Sutherland e Ken Schwaber em 1995 [31], e apresentado na conferência Ospsla em Austin no Texas. Neste mesmo ano foi publicado o artigo "SCRUM Software Development Process" [32].

Os autores herdaram o termo "Scrum" do artigo "The New Product Development Game", publicado por Takeuchi e Nonaka em 1986 [33].

Uma das ideias do Scrum é que a equipe de trabalho seja auto gerenciável, hora denominada "*Squad*", composta geralmente pelos perfis profissionais de: Product Owner (P.O) ou "Dono do Produto", facilitador e a equipe de desenvolvimento com seus respectivos desenvolvedores e analistas [34].

A *squad* é um modelo ágil de organização de times de trabalho e surgiu justamente para encurtar a comunicação entre os integrantes e melhorar o processo de trabalho.

Uma *squad*, além da sua composição de profissionais multidisciplinares, possui autonomia para tomar decisões [34], desde que alinhadas com os objetivos organizacionais, como também desenhar projetos.

Uma figura de grande responsabilidade na *Squad* é o P.O, que é o profissional responsável pelos requisitos do produto de software [34].

Além de prover os requisitos, o P.O é responsável pelo "*Backlog* do Produto" [34], que é uma espécie de repositório abstrato que contém todos os requisitos funcionais do produto de software.

Outra figura importante dentro das *squads* é a do Facilitador que pode ser exercido por qualquer pessoa [34], mas geralmente é ocupado por um membro da equipe de desenvolvimento.

O facilitador irá coordenar e facilitar algumas das cerimônias essenciais no processo ágil como: Concepções/*Inceptions*, Refinamentos, Planejamentos, Re*views* e Retrospectivas.

Além do PO e facilitador, as *squads* são compostas pelo "time de desenvolvimento" que são todos os responsáveis pela produção dos incrementos de software, como analistas e programadores.

Para facilitar ainda mais o trabalho por *squad*, a organização pode ter grupos maiores conhecidos como "*Tribes*", "*Chapter*" e "Guilds" [34].

- ***Tribes***: são vários times de *squads* trabalhando com objetivos comuns ou semelhantes. Exemplo: existe um grande projeto: "Conta corrente" e dentro dele alguns produtos como "Cheques", "Pix", "abertura de conta" etc. Para cada um dos produtos, podem ser criadas *squads* específicas.
- ***Chapter***: é formado por profissionais da mesma área de conhecimento, exemplo: designer, arquitetos, facilitadores, homologadores etc., com a mesma função, porém, que fazem parte de *squads* diferentes. A ideia é poder disseminar conhecimento e auxiliar no fluxo de trabalho.
- ***Guids***: grupos formados por pessoas de qualquer área, mas que se interessam por algum assunto em comum e, por isso, se juntam para manter uma comunicação.

O Scrum Framework separa o desenvolvimento de software por "*Sprints*" que são eventos por tempo definido que devem durar em duas a quatro semanas, sendo elas separadas em três cerimônias principais: planejamento, revisão e retrospectiva, como também em reuniões diárias de duração de minutos denominas "*dailys*", conforme figura 11:

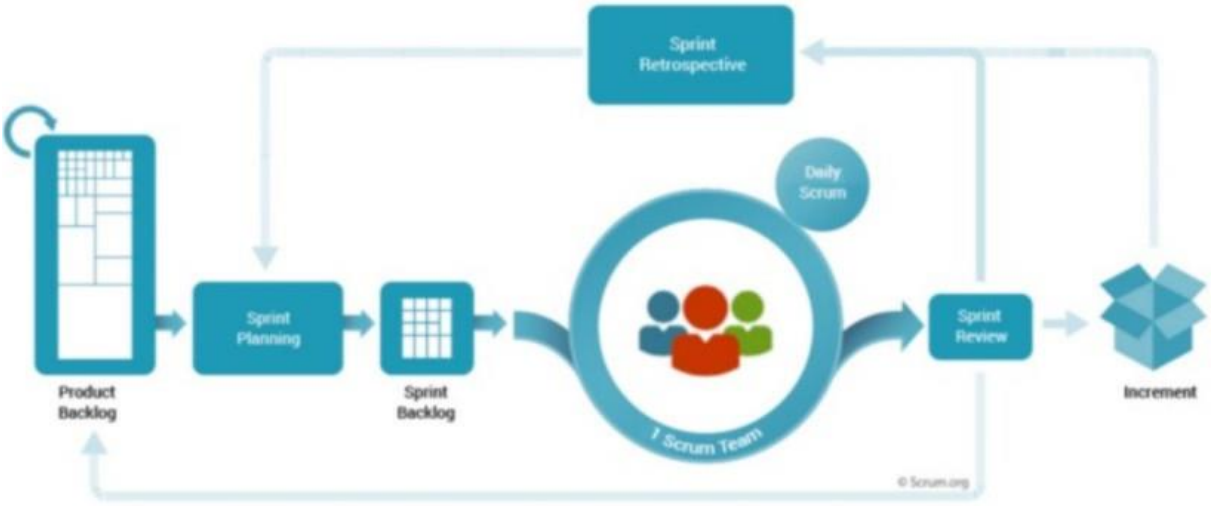


Figura 11 Scrum Esquematizado. Fonte: Scrum org [34]

No Scrum Esquematizado, a primeira fase contém o *Product Backlog*, onde estão contidos todos os requisitos funcionais do software que está sendo desenvolvido [34]. Geralmente são identificados através de reuniões de concepção do projeto, onde o P.O irá agrupá-los.

Para melhor entendimento das cerimônias do Scrum Framework, foi elaborada a figura 12:

Figura 12 Cerimônias do Scrum. Fonte: Próprio Autor, adaptado de Scrum org [34]

A primeira cerimônia é a "sprint planning" ou "planejamento", que é uma reunião utilizada para realizar o planejamento da próxima sprint a ser tratada [34], sendo que

suas entradas são os requisitos identificados nas reuniões de concepção ou refinamento do produto, além das ações geradas nas retrospectivas de outras *sprints*.

O planejamento da sprint possui uma participação mais forte do P.O, que deve apresentar "o que deve ser desenvolvido" [34], como também da equipe de desenvolvimento que irá metrificar o "como deve ser desenvolvido".

A planning é o momento de haver o alinhamento entre o P.O e o time de desenvolvimento sobre o que será entregue no próximo ciclo de desenvolvimento.

Se foi realizado um bom refinamento, essa cerimônia de planejamento será mais dinâmica e simples [34]. Caso contrário, o tempo poderá ser maior, tendo em geral uma média de 4 a 8 horas para sua execução.

Uma cerimônia importante, e que acontece no meio das *sprints*, são as *dailys* [35].

Que pequenos "pontos de controles" com a duração de 15 minutos, onde cada membro da *squad* tem a oportunidade de informar o que está fazendo, o progresso dessa atividade e se possui algum impedimento.

Caso haja algum impedimento, o Facilitador da *squad* terá por objetivo removê-lo

para que, o membro da equipe ao que reportou tal impedimento, consiga trabalhar [34].

Antes do encerramento da sprint, é realizada a cerimônia de "review" ou revisão, que é o momento em que os desenvolvedores do time têm a oportunidade de apresentar para o P.O, o incremento de software [34], fruto da sprint em questão.

A reunião de revisão tem uma duração média de duas a quatro horas e é apresentada pela equipe de desenvolvimento.

Caso haja melhorias, essas serão anotadas para que sejam incluídas nos planejamentos das próximas *sprints*, caso o P.O deseje.

Para finalizar a sprint, é realizada a cerimônia de retrospectiva, que possui uma du-

ração média de duas horas, é conduzida pelo Facilitador da *squad* e tem por objetivo identificar os pontos de ajustes a serem melhorados para a próxima sprint [34], como também reportar os itens que foram sucesso e podem continuar.

Através da identificação dos modelos de engenharia e reegenharia de software sejam ágeis ou tradicionais, foi possível traçar princípios a serem seguidos para as próximas atividades.

Na próxima seção será apresentado uso da Modelagem de Processos na Reengenharia de Software.

2.1.5 Modelagem de processos

Para auxiliar no processo de reengenharia de software, é possível utilizar alguns métodos de documentação do trabalho intelectual, como por exemplo o Business Process Model and Notation (BPMN) [36], que permite, através da notação em processos, documenta o fluxo atual de reengenharia de software ou até mesmo permitir redesenho dele de forma melhorada.

Muitas organizações usam modelos de processos de negócios (business process management BPM) para documentar operações de negócios e para formalizar esses requisitos em projetos de engenharia de software. [37].

O uso do BPM tem por intuito conhecer e explorar os processos que são realizados numa organização, mensurá-los e dispor de ciclos de melhoria contínua para evolução da organização [38].

O CBOK® [39] recomenda que a gestão proposital de processos cria práticas de negócios mais fortes que levam a processos mais eficazes, maiores eficiências, mais agilidade e, finalmente, maior retornos sobre os investimentos das partes interessadas.

Nessa linha, o mapeamento de processos consiste no levantamento e registro do fluxo de informações presentes dentro dos processos existentes na organização ("AS-IS"), fornecendo uma visão detalhada dos processos em questão, mostra a relação entre as entradas as saídas e as tarefas executadas [38].

Existem várias formas de externalizar e documentar o processo intectual para modelagens de processos de negócios, como por exemplo:

- Business Process Model and Notation (BPMN): Padrão criado pelo Object Management Group (OMG®) [24], últil para apresentar um modelo para públicos-alvo diferentes;
- Fluxograma: sendo originalmente aprovado como padrão American National Stan- dards Institute (ANSI), inclui um conjunto simples e limitado de símbolos não padronizados;
- Event-driven Process Chain (EPC): Considera eventos como gatilhos para uma estapa do processo;
- Unified Modeling Language (UML): mantido pela OMG® [24], consite em um cojunto padrão de notações técnicas de diagramação orientado à descrição de requisitos de sistemas de informação.

Existe uma tendência de aumento da importância da modelagem. Linguagens de especificação como UML e BPMN são aplicadas na maioria dos projetos de engenharia

de software atualmente [40]. Todavia, cada uma possui aplicação em etapas específicas do processo de engenharia de software.

Ao estudar algumas obras sobre o uso do BPM em processos de reengenharia de software [37] [41] [42], identificou-se que uma das notações mais utilizadas é o BPMN, quando o assunto está focado modelagem do processo de software com um todo.

Lisboa [38] propõe uma metodologia para a implantação da modelagem de processo numa organização que é dividida em quatro fases:

- Planejamento:
    - Identificar os objetivos do projeto;
    - Definir padrões de modelagem e ferramentas de apoio;
    - Elaborar plano de execução.
- Mapeamento e Análise:
    - Identificação e Mapeamento do Processo (AS-IS);
    - Modelagem do processo (TO-BE).
    - Implementação:
    - Elaboração do Plano de Ação;
    - Implantação das Melhorias.
- Monitoramento e Controle:
    - Avaliação do desempenho;
    - Melhoria continua.

A proposta dessa pesquisa em questão é um pouco mais simplificada, sugerindo a modelagem do processo atual (AS-IS) de reengenharia de software e o redesenho do processo melhorado (TO-BE).

Na próxima seção será apresentada a Gestão de Riscos.

2.2 Gestão de Riscos

Conforme a ISO 31000 [9], gestão de riscos refere-se à arquitetura (princípios, estrutura e processo) para gerenciar riscos eficazmente, enquanto gerenciar riscos refere-se à aplicação desta arquitetura para riscos específicos.

Já o risco em si, pode ser considerado qualquer coisa que impacte os objetivos de uma organização [9].

Todas as organizações de todos os tipos e tamanhos enfrentam influências e fatores internos e externos que tornam incerto se e quando elas atingirão seus objetivos [9].

No contexto dessa pesquisa, o objetivo está relacionado às expectativas de qualidade, bom funcionamento do software e sua conformidade com requisitos legais e institucionais, além de seu desenvolvimento ser cumprido dentro do prazo e orçamento estipulado.

O PMBoK® 6ª Edição [43] mostra que a atitude das organizações e das partes interessadas em relação aos riscos pode ser influenciada por um número de fatores, que são classificados de forma ampla em três tópicos:

- Apetite de risco: que é o grau de incerteza que uma entidade está disposta a aceitar, na expectativa de uma recompensa;
- Tolerância a riscos: que é o grau, a quantidade ou o volume de risco que uma organização ou um indivíduo está disposto a tolerar;
- Limite de riscos: que se refere às medidas ao longo do nível de incerteza ou nível de impacto no qual uma parte interessada pode ter um interesse específico. A organização aceitará o risco abaixo daquele limite. A organização não tolerará o risco acima daquele limite.

Quando um risco é negativo, torna-se uma ameaça para o ativo em questão. Quando positivo, torna-se uma oportunidade [9].

A ameaça pode ser qualquer coisa ou evento que prejudique o ativo causando um impacto negativo ao negócio da instituição [9], como por exemplo: incompatibilidade de sistemas, falhas de software em ambiente de produção, rotatividade de recursos humanos, falta de requisitos de software, etc.

Assim como todo indivíduo vive sujeito a ameaças de todas as formas, as organizações também passam por estes riscos e podem ser classificadas como: naturais, involuntárias ou voluntárias [44].

- Involuntárias: ocorrem por meio de acidentes. No contexto de reengenharia de software, por exemplo, pode estar associado à falta de conhecimento de um desenvolvedor em determinado sistema que está sendo migrado, possibilitando a consequência de mal funcionamento do software em produção.
- Naturais: estão relacionados a fenômenos que fogem ao controle humano, ou seja, é um fator da natureza, como por exemplo: um raio que pode atingir um prédio causando falhas nos datacenters com servidores de bancos de dados e aplicativos de software.

- Voluntárias: acontecem com o propósito de destruir o ativo ou prejudicar a organização de alguma forma. Fator este que pode ser causado por pessoas mal-intencionadas ou até métodos utilizados por essas, como códigos nocivos.

Um dos motivos para que o risco seja consumado, ou seja, concretizar-se, é através da exploração de falhas ou vulnerabilidades, sendo essas a ausência de determinado controle [45].

As vulnerabilidades são um conjunto de falhas ou fatores internos que podem ser exploradas pelas ameaças, com potencial de ocasionar um incidente, ou seja, um evento indesejado [46].

A gestão de riscos proativa deve ser aplicada com o princípio de proteger os ativos organizacionais, que conforme a ISO 13335 (2004) [11] é tudo aquilo que representa valor para uma organização, como por exemplo: Pessoas, Processos, Ambiente, Tecnologia, Informação e Comunicação.

Numa visão de apoio à organização, de acordo com ISO 31000 (2018) [9], ao realizar a gestão de riscos proativa é possível:

- aumentar a probabilidade da organização de atingir os seus objetivos;
- melhorar a identificação de oportunidades e ameaças;
- atender às normas internacionais e requisitos legais e regulatórios pertinentes;
- melhorar o reporte das informações financeiras;
- melhorar a governança;
- melhorar a confiança das partes interessadas;
- estabelecer uma base confiável para a tomada de decisão e planejamento;
- melhorar os controles; melhorar a eficácia e a eficiência operacional;
- melhorar a prevenção de perdas e a gestão de incidentes; aumentar a resiliência da organização.

Como uma das formas de resolução desses problemas, a ISO 31000 [9] propõe um processo de gestão de riscos, conforme figura 13:

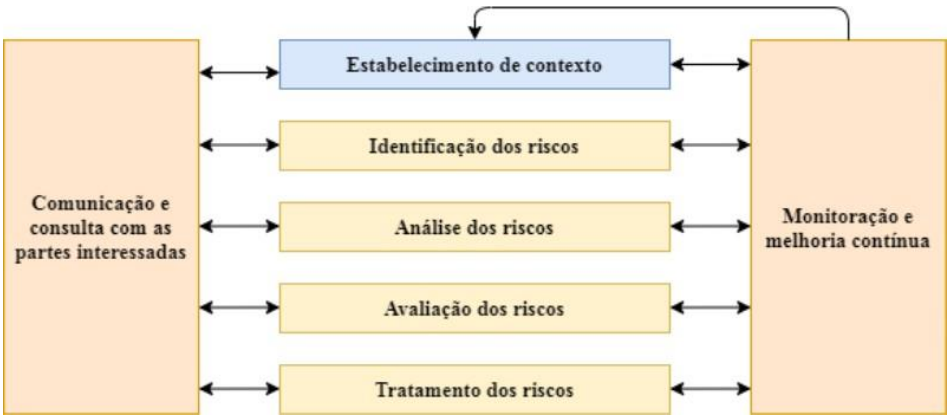


Figura 13 Processo de Gestão de Riscos. Fonte: ISO 31000 [9]

O processo de gestão de riscos é a aplicação sistemática de políticas, procedimentos e práticas de gestão para as atividades de comunicação, consulta, estabelecimento do contexto, e na identificação, análise, avaliação, tratamento, monitoramento e análise crítica dos riscos [47].

2.2.1 Estabelecimento do contexto

O estabelecimento do contexto define os parâmetros básicos para a gestão de riscos, o escopo e os critérios para o resto do processo [9].

No estabelecimento do contexto, os objetivos do processo de avaliação de riscos, critérios de riscos e o processo de avaliação de riscos são determinados e acordados [44].

Conforme TCU [48], para estabelecer o contexto, pode-se seguir os seguintes passos:

- identificar quais objetivos ou resultados devem ser alcançados;
- identificar os processos de trabalho relevantes para o alcance dos objetivos/resultados;
- identificar as pessoas envolvidas nesses processos e especialistas na área;
- mapear os principais fatores internos e externos que podem afetar o alcance dos objetivos/resultados (pessoas, sistemas informatizados, estruturas organizacionais, legislação, recursos, *stakeholders* etc.).

O quadro 2.1 apresenta o contexto externo e interno de forma clara:

Quadro 1 Definição de contexto interno e externo. Fonte: Adaptação de TCU [48].

Contexto Externo	Contexto Interno
Deve-se entender o contexto externo da organização para assegurar que as preocupações das partes interessadas externas sejam consideradas no desenvolvimento docritério de risco. *Deve incluir:* os fatores culturais, políticos, legais, regulatórios, financeiros, econômicos e ambientais competitivos, seja em nível internacional, nacional, regional ou local; fatores–chave e tendências que tenham impacto sobre os objetivos da organização; percepções e valores das partes interessadas externas.	É o ambiente interno no qual a organização busca atingir seus objetivos, alinhados com a cultura, processos, estruturas e estratégias da organização, sendo algo dentro da organização que pode influenciar a maneira pela a ela gerenciar os riscos. *Deve incluir:* capacidades da organização em termos de recursos e conhecimento; fluxos de informação e processos de tomada de decisão; partes interessadas internas; objetivos e das estratégias que estão em vigor a fim de atingi–los; percepções, valores e cultura; políticas e processos; normas e modelos de referência adotados pela organização; estruturas (por exemplo, governança, papéis e responsabilizações.

O quadro 2.1 exemplifica basicamente a compreensão do ambiente externo e interno no qual o objeto de gestão de riscos se encontra inserido e em identificar parâmetros e critérios a serem considerados no processo de gestão de riscos [48].

Tanto o contexto externo e interno deve ser definido como premissa no processo de gestão de risco e servirão como entradas para os demais processos, pois irão impactar diretamente na construção desses.

2.2.2 Identificação dos Riscos

Após o estabelecimento do contexto, se faz necessária a identificação dos riscos, ou seja, encontrar, reconhecer e registrar os riscos [44].

Essa etapa deve ser baseada na identificação das fontes de riscos, áreas de impactos, eventos (incluindo mudanças nas circunstâncias) e suas causas e consequências potenciais [12].

Esse processo busca responder perguntas como:

- O que pode atrapalhar o alcance dos objetivos ou resultados organizacionais?
- Quais são os eventos que podem afetar de alguma forma o sucesso de um determinado objetivo?
- Quais são as fontes dos riscos que permeiam os ativos organizacionais?

A saída do processo de identificação de riscos, é uma listagem abrangente e detalhada de todos os potenciais riscos que possam criar, aumentar, evitar, reduzir, acelerar ou atrasar o cumprimento de determinado objetivo [9].

O TCU [49] propõe passos que devem ser seguidos para identificação dos riscos, conforme figura 14:

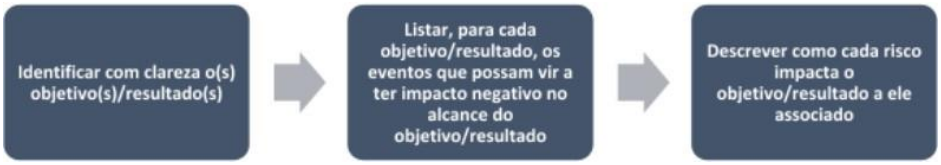


Figura 14 Identificação de riscos. Fonte: Próprio Autor, adaptado de TCU [48]

Conforme a figura 14, entende-se que o processo de identificação de risco é crítico [43], tendo em vista que um risco não mapeado não poderá ser analisado posteriormente [9], o que pode impactar diretamente no insucesso da gestão de riscos proativa como um todo.

Se faz necessária a identificação de todas as fontes de riscos, sendo estas sob o controle ou não da organização, mesmo que essas não sejam tão evidentes.

Ao inventariar um risco identificado, deve-se documentar o impacto ou consequência dele, caso venham a se concretizar tornando-se um evento/incidente [9].

Além de disso, é importante identificar e documentar o que pode acontecer, possíveis causas e cenários que mostram quais consequências esse incidente pode trazer para a organização.

Para o processo de identificação dos riscos, se faz importante o envolvimento de pessoas que possuam domínio sobre o assunto de gestão proativa de riscos [48].

Existem várias técnicas propostas para identificação de riscos como: *Brainstorming*, *brainwriting*, entrevistas, visitas técnicas, pesquisas etc., [43] [12] e convém que elas sejam adequadas aos seus objetivos e capacidades e aos riscos enfrentados [9].

Uma vez que um risco é identificado, convém que a organização identifique quaisquer controles existentes, tais como funcionalidades projetadas, pessoas, processos e sistemas, que possam ser aplicados a esses riscos [50], de forma a modificá-los de alguma forma [9] [43].

2.2.3 Análise de riscos

A análise do risco se refere ao desenvolvimento da compreensão sobre o risco e à determinação do nível do risco [48], como também a medição da probabilidade do risco se tornar um incidente e as consequências associadas [50]. Ela fornece uma entrada para o processo de avaliação de riscos e às decisões sobre se os riscos necessitam ser tratados.

As consequências (impacto) e suas probabilidades são combinadas para determinar um nível de risco (escore de risco), sendo a probabilidade a chance de o evento ocorrer dentro do prazo previsto para se alcançar o objetivo [48] e a consequência, os impactos que o risco trará para a organização caso aconteça [50].

Devem ser identificados também os processos críticos de maior vulnerabilidade e as ameaças associadas a eles [51], os quais serão diminuídos a cada implementação de controle e formas de proteção ao ativo em cada momento.

A análise dos riscos organizacionais é contínua e exige uma monitoração constante das ameaças que possam afetar os objetivos institucionais [51] e deve considerar os resultados obtidos através da implementação dos controles em ciclos anteriores [49] de forma que seja possível validar sua eficácia, melhorar ou substituir esses controles.

Os impactos podem ter uma baixa consequência, porém alta probabilidade, ou uma alta consequência e baixa probabilidade, ou algum resultado intermediário.

Três abordagens gerais são comumente empregadas para estimar a probabilidade; elas podem ser utilizadas individual ou conjuntamente, conforme quadro 2:

Quadro 2 Definição de contexto interno e externo. Fonte: Adaptação de ISO 31010 [12].

A utilização de dados históricos	**Previsões de probabilidade**	**A opinião de especialistas**
Pertinentes para identificar eventos ou situações que ocorreram no passado e, assim, capazes de extrapolar a probabilidade de sua ocorrência no futuro. Convém que os dados utilizados sejam pertinentes ao tipo de sistema, instalação, organização ou atividade que está sendo considerado também às normas operacionais da organização envolvida.	Utilizando técnicas preditivas tais como análise de árvore de falhas e análise de árvore de eventos. Quando os dados históricos forem indisponíveis ou inadequados, é necessário deduzir a probabilidade pela análise do sistema, atividade, equipamento ou organização e seus estados bemsucedidos ou com falha associados.	Pode ser utilizada em um processo sistemático e estruturado para estimar a probabilidade. Convém que os julgamentos dos especialistas recorram a todas as informações pertinentes disponíveis, incluindo informações históricas, específicas do sistema, específicas da organização, experimentais, de projeto etc.

Se historicamente há uma frequência muito baixa de ocorrência, então qualquer estimativa da probabilidade será muito incerta [12].

Caso sejam utilizadas técnicas preditivas, é importante assegurar que a devida consideração tenha sido efetuada na análise para a possibilidade de modos de falha em comum envolvendo a coincidência de falha de um número de partes ou componentes diferentes dentro do sistema, resultantes da mesma causa [12].

E no caso da opinião do especialista, como pode ser subjetiva, existem diversos métodos formais para induzir o seu julgamento que fornecem um auxílio para a formulação das questões apropriadas, classificação de categorias e julgamentos de probabilidade absoluta [12].

2.2.4 Avaliação de riscos

A avaliação de riscos consiste em comparar os níveis estimados de risco com critérios de risco definidos quando o contexto foi estabelecido [48], a fim de determinar a significância do nível e do tipo de riscos.

O limite de exposição a riscos representa o nível de risco acima do qual é desejável o tratamento do risco [9]. Espera-se que, com os resultados do tratamento, o nível de risco residual fique abaixo do limite de exposição.

A avaliação de riscos utiliza a compreensão do risco, obtida durante a análise de riscos, para tomar decisões sobre as ações futuras.

Considerações éticas, legais, financeiras e outras, incluindo as percepções do risco, são também dados de entrada para a decisão [9].

As decisões na avaliação de um risco podem incluir, conforme figura 15:

Figura 15 Decisões na Avaliação de Riscos. Fonte: Adaptação de ISO 31000 [9]

A natureza das decisões que necessitam ser tomadas e os critérios que serão utilizados para tomar essas decisões foram decididos no estabelecimento do contexto [12] [48] [52], mas precisam ser revistos em mais detalhes nesta fase, agora que se sabe mais sobre os riscos identificados em particular.

A estrutura mais simples para a definição dos critérios de risco é um nível único que divide os riscos que necessitam de tratamento daqueles que não necessitam [53]. Isso fornece resultados atrativamente simples, porém não reflete as incertezas envolvidas na estimativa de riscos e na definição da fronteira entre aqueles que necessitam de tratamento e aqueles que não necessitam.

A decisão sobre se como tratar o risco pode depender dos custos e benefícios de assumir o risco e os custos e benefícios da implementação de controles melhorados.

Uma abordagem comum é dividir os riscos em três faixas [12] [48]:

- Uma faixa vermelha: onde o nível de risco é considerado intolerável quaisquer que sejam os benefícios que possam trazer à atividade, e o tratamento de risco é essencial qualquer que seja o seu custo;
- Uma faixa amarela (ou área "cinzenta"): onde os custos e benefícios são levados em consideração, e oportunidades são comparadas com potenciais consequências;
- Uma faixa verde: onde o nível de risco é considerado desprezível ou tão pequeno que nenhuma medida de tratamento de risco seja necessária.

A avaliação dos riscos fornece subsídios para a tomada de decisão, não se constituindo em fator determinante para eventual tratamento do risco. Ou seja, cabe ao gestor, diante da lista de riscos ordenados por nível de risco, decidir quais merecem ações mitigadoras.

2.2.5 Tratamento de riscos

Compreende o planejamento e a realização de ações para modificar o nível de risco.

O nível de risco pode ser modificado por meio de medidas de resposta ao risco que mitiguem, transfiram ou evitem esses riscos [49].

O tratamento dos riscos deve seguir os seguintes passos descritos na figura 2.16:

Figura 16 Passos para o Tratamento dos Riscos. Fonte: Adaptado de CGU [49]

Completado um processo de avaliação de riscos, o tratamento de riscos envolve selecionar uma ou mais opções pertinentes para alterar a probabilidade de ocorrência [43], o efeito dos riscos, ou ambos, e a implementação destas opções.

Isto é acompanhado por um processo cíclico de reavaliação do novo nível de risco, tendo em vista a determinação de sua tolerabilidade em relação aos critérios previamente definidos, a fim de decidir se tratamento adicional é requerido [49].

A identificação das medidas de resposta ao risco, assim como a identificação de riscos, deve ser realizada em oficinas de trabalho ou, conforme o caso, pelo próprio gestor do risco [43], com a participação de pessoas que conheçam bem o objeto de gestão de riscos. As medidas mitigadoras podem envolver, por exemplo, a adoção de controles, o redesenho de processos, a realocação de pessoas, a realização de ações de capacitação, o desenvolvimento ou aperfeiçoamento de soluções de TI, a adequação da estrutura organizacional, entre outros.

2.2.6 Tratamento de riscos

De acordo com a ISO 31000 [9], controles são medidas que modificam o risco de alguma forma, podendo por exemplo, diminuir a probabilidade do risco se concretizar.

A figura 17 representa quais as possíveis modificações nos riscos através da aplicação de controles:

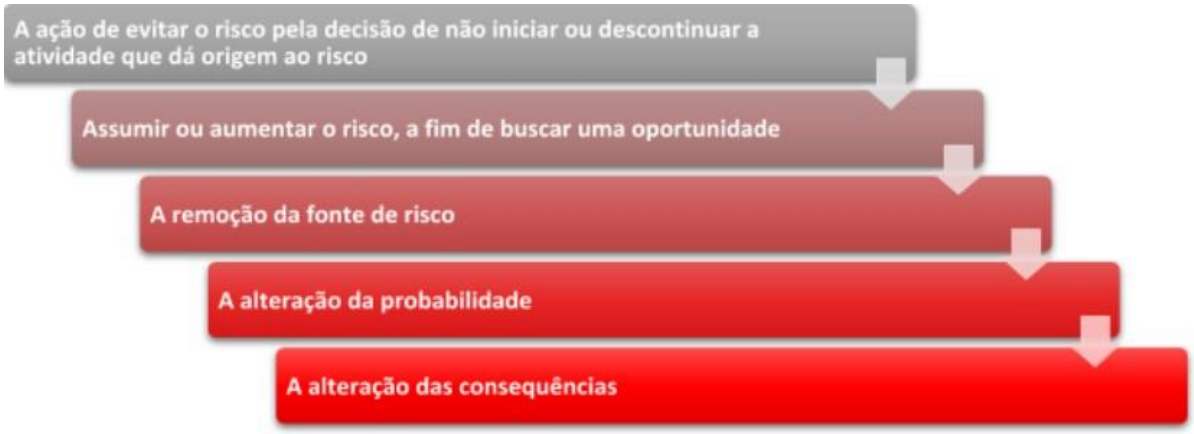


Figura 17 Modificações nos Riscos. Fonte: Adaptado de ISO 31000 [9]

Mesmo que não seja possível extinguir os riscos organizacionais por completo, existe a possibilidade de mitigá-los de alguma forma através da aplicação dos controles, conforme mostra a figura 17.

Dessa forma, entende-se que um controle não aplicado pode ocasionar brechas ou vulnerabilidades que poderão ser exploradas por determinada ameaça, causando então um impacto negativo ao negócio [46].

O controle pode ser qualquer processo, política, dispositivo ou ação a ser tomada, com o objetivo de modificar o risco [12].

Para auxílio na atividade de identificação de riscos de reengenharia de software, podem ser utilizados controles que fomentem um bom desenvolvimento de software como, por exemplo, a ISO 25010 [5] que propõe um modelo de qualidade de produto de software conforme apresentado no item de qualidade de software.

Numa situação do cotidiano, podemos verificar, por exemplo, que um sistema migrado através do processo de reengenharia de software, que não possui controles de segurança da informação, como algoritmos que garantam a confidencialidade dos dados [54], exemplo: através da criptografia.

Nessa situação, existem vulnerabilidades que podem ser exploradas por uma ameaça de hackers que esteja monitorando a rede e deseja capturar, por exemplo, dados de transações bancárias que não foram criptografadas.

Outro exemplo, é um sistema bancário que está migrado não possuir compatibilidade com outros softwares necessários, ocasionando consequências como: duplicação de lançamentos de créditos, por exemplo.

Para um melhor entendimento, foi elaborado o seguinte quadro que exemplifica alguns tipos de controles conforme quadro 3:

Quadro 3 Possíveis Tipos de Controles. Fonte: Próprio Autor.

Tipos de Controles	**Obrigatoriedade**	**Exemplo**	**Fonte**
Leis	Sim, dependendo do tipo de ativo e seguimento da organização.	Possui caráter punitivo caso não seja seguido.	Lei geral de proteçãoaos dados pessoais (LGPD) [55].
Normas	Sim, dependendo do tipo de ativo e seguimento da organização.	Pode possuir caráter punitivo caso não seja seguido. Se seguida aumenta o nível de conformidade da organização.	ISO/IEC 25010 *System and software quality model*s [5].
Boas Práticas	Não é obrigatório.	Segmento de mercado, difundido, testado e aprovado por entidades mantenedoras que pode auxiliar a organização.	Capability Maturity Model® Integration (CMMI ®) [56]; Melhoria no processo de software no Brasi (MPsBR) [57]
Benchmarking	Não é obrigatório.	Métodos de sucesso no mercado que auxiliam a organização a mitigar seus riscose aumentar as chances de atingir seus objetivos.	*Lean Inception* [26]; *Lean Startup* [58].
Controles Internos	São Obrigatórios.	São controles de dentro da organização, padrões de qualidade.	Auditoria Interna

Uma das características do controle, é que nem sempre sua aplicação é facultativa. Isso se dá porque algumas empresas, como instituições financeiras, por exemplo, possuem leis, normativos ou normas que regem o seu funcionamento.

Com isso, a própria inexistência de determinado controle pode ocasionar em consequência de perdas financeiras, interrupção indesejada de projetos e até mesmo multas, como prevê a Lei Geral de Proteção aos Dados Pessoais [55], que pode aplicar de multa de 2% sobre o faturamento líquido de organizações que tiverem vazamento de dados pessoais de seus clientes.

Dessa forma, entende-se que é necessária a criação de uma base de conhecimento com controles aplicados aos que foram identificados, forma de resposta e tratamento a riscos.

2.2.7 Monitoramento e análise crítica

Compreende o acompanhamento e a verificação do desempenho ou da situação de elementos da gestão de riscos, podendo abranger a política, as atividades, os riscos, os planos de tratamento de riscos, os controles e outros assuntos de interesse [48].

O processo de avaliação de riscos destaca o contexto e outros fatores que se pode esperar que variem ao longo do tempo e que poderiam alterar ou invalidar o processo de avaliação de riscos. Convém que estes fatores sejam especificamente identificados para o contínuo monitoramento e análise crítica, de modo que o processo de avaliação de riscos possa ser atualizado quando necessário [12].

Convém também que os dados a serem monitorados para refinar o processo de avaliação de riscos sejam identificados e coletados [12].

Convém que a eficácia dos controles também seja monitorada e documentada a fim de fornecer dados para uso na análise de riscos. Convém que as responsabilidades para a criação e análise crítica das evidências e da documentação sejam definidas [12].

2.2.8 Melhoria contínua

Compreende o aperfeiçoamento ou ajuste de aspectos da gestão de riscos avaliados no monitoramento.

A melhoria contínua pode ser entendida em duas dimensões: uma relativa ao próprio Sistema de Gestão de Riscos da organização e outra relacionada aos resultados do monitoramento sobre a efetividade do tratamento do risco, a cargo dos gestores de riscos [12].

2.2 Gestão de Riscos

A evolução dos negócios de uma empresa ao longo dos anos requer a evolução sincronizada de sua tecnologia legada, porém, tais aplicações devem sempre oferecer um nível de qualidade adequado [59].

Essa migração acontece porque a organização tem interesse em manter as regras negociais, funções e aplicações do sistema legado em uma nova plataforma, corrigir erros conhecidos ou simplesmente não existe mais suporte para ela.

Para evoluir essa tecnologia se faz necessária a reengenharia de software, que está focada na migração de um sistema legado [2], que como apresentado, é um ativo de valor para a organização.

A migração de aplicações legadas que visam a adequação com novas tecnologias é uma boa solução para que as organizações mantenham seus aplicativos críticos, ao invés de construir um novo sistema desde o início, o que geralmente envolve mais riscos e um processo mais complexo e trabalhoso [59].

Alguns autores explicam que o sucesso dos projetos de software depende da identificação dos riscos do projeto e do gerenciamento dos riscos de maneira proativa [60] [44] [61].

O gerenciamento de risco requer insights completos sobre a interrelação de vários fatores de risco para propor estratégias para minimizar a taxa de falha [60].

Uma migração de software por si só é arriscada e idealmente requer atenção adicional por meio de cuidados revisão completa do código, design e possivelmente mais testes. Todavia, essa visão de risco na mudança de software varia de acordo com a visão de cada equipe de desenvolvimento [62].

Partindo da premissa que os riscos são as incertezas sob os objetivos [9], em projetos de reengenharia de software, caso esses riscos se concretizem, eles estarão impactando diretamente na conclusão do projeto trazendo consequências como: aumento de custos, atraso no cronograma, aumento de escopo e a baixa qualidade do produto de sistema migrado.

O procedimento para análise de risco é um dos procedimentos importantes durante o desenvolvimento do sistema de informação. No entanto, esse procedimento geralmente não é realizado de maneira adequada [63].

Vale ressaltar que o gerenciamento de riscos é um trabalho muito complexo e crítico no desenvolvimento de projetos de software [64].

Em revisão a literatura é possível encontrar alguns modelos propostos que podem apoiar o gerenciamento de riscos no processo de gerenciamento de riscos.

Kumar et al. [64] propõe o uso de modelo de rede Bayesiana, que modela as relações causais de um sistema ou conjuntos de dados, para representar a probabilidade de um risco de concretizar em projetos de desenvolvimento de software.

Esse modelo considera o uso da rede Bayesiana apenas como método para a avaliação dos riscos [64], não abarcando, por exemplo, a identificação dos fatores de riscos.

Para Sadiq e Shahid [65] o gerenciamento de riscos começa com a capacitação da equipe envolvida para conscientização e treinamento nos detalhes do paradigma de gestão de risco e recomendam que seja realizada a identificação dos riscos que podem comprometer o sucesso do projeto.

No trabalho de Sadiq e Shahid [65] eles propõem o processo de gerenciamento de riscos em projetos de desenvolvimento de software dividido em seis fases:

1. A identificação dos riscos e produção de uma lista de itens específicos que podem comprometer o sucesso do projeto;
2. Análise e avaliação da probabilidade de perda e magnitude para cada item de risco;
3. Priorização e classificação dos riscos;
4. Planejamento do gerenciamento de riscos;
5. Tratamento ou resolução dos riscos, incluindo abordagens de simulação com protótipos e benchmark;
6. Monitoração dos riscos para rastrear o progresso do projeto para resolver seus itens de riscos e tomada de ação corretiva de forma cada vez mais apropriada.

Já para Shabid et al. [62] uma das abordagens que mais apresentou eficácia sobre os riscos na migração de software foi a classificação desses riscos pela própria equipe de desenvolvimento, que possui seus próprios fatores que modelam esses riscos.

Kendrick [66] no contexto de sua obra mostra que o PMBoK® [43] possui ferramentas que podem ser utilizadas para gerenciamento de riscos.

Todavia, essa pesquisa possui uma abordagem um pouco diferentes das anteriores, focando especificamente na reengenharia de software e propondo um instrumento computacional que pudesse auxiliar os gestores do projeto no gerenciamento dos riscos do processo.

Essa pesquisa em questão propõe realizar a identificação, análise e avaliação dos riscos, como também a modelagem de processo melhorado (TO-BE) de reengenharia de software como respostas a esses riscos.

Capítulo 3

Metodologia de Pesquisa

O termo metodologia vem do grego da junção das palavras “meta” que é “ao largo”; “odos”, caminho e “logos” que é discurso ou estudo [67] e é um conjunto de procedimentos sistemáticos, baseados no raciocínio lógico, que tem por objetivo encontrar soluções para os problemas propostos mediante o emprego de métodos científicos [68].

Já método é o caminho em direção a um objetivo e metodologia é o estudo do método, ou seja, é o corpo de regras e procedimentos estabelecidos para realizar uma pesquisa, a qual compreende o conjunto de conhecimentos precisos e metodicamente ordenados em relação a determinado domínio do saber [69].

Silveira et al. [69] recomenda que a pesquisa deva ser estruturada conforme sua modalidade inerente ao objetivo da mesma. Dessa forma, o método de pesquisa deste trabalho foi arquitetado conforme figura 18:

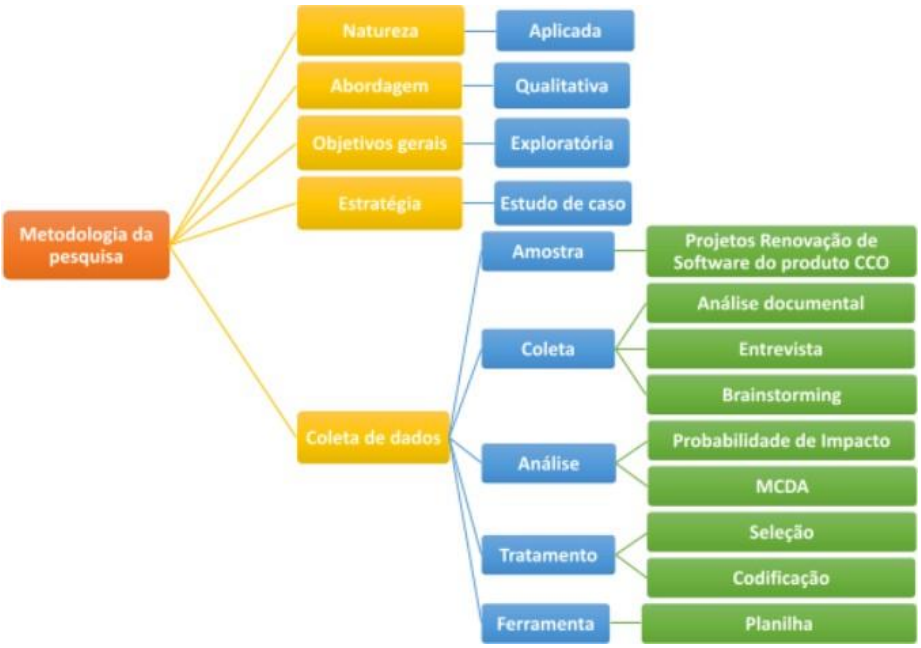


Figura 18 Método de Pesquisa. Fonte: Próprio Autor

Esse trabalho em questão busca estabelecer um instrumento para gestão de riscos no processo de reengenharia de software que será aplicado in-loco no projeto denominado Renovação de Software do Produto Conta Corrente e Benefícios da Previdência Social da CredTec.

Dessa forma, a natureza da pesquisa escolhida foi a aplicada.

De acordo com Silveira et al. [69], a pesquisa aplicada busca gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos. Envolve verdades e interesses locais.

Silva [70] defende que a pesquisa quantitativa utiliza a matemática para realizar a análise de dados e interpretação dos resultados colhidos para fazer a correlação da realidade empírica com a teoria que embasa o estudo.

Dessa forma a natureza dessa pesquisa é elencada como qualitativa, tendo em vista que será necessário colher os dados das vulnerabilidades e ameaças no processo de reengenharia de software através de todo levantamento documental, análise in loco e identificação dos dados por meio de técnicas de obtenção de informações.

Uma vez sendo feito isso, os dados serão analisados na perspectiva de gestão de riscos através de ferramentas de técnicas matemáticas para cálculo de probabilidade e impacto dos riscos no processo.

Dessa forma esse trabalho em questão busca identificar na literatura normas, boas práticas, frameworks e casos de sucesso que possam servir de controles para combater as vulnerabilidades e ameaças ao processo de reengenharia de software, sendo possível estabelecer um instrumento de gestão de riscos para ele aplicando os conhecimentos na prática in-loco.

Como objetivos gerais, a pesquisa é a exploratória porque busca estabelecer o conhecimento dos riscos existentes e a aplicação prática para análise, avaliação e tratamento dos mesmos, sendo que a pesquisa exploratória busca estabelecer maior familiaridade com o problema, com vista a torná-lo mais explícito ou construir hipóteses [69].

A estratégia da pesquisa em questão é o "estudo de caso" partindo do pressuposto que será realizada o estudo de forma crítica, testando a hipótese da aplicação de instrumento computacional como forma de apoio a gestão de riscos na reengenharia de software.

De acordo com Engel apud Gil [71], um estudo de caso pode ser caracterizado como um estudo de uma entidade bem definida como um programa, uma instituição, um sistema educativo, uma pessoa, ou uma unidade social.

Visa conhecer em profundidade o como e o porquê de uma determinada situação que se supõe ser única em muitos aspectos, procurando descobrir o que há nela de mais essencial e característico [71].

De forma a melhorar o método de construção da pesquisa, o presente trabalho foi estruturado em IV etapas compreendidas na figura 19:

Figura 19 Etapas da Pesquisa. Fonte: Elaboração própria

A primeira etapa desse trabalho está dividida em duas partes onde busca estudar e selecionar métodos, ferramentas e técnicas que podem ser utilizadas para identificar riscos no processo de reengenharia de software e posteriormente estabelecer o contexto organizacional.

A segunda etapa desta pesquisa é voltada para pesquisa bibliográfica e documental para identificar métodos de identificação de riscos nesse processo, como também a realização do mapeamento do processo atual (AS-IS) de migração da tecnologia dos sistemas legados do produto CCO do projeto Renovação da CredTec através das técnicas e princípios do BPMN.

A terceira etapa dessa pesquisa procura identificar, através da revisão bibliográfica e documental quais são as ferramentas e técnicas utilizadas para análise e avaliação dos riscos no processo de reengenharia de software e de posse disso e em conjunto com as etapas anteriores, elaborar uma proposta de processo TO-BE com a mitigação dos riscos identificados no processo AS-IS.

A última etapa de todo esse trabalho foca na aplicação do conhecimento adquirido nas etapas anteriores para criação de proposta de ferramenta computacional que permita a gestão de riscos aplicada ao processo de reengenharia de software. Posteriormente, a aplicação prática do instrumento computacional.

Capítulo 4

Estabelecimento de contexto

Nesse capítulo será abordado o estabelecimento do contexto organizacional e estrutura do projeto de reengenharia de software ao qual essa pesquisa se aplica.

4.1 A organização

A empresa CredTec é considerada umas das maiores instituições financeiras privadas do Brasil, sendo a 5ª maior nesse ranking.

Possuindo uma média de 400 agências, 2.7 mil pontos de atendimento, mantém cerca de 4.6 milhões de clientes oferecendo os melhores produtos do mercado financeiro como: conta corrente, conta poupança, cheque, pagamentos do Instituto Nacional do Seguro Social (INSS), etc.

Para que a CredTec consiga manter-se em destaque no mercado, ela busca estar alinhada com os contextos que impactam diretamente em seu negócio, sejam eles externos ou internos.

4.1.1 Contexto

Para definição do contexto externo, são considerados atores de fora da instituição que possuem um impacto direto ou indireto em seu negócio, sendo eles: ambiente cultural, social, político, legal, regulatório, financeiro, tecnológico, econômico, natural e competitivo, seja internacional, nacional, regional ou local.

Para o ambiente cultural, social e econômico, a CredTec possui por prioridade trazer a transformação ao meio em que está instalado, devido ao fato de ser um sistema de cooperativos e não de fins lucrativos, ou seja, todo aquele que é associado a CredTec ganha participação em seu crescimento.

Para a CredTec, que é uma instituição financeira de cooperativismo sem fins lucrativos e que segue os normativos para pessoas jurídicas de atuação financeira:

- Documentação regulamentar do Bacen;

- Documentação regulamentação da Febraban;
- Documentação regulamentação da Presidência da República;
- Documentação regulamentada da Receita Federal;
- Documentação regulamentadora do Ministério do Planejamento de Orçamento, entre outros.

Para Governança, Risco e Conformidade, a CredTec recorre a uma série de controles externos como leis, normas, *Benchmarking*, podendo ser listados alguns abaixo:

- Lei geral de proteção aos dados pessoais – LGPD;
- Família ISO 9000 para gestão de qualidade;
- ISO 27001, ISO 27002 para segurança da Informação;
- Para desenvolvimento de Software: Rational Unified Process (RUP), Unified Modeling Language (UML), SCRUM para alguns de seus projetos;
- Information Technology Infrastructure Library (ITIL), Control Objectives for Information and Related Technologies (COBIT) e Project Management Body of Knowledge (PMBoK®).

São considerados também como contexto externo são fatores–chave e as tendências que tenham impacto sobre os objetivos da organização:

- Mercado financeiro em geral;
- Bolsa de Valores;
- Normas do Bacen;
- Normas da Febraban;
- Demais normas governamentais regulamentadoras para instituições financeiras.

São consideradas itens de impacto ao contexto externos as relações com partes interessadas externas e suas percepções e valores, como:

- Existem as parcerias com outras instituições financeiras e de crédito para administração de aquisição de outros produtos;
- Acessibilidade de seus produtos para todas as classes sociais;
- Ganhos financeiros para todos os seus clientes que irão possuir participação dos lucros e retorno das taxas investidas pelo cliente;
- Proporcionar financiamentos, entre outros produtos com a menor taxa possível. Para o contexto interno. a CredTec busca atingir seus objetivos que impactam diretamente seus ativos são pautados pela estrutura organizacional,

cultura interna, políticas, diretrizes, normas, pessoas, processos e suas tecnologias.

Para o contexto interno de governança, estrutura organizacional, funções e responsabilidades, são estabelecidos os seguintes princípios:

- Regulamentar a prática de relacionamento ético e íntegro entre os acionistas/associados, os órgãos de administração e fiscal, bem como a diretoria e a auditoria;
- Agir de forma a satisfazer as diferentes necessidades de informações, por meio da adoção de gestão transparente e íntegra;
- Propiciar tratamento justo e igualitário;
- Estimular a disseminação dos princípios éticos e os compromissos de condutas constantes deste código;
- Promover negociações honestas e justas, sem auferir vantagens indevidas por meio de manipulação, uso de informação privilegiada e outros artifícios dessa natureza;
- Manter canal de recepção, encaminhamento e processamento de opiniões, sugestões, reclamações, críticas e denúncias sobre os mais variados assuntos, inclusive transgressões éticas.

Para apoio aos sistemas de informação, fluxos de informação e processos de tomada de decisão (tanto formais como informais), são adotadas medidas como:

Organizar e padronizar procedimentos relacionados às atividades desenvolvidas pelas agências da CredTec, instituindo padrões normativos, tecnológicos, operacionais, econômicos, financeiros, jurídicos, contábeis, de modo a prevenir e a combater riscos operacionais e sistêmicos;

- Supervisionar as atividades desenvolvidas em entidades que compõem a CredTec, por meio de auditorias direta e indireta, incluindo, nessa supervisão, avaliações econômico-financeiras, contábeis e de controles internos;
- Organizar, manter e coordenar fóruns de discussão de temas de interesse da CredTec, por meio de comitês consultivos ou grupos de trabalhos;
- Desenvolver soluções tecnológicas e prestar serviços de processamento eletrônico, transmissão e arquivamento de dados, de operações financeiras e de controles operacionais das entidades da CredTec, por meio de sistema tecnológico;

- Prestar serviço de monitoramento do Sistema de Controle Interno para as entidades que compõem a CredTec, formulando e implantando instrumentos de regulação de aplicação sistêmica e processos padronizados;
- Estabelecer diretrizes de atuação sistêmica com vistas à observância dos princípios da eficiência, da economicidade, da utilidade e dos demais princípios cooperativistas.

Além dos fatores culturais, outros itens que influenciam diretamente no contexto interno da CredTec são suas normas, diretrizes e modelos adotados por ela conforme exemplos:

- Política de Segurança da Informação da CredTec;
- Código de Ética da CredTec;
- Regimento Interno e Regulatório;
- Política Institucional de Auditoria Interna;
- Política Institucional de Gerenciamento de Risco de Mercado;
- Política Institucional de Governança Corporativa;
- Política Institucional de Risco de Crédito;
- Política Institucional de Risco Operacional;
- Metodologia de Desenvolvimento de Software;
- Manuais de Procedimentos Internos.

E for fim, a CredTec busca estabelecer parcerias com outras empresas prestadoras de serviços de natureza tecnológica, com a finalidade de prestar serviços de tecnologia, como por exemplo: disponibilização de software para pagamento de benefícios da previdência social ou convênios como água, luz, telefone etc., de forma a estender suas relações contratuais.

Após definição do contexto, de forma a apoiar o processo de gestão de riscos, foi realizada uma modelagem do processo atual de reengenharia de software, explicado no próximo tópico.

4.2 Modelagem do processo AS-IS

Foram selecionados dois projetos para realização da modelagem do processo atual de reengenharia de software da empresa CredTec, sendo eles: o projeto de conta corrente e de benefícios da previdência social.

Cada um dos projetos, tanto conta corrente e benefícios da previdência social, possuem uma série de softwares que diariamente necessitam de manutenções, melhorias, e correções de incidentes em ambientes de produção, ocasionando uma necessidade de reengenharia deles.

Para seleção desses dois projetos foi realizada uma análise in loco que buscou estabelecer um grau de criticidade dos softwares migrados como um todo, que posteriormente serviu de apoio aos gestores técnicos na tomada de decisão.

Com análise in loco, foi possível criar uma matriz que considera os aplicativos mais críticos, suas funções em média de transações realizados por meio deles, conforme o quadro 4:

Quadro 4 Média de transações por aplicativos a ser migrado. Fonte: Elaboração Própria.

Aplicativos	Funções	Média de Transações
Abertura de Conta: Canais de retaguarda, *internet bank*, mobile *banking* e parceiros.	Porta de entrada e captação de clientes para o banco; Abertura de conta salário; Abertura de conta corrente depósito à vista; Abertura de conta INSS.	4.247 aberturas de novas contas.
Cheques	Solicitação e depósito de cheque; Compensação de cheques; Cadastro de restrições e etc.	30.795 transações envolvendo cheques.
Movimentação Financeira	Transações financeiras de crédito; Transações financeiras de débito.	187.568 transações financeiras de crédito e débito em contas.
Benefícios da previdência social	Pagamento de benefícios do INSS.	560.000 benefícios processados

Essa análise considerou a avaliação de dados em ambiente de produção por média de transações diárias no período de abril de 2019 até novembro de 2020 de aplicativos migrados.

Com o resultado da análise in loco e geração do quadro de aplicativos migrados mais críticos, os gestores técnicos de ambos os projetos puderam realizar a tomada de decisão sobre quais softwares seriam utilizados para a gestão de riscos proativa no processo de reengenharia de software.

Com o auxílio dos gestores técnicos desses projetos de reengenharia de software e o especialista em gestão de riscos, foi realizada uma modelagem do cenário atual com utilização do BPMN.

Como método para identificação do processo atual (AS-IS), foi utilizada a técnica *Brainstorming*, onde os dois gestores técnicos tiveram a oportunidade de explanar sobre o processo atual de reengenharia de software dos seus projetos.

Os dois gestores técnicos mostraram de forma simples e direta o funcionamento do processo atual de reengenharia de software do projeto conta corrente e benefícios da previdência social.

Com isso foi identificado que cada um desses projetos possuem um núcleo de profissionais dedicados composto em média por cinco analistas de sistemas, cinco desenvolvedores e um líder de projetos.

Além da equipe dedicada, os projetos contam com o apoio de um núcleo de teste e homologação que atende demandas de qualidade dos softwares, uma equipe de arquitetura da informação para validação dos modelos de dados Modelo de Entidade e Relacionamento (MER) gerados e *Data Definition Language* (DDL) e uma área negocial gestora que possui o entendimento do negócio inerente à tecnologia que precisa ser renovada ou construída. Todas as equipes trabalham de forma separada, sendo essas demandadas periodicamente, conforme necessidade de cada projeto.

Após implantação da demanda em ambiente produtivo, o projeto conta com uma equipe externa responsável pelo atendimento às solicitações em primeiro nível, onde os clientes da CredTec e usuários dos sistemas registram suas reclamações de eventuais problemas da funcionalidade.

Caso esse suporte de primeiro nível não seja satisfatório ou não consiga resolver o problema apresentado, a solicitação é repassada para a equipe técnica de suporte em segundo nível que é integrada ao projeto.

Quando a equipe de segundo nível não consegue resolver a solicitação proposta, a demanda é repassada para o analista de sistemas que originalmente migrou o software que deve interromper todas as suas atividades atuais para correção do incidente apresentado. Tal incidente, terá a solução desenvolvida e passará por todo o processo de especificação, teste, homologação, validação e disponibilização em produção.

Um ponto crítico do projeto é que muitas demandas envolvem outras equipes internas e até mesmo parceiros externos à CredTec, fazendo com que as solicitações tenham demora em seu desenvolvimento, incompatibilidade, redundâncias, erros em ambiente produtivo e até mesmo cancelamento do projeto.

Outro ponto crítico é que o cliente negocial, responsável pela validação do produto de software migrado, não é envolvido em todas as etapas do projeto, correndo o risco, muitas

das vezes, do projeto sofrer alterações no momento de sua homologação, aumentando o tempo, custo e risco.

Em posse dessas informações e utilizando técnica de modelagem de processos BPMN

- *Business Process Model and Notation* e o software Bizagi, foi realizado o desenho do processo AS-IS de reengenharia de software dos projetos de conta corrente e benefícios da previdência social, onde ele foi dividido em 6 subprocessos sendo: pré-projeto, concepção, requisitos, desenvolvimento, teste integrado e homologação conforme figura 20:

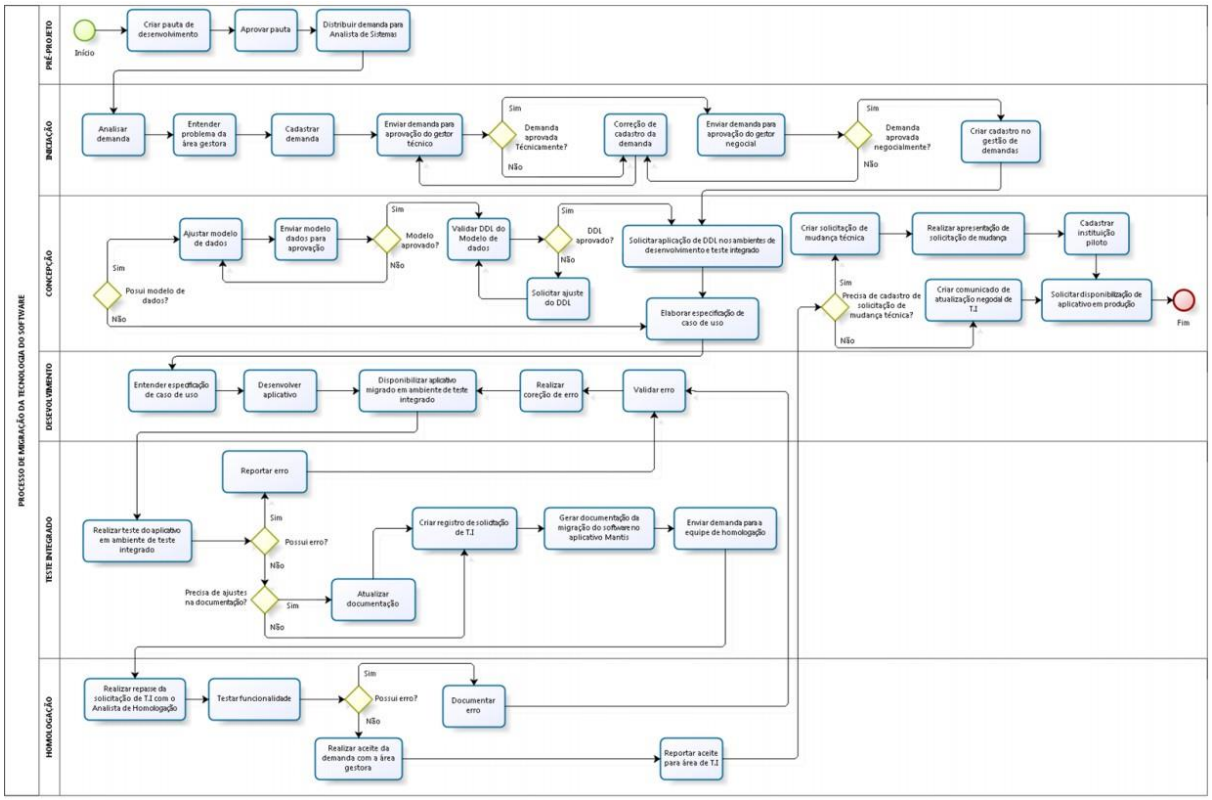


Figura 20 Processo AS-IS de Reengenharia de Software. Fonte: Elaboração própria

Para melhor visualização, o arquivo foi disponibilizado no ***Link***.

A modelagem desse processo não contempla o detalhamento dos processos específicos do Pré-Projeto e Homologação tendo em vista que são atividades executadas externamente aos projetos de conta corrente benefícios da previdência social, detalhadas a seguir:

- Pré-Projeto: Essa fase diz respeito ao surgimento de cada projeto particularmente. É criada uma pauta de desenvolvimento onde estão contidos os projetos de migração de software, novas funcionalidades e melhoria dos aplicativos em ambiente produtivo.

- Concepção: Essa fase é uma das mais dispendiosas no processo pois a mesma diz respeito a fazer todo o entendimento do negócio, analisar se existe impacto nos sistemas legados, mapear quais são as funcionalidades que estão ligadas àquilo que será migrado, se haverá transformação de dados, verificar quais são as tecnologias necessárias como rotinas Beth, filas, api's, migração de dados, etc.
- Requisitos: Essa fase tem por objetivo a documentação de tudo aquilo que foi identificado e é necessário para o processo de migração do software.
- Desenvolvimento: Essa é a fase de implementação de toda a solução elicitada. O desenvolvimento é executado pelo Analista de Sistemas e Desenvolvedor de Software e pode estar nos seguintes âmbitos:

– Banco de Dados: migração de dados e transformação de dados, cargas de dados, criação de objetos de banco como *views*, *trigger*'s, *functions*, tabelas locais e tabelas replicadas;

– Implementação de código: serviços externos e internos *Application Programming Interface* (API), rotinas batch, construção de relatórios, implementação de telas e filas (escutas);

– Migração de dados: Quando o banco de dados pertence a outras equipes e é necessário fazer alguma carga de um banco de dados de outro projeto para o banco de dados do projeto conta corrente, é possível solicitar uma carga de dados.

- Teste Integrado: Essa fase ocorre quando a equipe de desenvolvimento disponibiliza todos softwares gerados em ambiente de teste integrado para avaliação dos analistas de sistemas.
- Homologação: A homologação é executada por um núcleo de trabalho externo à equipe de CCO e BPS.

Após a modelagem do processo atual de reengenharia de software dos projetos CCO/BPS, foi possível identificar e criar uma base de conhecimento com os possíveis controles aplicados ao processo AS-IS que serão apresentados no próximo capítulo e dar início a proposta de gestão de riscos, que serão apresentados no próximo capítulo.

Capítulo 5

Modelo de gestão de riscos

Nesse capítulo será apresentada a proposta de gestão de riscos aplicada ao processo de reengenharia de software.

Em alinhamento com um especialista em gestão de riscos da CredTec, foi selecionado o modelo de gestão de riscos da ISO 31.000 em conjunto com as ferramentas e técnicas propostas pela ISO 31.010 e outras ferramentas associadas para realização da gestão de riscos no processo de reengenharia de software.

Tal estrutura foi selecionada, tendo em vista que já é utilizada em outras áreas da instituição, não sendo aplicada até em então ao processo de reengenharia de software.

A gestão de risco proposta foi dividida em 4 etapas principais, sendo elas: Identificação, Análise, Avaliação e Tratamento dos riscos.

- Identificação de Riscos: O intuito dessa etapa foi identificar os possíveis riscos, ameaças e controles associados ao processo atual AS-IS de reengenharia de software. Ela distribuída em mais três sub etapas:
  - Identificação dos objetivos organizacionais;
  - Seleção das atividades de processo a serem analisadas;
  - Definição de pessoas e perfis para participar do processo.
- Análise de Riscos: Nessa etapa buscou-se estabelecer a probabilidade e impacto de cada risco identificado. Ela foi dívida em duas sub etapas:
  - Análise de Probabilidade;
  - Análise de Impacto.
- Avaliação dos riscos: Para essa etapa, foram definidos os níveis de cada risco, como também a priorização deles.
- Tratamento de Riscos: No tratamento dos riscos, foi realizada uma modelagem de processo TO-BE com a mitigação dos riscos identificados no processo AS-IS.

Nas próximas seções será apresentado o detalhamento de cada uma das etapas citadas.

5.1 Identificação dos riscos

Essa sessão descreve as ações realizadas para identificação dos riscos associados ao processo AS-IS de reengenharia de software do projeto CCO e BPS.

Como premissa para esse processo, foi indicado pelas áreas participantes que seria realizada a identificação apenas dos possíveis riscos que podem trazer um impacto negativo para a organização.

Essa etapa foi dívida em mais três sub etapas descritas a seguir.

5.1.1 Seleção de atividades de processo

Foi realizada uma reunião com o gestor técnico do produto de benefícios da previdência social para definição dos objetivos da pesquisa em questão aplicada a reengenharia de software desse produto.

Para a ocasião, a proposta da gestão de riscos aplicada ao processo AS-IS de reengenharia de software foi revisada e buscou-se estabelecer uma aplicação para o processo juntamente com o requisito de se desenvolver uma ferramenta que possibilitasse tal ação.

Utilizando a técnica de *Brainstorming*, permitiu-se que o gestor técnico pudesse expor a maiores riscos, na sua visão relacionados à reengenharia de software.

Identificação dos objetivos organizacionais

Foi ressaltado para o gestor técnico que seria de suma importância a definição inicialmente dos objetivos organizacionais associados à reengenharia de software, ou seja, o alvo que a organização pretende atingir com o produto benefícios da previdência social.

Dessa forma, foi definido o objetivo organizacional: "Migração do produto BPS da plataforma legada para novas tecnologias de mercados" para "Expansão da carteira de clientes e adequação a auditorias internas".

Com o apoio de um especialista em gestão de riscos na instituição, associados a esse objetivo, identificou-se os riscos, fatores de riscos, impacto, tratamento e possíveis controles associados a ele nos projetos CCO e BPS. No quadro 5 a seguir estão descritos alguns deles:

Quadro 5 Objetivos Organizacionais e Riscos Associados. Fonte: Elaboração própria.

Risco	**Afastamento do Analista Responsá-vel**
Fator de Risco	Baixa remunaração, insatisfação, acidente etc.
Impacto	Impossibilidade de expansão da carteira de benefícios
Tratamento	Redesenho do Processo e Reengenharia do Software.
Controle	Repasse de conhecimento; Manutenção de um bom clima organizacional; Medidas de segurança na empresa, vacinação e campanhas de saúde. Constantes treinamentos sobre a ferramenta
Risco	**Perda do conhecimento, inteligência do negócio**
Fator de Risco	Mudança de emprego, acidente, morte etc.
Impacto	Impossibilidade de expansão da carteira de benefícios
Tratamento	Redesenho do Processo e Reengenharia do Software.
Controle	Repasse de conhecimento; Manutenção de um bom clima organizacional; Medidas de segurança na empresa, vacinação e campanhas de saúde. Constantes treinamentos sobre a ferramenta

Foram selecionados apenas dois riscos associados ao objetivo organizacional em relação ao processo de reengenharia de software para apresentação.

É possível observar que ambos os riscos possuem um tratamento específico que na visão do gestor seria a remodelagem do processo de reengenharia de software.

A identificação dos objetivos organizacionais serviu de direcionamento e entrada para as demais atividades.

Seleção de atividades de reengenharia de software

Após a identificação dos objetivos organizacionais, foi solicitado ao gestor técnico que pudesse elencar quais atividades do processo de reengenharia de software, que em sua visão e experiência, seriam a mais críticas e importantes.

Essa seleção foi necessária para definir quais seriam as atividades do processo de reengenharia de software utilizadas para a aplicação da gestão de riscos.

O gestor técnico informou que todo o processo é importante para a gestão de riscos. Todavia, priorizou as atividades relacionadas ao pré-projeto, iniciação e concepção, tendo em vista que essas são mais críticas e impactavam diretamente a conclusão do projeto.

Exemplo, uma solução de migração que depende do envolvimento de outras equipes tem um alto risco de não ser concluída caso as partes interessadas não sejam identificadas e envolvidas no início da reengenharia.

Outra definição seria que algumas vezes os gestores negociais não tinha com clareza a necessidade de solução tecnológica bem definida, o que demandava a criação de solução de T.I, porém, sem um desenho da proposta.

Definição de pessoas e perfis

Outro ponto traçado nessa reunião para início da gestão de riscos aplicada ao processo de reengenharia de software, foi a definição das pessoas e seus perfis para apoio na identificação, análise e avaliação de riscos.

A definição ficou como: um gestor técnico, um gestor negocial, um desenvolvedor de software, um analista de sistemas e um especialista em riscos.

Cada um dos perfis poderia auxiliar no processo de identificação de riscos conforme atividades associadas em seus perfis, sendo que análise e avaliação de riscos estaria restrita aos cargos de gestão.

Exemplo: O analista pode responder perguntas sobre a fase de iniciação, concepção, desenho e teste de software. Já o desenvolvedor, desenvolve e testa a aplicação.

Após o encerramento da reunião e documentação dos requisitos iniciais para a gestão de riscos aplicada ao processo de reengenharia de software, foi realizada a elaboração de questionário para identificação dos riscos nas atividades de processo.

5.1.2 Elaboração de questionário

Baseando-se no perfil organizacional, disponibilidade da equipe definida no tópico 5.1.1 e com o auxílio do especialista em gestão de riscos, foi definida a técnica de questionário para identificação dos possíveis riscos associados ao processo de reengenharia de software. Para o questionário buscou-se permitir que o entrevistado pudesse informar de maneira clara possíveis riscos, ações preventivas e contingenciais relacionadas a cada atividade elencada do processo de reengenharia de software.

Na elaboração do questionário foi utilizada a ferramenta "Google Forms"e criado um questionário para cada perfil.

Foram elaborados três questionários, sendo um para os perfis de gestor técnico e negocial, um para analista de sistemas e outro para o perfil de desenvolvedor.

No questionário, foi indicada uma orientação para o seu preenchimento, conforme figura 21:

Figura 21 Cabeçalho do questionário. Fonte: Elaboração própria

Além do cabeçalho com as orientações sobre o preenchimento da pesquisa, foram definidas as perguntas, conforme próximo tópico.

Definição de perguntas

Utilizando a opinião de especialista, as perguntas foram definidas cada atividade do processo relacionada aos perfis elencados, onde o entrevistado pudesse descrever e enumerar de forma sequencial os riscos relacionados a atividade em questão, com suas respectivas ações preventivas e contingenciais.

A ação preventiva é o controle associado ao risco, ou seja, aquilo que pode ser feito para modificar o risco.

Já ação contingencial é um possível tratamento ao incidente ocorrido com a materialização do risco.

Para os perfis de "Analista de Sistemas" e "Desenvolvedor", o gestor técnico elencou três atividades mais críticas.

Não houve questionário para o especialista em gestão de riscos, tendo em vista que esse colaborador fez parte da validação da pesquisa.

Validação de Perguntas

De maneira a validar as perguntas, foi realizada a submissão de link da proposta de questionário para cada um dos entrevistados de forma individual via aplicativo de mensagem.

Para os perfis mais técnicos, houve dúvida na separação dos riscos por tópico, as ações preventivas e contingências relacionadas a eles.

Dessa forma, cada caso e questionamento foi tratado de maneira particular com a explicação sobre o procedimento para resposta ao questionário.

Disponibilização de questionário

Para disponibilização do questionário, foi gerado um link através do Google Forms para cada um dos três questionários e enviado de forma individual para os colaboradores selecionados.

Para apresentação nesse trabalho, foram selecionadas duas perguntas respondidas por um colaborador com perfil de gestor técnico:

- Riscos na aprovação da pauta

– Risco: Grande quantidade de demandas que não agregam valor aprovadas.

∗ Ação preventiva: Classificar em ordem de agregação de valor todas as demandas.

∗ Ação preventiva: Capacitar o Gestor para que conheça a solução e possa decidir favoravelmente pelas demandas que agregam mais valor.

∗ Ação contingencial: Não avançar com o desenvolvimento da demanda e comunicar ao Gestor esta decisão

∗ Ação contingencial: Selecionar demanda que agrega mais valor.

– Risco: Número excessivo de demandas selecionadas para desenvolvimento

∗ Ação preventiva: Deixar o Gestor a par da capacidade de desenvolvimento da equipe.

∗ Ação preventiva: Tornar o Gestor componente ativo no processo de reengenharia.

∗ Ação contingencial: O gestor deverá priorizar as demandas que deverão ser desenvolvidas de acordo com a ordem de agregação de valor.

• Riscos na distribuição de demandas

– Risco: Ausência de Capacitação da equipe da *Squad* para desenvolvimento da demanda.

∗ Ação preventiva: Capacitar a equipe para desenvolvimento de qualquer demanda.

∗ Ação preventiva: Envolver na *Squad* como membro permanente ou temporário os maiores conhecedores dos assuntos que envolvem a funcionalidade a ser desenvolvida (negocial/técnico).

– Risco: Seleção de número excessivos para a *squad* com baixo número de componentes

∗ Ação preventiva: Sempre avaliar a capacidade da *squad* para o desenvolvimento das demandas.

∗ Ação contingencial: O Gestor deverá selecionar as demandas que farão parte do ciclo de desenvolvimento de acordo com seu grau de agregação de valor.

Os três questionários foram respondidos e posteriormente cada uma das respostas foram tratadas e documentadas.

5.1.3 Tratamento e documentação dos dados

Foi realizada uma análise em cada um dos questionários respondidos de forma a consolidar e documentar esses dados para validação posterior.

Os dados foram consolidados e documentados, conforme exemplo do quadro 6:

Quadro 6 Documentação dos Riscos. Fonte: Elaboração própria.

Atividade de Processo	Riscos	Tratamento	Controles
Riscos na criação da pauta	Tentativa de atender o querer do Gestor Negocial e não o que é necessário para a solução.	Identificar demandas que agregam valor para o cliente e para a sociedade. - Identificar demandas mandatórias de órgãos regulamentadores e normatizadores. - Envolvimento das partes interessadas na definição das demandas que serão incluidas na Pauta para tomarem conhecimento do contexto atual no momento da montagem da pauta e entenderem as dores das outras partes interessadas.	Criar um radiador de comunicação e maior proximidade com o Gestor e comunicar que a demanda que é um "querer" e não uma necessidade deverá perder o nível de prioridade e ser escalada para um próximo ciclo de desenvolvimento.

Riscos na criação da pauta	Gestor Negocial não seleciona adequadamente as demandas que agregam mais valor para o Negócio.	Capacitar o Gestor para executar com louvor a seleção das demandas que serão componentes da Pauta; - Informar ao Gestor as dificuldades do desenvolvimento em cada demanda e classificar a demanda quanto ao valor para o cliente. - Trazer o Gestor ou algum membro de sua equipe para compor a Equipe da *Squad* para que todas as informações sejam conhecidas pelos Gestores.	Não avançar com o desenvolvimento da demanda e comunicar ao Gestor esta decisão.

Análise e validação das respostas

Após a documentação das respostas dos questionários, as mesmas foram analisadas e consolidadas para apresentação ao time.

Para validação das respostas foi utilizada a técnica de *Brainstorming* onde cada um dos riscos e suas respectivas ações preventivas e contingenciais foram avaliadas pela equipe como um todo.

Além da validação dos riscos, foi colocado em pauta também a associação desses riscos com fatores de riscos ou possíveis vulnerabilidades no processo, como o exemplo:

- Documentação inexistente do sistema legado;
- Falta de repasse negocial e técnica para a equipe de desenvolvimento;
- Falta de envolvimento com outras envolvidas impactadas pelo software migrado;
- Inexistência de plano de comunicação;
- Inexistência de participação da área gestora;
- Falta de planejamento;
- Mudança constante de gestor negocial;
- Constante troca de tecnologia;
- Sobreposição de demandas;
- Poucos recursos humanos.

Conforme os participantes pontuaram, essas vulnerabilidades foram identificas a partir do *Benchmarking* de projetos anteriores de reengenharia de software, em experiência vivenciada por eles.

Esse levantamento serviu para complementar a documentação dos riscos.

Documentação dos riscos

Com os fatores de riscos identificados, foi realizada a associação desses fatores, com os controles anteriormente levantados em conjunto com a equipe, gerando por sua vez o quadro a seguir com a indicação da associação das vulnerabilidades, riscos e controles aplicáveis conforme quadro 7:

Quadro 7 Associação das ameaças com os riscos. Fonte: Elaboração própria.

Etapa do Processo	Fator de Risco (Ameaça)	Riscos
Pré-Projeto	Inexistência de planejamento e falta de participação da área gestora.	Mudanças não mapeadas; - Falta de compatibilidade com outras funcionalidades; - Erros desconhecidos em ambiente produtivo; - Descumprimento de requisitos negociais; Aumento do custo do projeto; Atraso no projeto.
Concepção	Sobreposição de demandas e falta de envolvimento de outras áreas afetadas pelo software migrado.	Erros desconhecidos em ambiente produtivo; - Desenvolvimento desnecessário de atualização de software ou nova funcionalidade; Insatisfação do usuário final; Aumento do custo do projeto; Aumento do prazo do projeto.

Essa etapa da documentação dos riscos gerou uma entrada para a ação de identificação e documentação de uma base de conhecimento com os controles internos e externos.

Documentação de base de conhecimento

Após a documentação inicial dos riscos, foi realizado o levantamento e documentação de uma base de conhecimento com os possíveis controles aplicados às atividades do processo de reengenharia de software.

Para essa base, foram considerados os controles internos, apontados pela equipe que respondeu o questionário, como também os controles externos a partir de um levantamento documental.

Utilizando a técnica de revisão bibliográfica e com o apoio do especialista em gestão de riscos, foram analisadas leis, normas, técnicas e boas práticas de mercado associadas ao processo de reengenharia de software, como por exemplo: ISO 25010 sobre qualidade de software ou PMBoK® sobre gestão de riscos e gerenciamento de projetos.

Foram considerados também, os riscos identificados no mapeamento do processo atual AS-IS de reengenharia de software e dessa forma, gerando uma base de conhecimento conforme exemplo no quadro 8:

Quadro 8 Exemplo de controles na base de conhecimento. Fonte: Elaboração Própria.

Cód.	Controles	Tipo	Fonte
C01	Capacitar o Gestor para melhor seleção de itens demandas da pauta	Controle Interno	Análise técnica.
C02	Comunicar ao Gestor as dificuldades do desen- volvimento em cada demanda	Controle Interno	Análise técnica.
C03	Classificar a demanda quanto ao valor para o cliente	Controle Interno	Análise técnica.
C04	O Gestor ou algum membro de sua equipe deve compor a Equipe a equipe de desenvolvimento	Controle Interno	Análise técnica.
C05	Manter o Gestor como componente da equipe da equipe	Boas práticas	SCRUM Fra- mework
C06	Deve-se realizar a proteção de erros de usuários	Norma	ISO/IEC 25010:2011 Sys- tems and software engineering
C07	Desenvolver requisitos não funcionais de auten- ticação na aplicação para garantir que nenhumuso idenvido seja realizado	Lei	Lei 13.709 (2018) - LGPD

C08	Identificar todos os sitemas que devem integrar com o software migrado	Norma	ISO/IEC 25010:2011 Systems and software engineering
C09	Aumentar a capacidade do sistema de integrar com outros softwares (Interoperabilidade)	Norma	ISO/IEC 25010:2011 Systems and software engineering
C10	Identificar todas as necessidades, premissas, restrições e requisitos iniciais do projeto	Boas práticas	PMBoK® (Project Management Body Of Knowledge)
C11	Identificar os responsáveis pelo projeto	Boas práticas	PMBoK® (Project Management Body Of Knowledge)
C12	Identificar todas as partes interessadas	Boas práticas	PMBoK® (Project Management Body Of Knowledge)

A elaboração dessa base de conhecimento levou em consideração projetos anteriores e o modelo de *Benchmarking*, finalizando assim a primeira etapa do processo de gestão de riscos que é a identificação dos riscos.

Na próxima seção será apresentada a análise dos riscos.

5.1.4 Análise de riscos

A etapa de análise dos riscos consistiu em identificar o grau de probabilidade de um risco se concretizar e o grau do impacto que o incidente poderia ocasionar.

Como entrada nesse processo foi utilizado o objetivo organizacional e os riscos identificados.

Utilizando o método de apoio de especialista e análise de *Benchmarking* de outros projetos internos na instituição, foi elencada a ferramenta de matriz e probabilidade para documentar a análise dos riscos.

Nessa etapa buscou-se também identificar e documentar os possíveis impactos à organização relacionados a cada risco identificado.

Identificação do impacto

Foi realizada uma reunião com o especialista em gestão de riscos e o gestor técnico para definição do grau de probabilidade e impacto de cada riscos.

Tendo como base a vivência em projetos anteriores de reengenharia de software, o gestor técnico e o especialista em gestão de riscos, conseguiram avaliar a probabilidade e impacto de cada risco identificado.

Antes de realizar a indicação do grau de probabilidade e impacto, foi realizada a identificação e registro dos impactos relacionados a cada risco, utilizando os métodos de análise histórica e opinião do Especialista, conforme ISO 31010.

Os possíveis impactos foram documentados conforme o quadro 9:

Quadro 9 Associação dos fatores de riscos aos possíveis impactos. Fonte: Elaboração Própria.

Cód.	Fatores	Possíveis Impactos
F01	Não atendimento das necessidades para a solução	R01: Descontinuidade inexperada do projeto; R02: Erros em ambiente produtivo; R03: Atraso na conclusão do projeto.
F02	Falta de seleção das demandas que agregam mais valor para o negócio	R04: Criação de software que será descontinuado; R05: Prejuízo financeiro; R01: Descontinuidade inexperada do projeto.
F03	Gestor Negocial influenciado negativamente pelas partes interessadas	R01: Descontinuidade inexperada do projeto; R03: Atraso na conclusão do projeto; R06: Desmotivação dos membros da equipe;R05: Prejuízo financeiro.
F04	Baixo envolvimento do gestor com o desenvolvimento da solução	R02: Erros em ambiente produtivo; R07: Retrabalho para corrigir incidentes.
F05	Desenvolvimento de grande quantidade de demandas que não agregam valor aprovadas	R01: Descontinuidade inexperada do projeto; R05: Prejuízo financeiro.
F06	Inclusão de número excessivo de demandas	R03: Atraso na conclusão do projeto; R06: Desmotivação dos membros da equipe;R07: Perda de recursos humanos.

F07	Ausência de Capacitação da equipe da demanda	R02: Erros em ambiente produtivo; R03: Atraso na conclusão do projeto;R05: Prejuízo financeiro; R07: Retrabalho para corrigir incidentes;
F08	Número excessivos de demandas em relação a baixa quantidade derecursos humanos	R03: Atraso na conclusão do projeto; R06: Desmotivação dos membros da equipe;R07: Perda de recursos humanos.
F09	Despadronização de tecnologias desenvolvidas	R02: Erros em ambiente produtivo; R03: Atraso na conclusão do projeto;R05: Prejuízo financeiro; R07: Retrabalho para corrigir incidentes.
F10	Erros em ambiente produtivo	R03: Atraso na conclusão do projeto; R05: Prejuízo financeiro; R07: Retrabalho para corrigir incidentes;

Foi possível observar que alguns impactos se tornaram possíveis riscos na visão dos gestores técnicos e especialista em gestão de riscos, como o caso do I02: Erros em ambiente produtivo.

Após os registros dos impactos, foi realizada a análise para registros dos graus de probabilidade de impacto relacionados a cada risco identificado.

Registro do grau de probabilidade de impacto

Como citado no tópico anterior, através da opinião do especialista, foi elencada a ferramenta "Matriz de Probabilidade e Impacto" indicada por boas referências como PMBoK® e a própria ISO 31010 como um método eficaz para registro do grau dos riscos.

Tal tarefa buscou relacionar o grau de riscos de cada vulnerabilidade, pontuando a probabilidade de determinado risco se concretizar como também o impacto que o mesmo causaria ao processo de mudança de tecnologia do software.

Essas informações foram documentadas em uma matriz de probabilidade e impacto (ISO 31010, 2011) na qual se utiliza uma pontuação de probabilidade crescendo de 0,1 a 0,9, sendo 0,1 para pouco provável e 0,9 para muito provável que determinado risco se concretize em um incidente, e no caso do impacto de 1 a 5, sendo 1 para impacto muito baixo e 5 para impacto muito alto.

Para fins de cálculos, os valores foram multiplicados entre si gerando uma classificação de escore com o grau de cada risco, conforme o quadro 10:

Quadro 10 Matriz de Probabilidade e Impacto. Fonte: Próprio Autor, Adaptado de ISO 31010 [12].

Cód.	Riscos	Prob.	Impact.	*Score*
R01	Descontinuidade inexperada do projeto.	0,7	5	*3,5*
R02	Erros em ambiente produtivo.	0,7	5	*3,5*
R03	Atraso na conclusão do projeto.	0,3	4	*1,2*
R04	Migração de software que será descontinu-ado.	0,1	2	*0,2*
R05	Prejuízo financeiro.	0,9	2	*1,8*
R06	Desmotivação dos membros da equipe.	0,9	3	*2,7*
R07	Retrabalho para corrigir incidentes.	0,3	5	*1,5*
R08	Mudanças não mapeadas.	0,2	1	*0,2*
R09	Despadronização de tecnologias desenvolvi-das.	0,4	5	*2*
R10	Erros em ambiente produtivo.	0,9	2	*1,8*

Para indicar o grau de probabilidade e impacto, os participantes da reunião realizaram a análise dos registros dos impactos, como também uma avaliação histórica de outros projetos de reengenharia de software.

A indicação da probabilidade e impactou serviu como entrada para o processo de avaliação dos riscos que será abordado no próximo tópico.

5.1.5 Avaliação dos riscos

O objetivo dessa etapa foi definir o nível dos riscos identificados, como também priorizá-los para tratamento.

Como entrada nesse processo, foram utilizados os registros dos riscos e a matriz de probabilidade e impacto que contém o cálculo do score de cada risco.

Para definição do grau do risco, foi utilizada três níveis de avaliação a saber:

- Faixa 01 (cor verde): representa os riscos com menor nível de relevância em relação a análise realizada. Para esse nível o risco deve possuir um score entre 0,1 e 0,9.
- Faixa 02 (cor amarela): representa os riscos com nível intermediário de relevância em relação a análise realizada, mas que possuem um grau de atenção para avaliação. Para esse nível o risco deve possuir um score entre 1 e 2,1.
- Faixa 03 (cor vermelha): representa os riscos com alto nível de relevância em relação a análise realizada e indica um ponto crítico que deve ser considerado na avaliação dos riscos. Para esse nível o risco deve possuir um score entre 2,5 e 4,5.

O melhor entendimento do método está apresentando na figura 22:

Matriz de Probabilidade X Impacto					
Probabilidade	**Ameaças**				
0,9	0,9	1,8	2,7	3,6	4,5
0,7	0,7	1,4	2,1	2,8	3,5
0,5	0,5	1	1,5	2	2,5
0,3	0,3	0,6	0,9	1,2	1,5
0,1	0,1	0,2	0,3	0,4	0,5
Gravidade	1	2	3	4	5

Figura 22 Níveis de Riscos. Fonte: Elaboração própria adaptada de ISO 31010 [12]

Com essa avaliação foi possível auxiliar a área gestora sinalizando quais os riscos mais altos e que devem ter uma tomada de decisão para resposta aos mesmos.

Todavia, mesmo com a visão do nível de cada riscos, o gestor realizou a seleção daqueles que representam maior significância para o negócio, exemplo:

- R01 Descontinuidade inesperada do projeto;
- R02 Erros em ambiente produtivo;
- R03 Atraso na conclusão do projeto;
- R04 Migração de software que será descontinuado.

Conforme identificação dos riscos nos objetivos organizacionais e uma análise e avaliação dos riscos no processo de reengenharia de software, foi proposto para a área gestora uma modelagem de processo TO-BE com a mitigação dos riscos identificados no processo AS-IS.

Na próxima sessão será apresentado o tratamento dos riscos através da modelagem do processo TO-BE.

5.1.6 Modelagem do processo TO-BE

Como forma de diminuir os riscos identificados no processo AS-IS de reengenharia de software, foi proposta a modelagem do processo TO-BE de maneira a melhorar o processo anterior e mitigar os riscos selecionados.

A melhoria do processo atual reengenharia de software do produto CCO/BPS da Instituição Financeira CredTec, está pautada numa boa gestão dos riscos e no processo de qualidade e melhoria contínua, minimizando os fatores de riscos como:

- F01 Não atendimento das necessidades para a solução;
- F05 Desenvolvimento de grande quantidade de demandas que não agregam valor aprovadas;
- F07 Ausência de Capacitação da equipe para desenvolvimento da demanda;
- F10 Possíveis erros em ambiente produtivo.

Ameaças essas que podem causar impactos negativos nos objetivos organizacionais, por exemplo:

- R01: Descontinuidade inesperada do projeto;
- R03: Atraso na conclusão do projeto.
- R04: Migração de software que será descontinuado;
- R05: Prejuízo financeiro.

Baseando-se nos riscos, seus fatores e impacto, focando na qualidade da reengenharia de software, priorizando a mitigação e eliminação dos riscos e melhoria contínua no processo, foi criado o processo TO-BE representado pela figura 5.3:

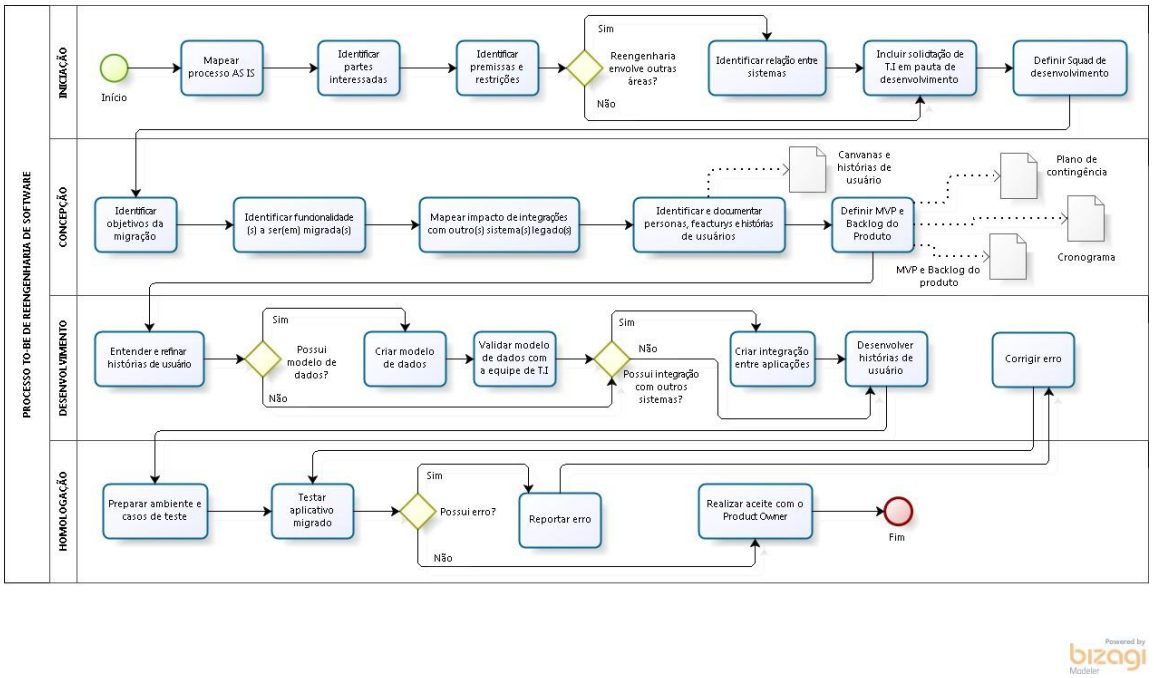


Figura 23 Processo TO-BE de Reengenharia. Fonte: Elaboração própria

Na modelagem do TO-BE, sugeriu-se uma abordagem mais ágil ao processo de reengenharia de software, utilizando algumas ferramentas e técnicas propostas por

métodos como Scrum, Kanban, PMBoK® e ferramentas como Lean Inception, que podem ser observados na figura 23.

Além da abordagem mais ágil para reengenharia de software, sugeriu-se um maior envolvimento das partes interessadas, considerando o gestor negocial como Product Owner, que por sua vez passou a integrar a equipe de desenvolvimento, denominada *Squad*.

Outra atividade proposta no processo TO-BE foi a identificação de todas as partes interessadas envolvidas em todo o processo, conforme abordado no PMBoK®, não somente das áreas de negócio como também dos gestores de tecnologia, pois caso o software a ser evoluído necessite da intervenção de outra área externa ou interna, se faz necessário um plano de ação conjunta entre as equipes e planejamento como forma de resposta ao risco de incompatibilidade ou erro em ambiente produtivo.

Após identificação das partes interessadas, identificou-se a necessidade de envolvimento delas desde o início do processo até o final como forma de mapear as suas necessidades e fazer com cada stakeholder contribua de forma efetiva, pois entende-se que dessa forma o risco de mudanças no projeto ou não cumprimento de requisito será muito menor.

Já para equipe interna, antes da elicitação de requisitos, foi determinada a criação da equipe de trabalho que irá atender a demanda em questão, denominada *squad* sendo essa composta por analista de sistemas, desenvolvedores de software, um analista de teste, um Scrum Master e um Product Owner.

Como forma de identificação de requisitos, foi mapeada a necessidade de entender o

sistema legado através de documentações existentes e principalmente a opinião de Especialistas técnicos por *Brainstorming* e a partir daí, elaborar canvas das funcionalidades contendo as possíveis regras, telas (se aplicável), mensagens e entidades de banco de dados (se aplicável).

Como documentação principal de mapeamento de necessidade e desenho de funcionalidades e suas respectivas regras, foi utilizado o conceito e elaboração de interface adaptando a ideia do Lean Inception de forma que todos os *stakeholders* consigam opinar sobre a viabilidade, necessidades e regras das funcionalidades migradas.

O processo de validação de regras, mensagens, telas (quando aplicável) e entidades de banco de dados (quando aplicável) – foi cíclico iniciando no gestor técnico, equipe de

desenvolvimento, representando o usuário final do software migrado e por último o gestor do negócio, fazendo cada vez mais a documentação e detalhamento do serviço.

Esse método foi adotado para mitigação dos riscos de mudança e não cumprimento dos requisitos, como também mitigação da inviabilidade técnica de desenvolvimento. Ele foi repetido com cada uma das equipes até finalização de todas as propostas, sempre ao final revisitando a equipe técnica de desenvolvimento para avaliação de viabilidade de desenvolvimento.

Após o requisito elicitado, a demanda foi repassada para a equipe de desenvolvimento que já possuía um conhecimento negocial por ter participado do processo de identificação dos requisitos. Em cada uma das interações com as equipes as telas (quando aplicável), as entidades de banco de dados (quando aplicável), regras e mensagens eram ajustadas.

Como forma de mitigar os riscos de mudança no software migrado após o desenvolvimento, foram realizados os testes pelo analista de sistemas responsável e não havendo necessidade de mudança, a solução prévia foi apresentada para área gestora com a validação das regras, pois caso houvesse alguma mudança, não seria necessário iniciar um ciclo de teste com a equipe qualidade, mitigando também os riscos de tempo e custo do projeto.

Após validação da área gestora, solução final, assim como também as regras, mensagens, telas (quando aplicável) e modelo de dados (quando aplicável) foram apresentados para equipe de qualidade, que por sua vez realizou os testes para identificação de erros remanescentes.

Nesse ponto, percebeu-se que não houve mais erros negociais dos sistemas. Como forma de gerenciamento de todo o processo de reengenharia do software, foram utilizadas as boas práticas como o Scrum e o quadro físico Kanban para que toda a equipe pudesse visualizar as atividades em andamento de forma rápida e constante.

O quadro Kanban auxiliou os membros da equipe no acompanhamento das atividades a visualizarem os impedimentos associados a determinadas, além do visual do andamento das atividades.

Além do processo TO-BE foi sugerido aos gestores técnico e negociais menores entregas de um produto de software viável, em forma de *sprints* de 7 há 14 dias de duração. Dentro dos ciclos de entrega, reuniões diárias com a duração de 15 minutos.

No próximo capítulo será apresentado o protótipo computacional para riscos no processo de reengenharia de software.

Capítulo 6

Proposta de ferramenta computacional

Esse capítulo descreve a realização da criação do protótipo computacional para gestão de riscos no processo de reengenharia de software.

De forma a atender o princípio de melhoria contínua, como também as necessidades da organização, foi criada uma proposta de ferramenta computacional para a gestão de riscos aplicada ao processo de reengenharia de software.

Foi solicitado pela instituição a criação da ferramenta tendo em vista a obrigação do uso da sua metodologia própria de desenvolvimento de software e adaptação com os requisitos de negócio.

O processo de criação da proposta de ferramenta computacional levou em consideração as seguintes etapas:

- Identificação das necessidades de negócio;
- Criação de protótipo não funcional;
- Validação de protótipo não funcional;
- Criação de protótipo funcional.

Nas próximas sessões, será abordado o passo a passo de construção da ferramenta.

6.1. Identificação das necessidades de negócio

O primeiro passo na construção da proposta de ferramenta computacional foi realizar o levantamento das necessidades de negócio e verificação de adequação com o esforço x necessidade.

Identificou-se que a CredTec possui uma ferramenta computacional para indicação de riscos em demandas de mudança de software. Todavia, abordando apenas a indicação de possíveis riscos associados ao processo de mudança do software em questão, o impacto aos itens de serviço associados e uma escala de probabilidade do risco se concretizar.

Esse software existente, que aqui chamaremos de “DemandasTI” como forma de proteger o sigilo da organização, é proprietário da própria instituição e apresenta uma escala de probabilidade dos riscos se tornarem incidentes como: “baixa”, “média” e “alta”.

Ainda através dessa pesquisa, identificou-se que o software “DemandasTI” permite apenas a indicação dos riscos associados aos ativos de T.I ligados ao software que está sendo alterado ou criado como: servidores, aplicativos, itens de serviços e monitoração.

Uma das observações importantes desse sistema existente, é que essa ferramenta somente permite a indicação dos riscos em software que já foram desenvolvidos, testados e homologados e serão apenas disponibilizados em ambiente de produção, não abrangendo o processo de reengenharia como um todo.

Além da ferramenta “DemandasTI”, foi avaliada a metodologia de desenvolvimento de software da instituição que conta com um documento que serve como uma norma de controle interno para o processo de engenharia e reengenharia de software da CredTec.

Um dos grandes pontos levados em consideração na tomada de decisão de construção de um protótipo computacional de gestão de riscos aplicado ao processo de reengenharia de software é que, além do sistema atual não abranger todo o processo de reengenharia, a CredTec possui sua própria metodologia de desenvolvimento, mitigando a possibilidade de adquirir ferramentas já prontas para gestão de riscos.

Dessa forma, o item de metodologia própria para desenvolvimento de software entrou como premissa para o projeto que fomentou ainda mais a necessidade de construção de uma ferramenta própria de gestão de riscos aplicada à reengenharia de software.

Esse processo de pesquisa de conteúdo envolveu apenas o especialista de riscos tendo como material estudo a ferramenta “DemandasTI” e a metodologia de desenvolvimento de software a CredTec.

Foi realizada uma entrevista inicial com o gestor técnico do produto de benefícios da previdência social para identificar as necessidades macro da tecnologia ao qual ele é responsável, em relação ao uso de uma ferramenta para gestão de riscos no processo de reengenharia de software.

As perguntas e respostas foram consolidadas para serem incluídas nessa pesquisa. As demais não foram desconsideradas para inclusão neste projeto, devido à necessidade de sigilo da organização.

Em posse dessa entrevista, foi realizada uma validação com o especialista em gestão de riscos para o desenho de um diagrama de uso da ferramenta, que serviu como guia para as demais etapas:

Figura 24 Desenho do diagrama de uso da ferramenta. Fonte: Elaboração própria

O desenho do fluxo surgiu através da necessidade de identificar o processo de uso da ferramenta e confrontá-lo com a realidade dos projetos de reengenharia de software dos produtos de conta corrente e benefícios da previdência social.

Conforme apresentando no desenho de uso da ferramenta, foram identificados e definidos dos perfis de uso da ferramenta conforme a lista:

- **Analista de riscos**: responsável por selecionar as atividades de processo conforme orientação do gestor técnico, selecionar controles existentes, elaborar a pesquisa, disponibilizar a pesquisa e registrá-la na ferramenta.
- **Gestor técnico**: responsável por registrar o grau de probabilidade e impacto na ferramenta, como também avaliar os riscos através do gráfico de riscos.

Identificação dos requisitos de sistema

Após a criação do desenho, foi realizada uma nova apresentação para o gestor técnico para definição dos requisitos de sistema, sendo esses: funcionais e não funcionais.

Utilizando a técnica de *Brainstorming* e com apoio do Analista de riscos foram identificados os seguintes requisitos do sistema, gerando a seguinte lista:

• Requisitos funcionais: são requisitos do que o sistema deve fazer, ou seja, as funcionalidades.

– Cadastro de riscos identificados
– Cadastro de base de conhecimento
– Cadastro de Análise de Riscos
– Cadastro da Avaliação de Riscos

• Requisitos não funcionais: está relacionado ao "como o sistema dever fazer", diretamente ligado a restrições que devem ser consideradas.

– Utilizar metodologia de desenvolvimento da própria instituição
– Utilizar padrão de design homologado pela instituição

Após a definição e validação dos requisitos de sistema, juntamente com desenho de utilização da ferramenta, iniciou-se o processo de criação do protótipo não funcional.

6.2. Elaboração de protótipo não funcional

Para o processo inicial de elaboração do protótipo, foram utilizadas técnicas de *Brainstorming* e pesquisa de conteúdo que são métodos de elicitação de requisitos utilizados em processos como Rational Unified Process (RUP) e *Design* User Experience (UX).

Em posse das informações identificadas através da pesquisa de conteúdo, entrevista e com o uso da ferramenta Enterprise Architect (E.A) e o Excel, foi criada uma primeira versão de um protótipo não funcional para o sistema de gestão de riscos.

O E.A e o Excel foram escolhidos para essa tarefa, tendo em vista que são ferramentas utilizadas pela CredTec e incluídas na sua própria metodologia de desenvolvimento de software.

As próximas sessões descrevem a criação do protótipo não funcional para cada um dos requisitos funcionais identificados anteriormente.

6.2.1 Protótipo não funcional da tela de controle

A primeira funcionalidade foi "manutenção de controles" que permite a inclusão, alteração, exclusão e pesquisa de controles associados aos riscos, que teve como base de requisitos identificada através da pesquisa de conteúdo, o modelo do CGU para criação de bases de conhecimento, a ISO 31000 e própria ferramenta "DemandasTI".

A figura a seguir apresenta a tela de pesquisa de controles conforme figura 25:

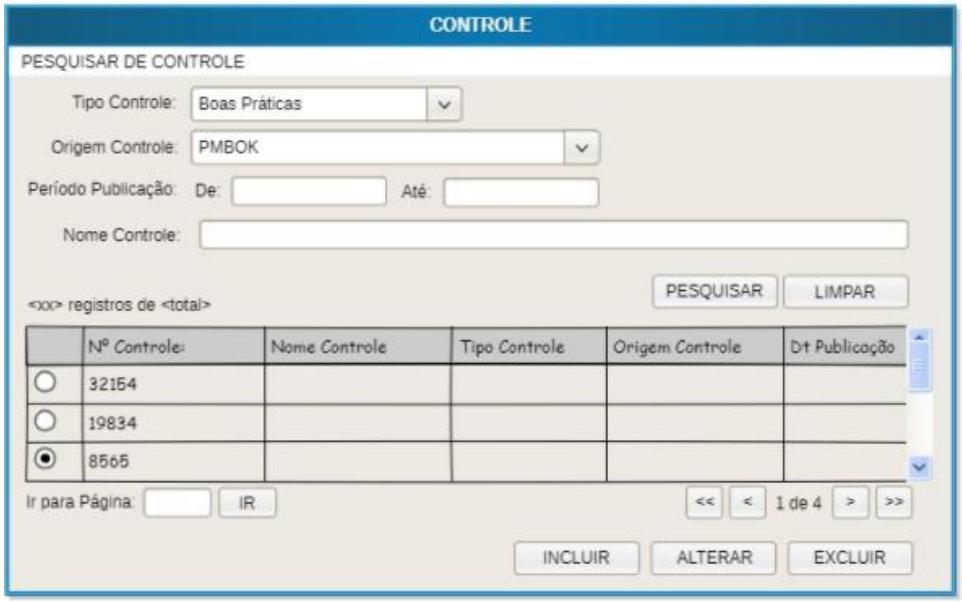


Figura 25 Protótipo não funcional da tela de controle. Fonte: Elaboração própria

A proposta de uma funcionalidade manutenção de controles, através das boas práticas de gestão de riscos, foi a de permitir ao gestor de riscos que irá trabalhar no processo de reengenharia de software, cadastrar via ferramenta os controles identificados que podem ser aplicados ao processo de reengenharia de software.

A figura a seguir apresenta o protótipo de tela de inclusão de controles que também pode ser observada na tela de alteração de controles conforme figura 16:

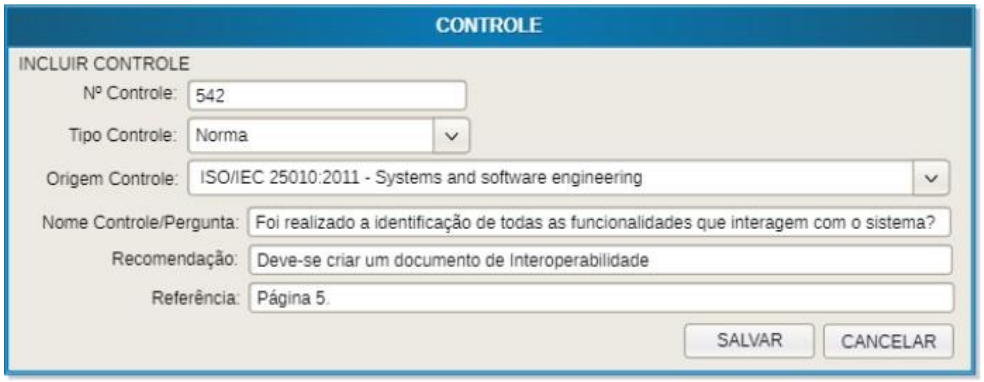


Figura 26 Protótipo da tela de inclusão de controle. Fonte: Elaboração própria

O protótipo da funcionalidade de cadastro de controles foi criado através da necessidade observada na ferramenta “DemandasTI” de cadastramento de uma base de conhecimento de controles aplicados a reengenharia de software.

6.2.2 Protótipo não funcional da tela de risco

A segunda tela proposta no protótipo não funcional é a de cadastro de riscos. Para sua concepção, foram utilizados os requisitos identificados através do levantamento de requisitos iniciais e opinião de especialista.

O protótipo dessa tela foi desenvolvido utilizando o Excel, conforme indicado na figura 27:

Figura 27 Protótipo não funcional da tela cadastro de riscos. Fonte: Elaboração própria

Para o protótipo, conforme figura 6.4, foram considerados os campos de identificação do risco, nome do risco, categoria do risco, descrição do risco e controle associado.

6.2.3 Protótipo não funcional da tela de resposta de entrevista

A terceira tela prototipada foi de resposta de entrevistas, servindo como uma funcionalidade para consulta dos questionários respondidos pelos entrevistados, conforme figura 28:

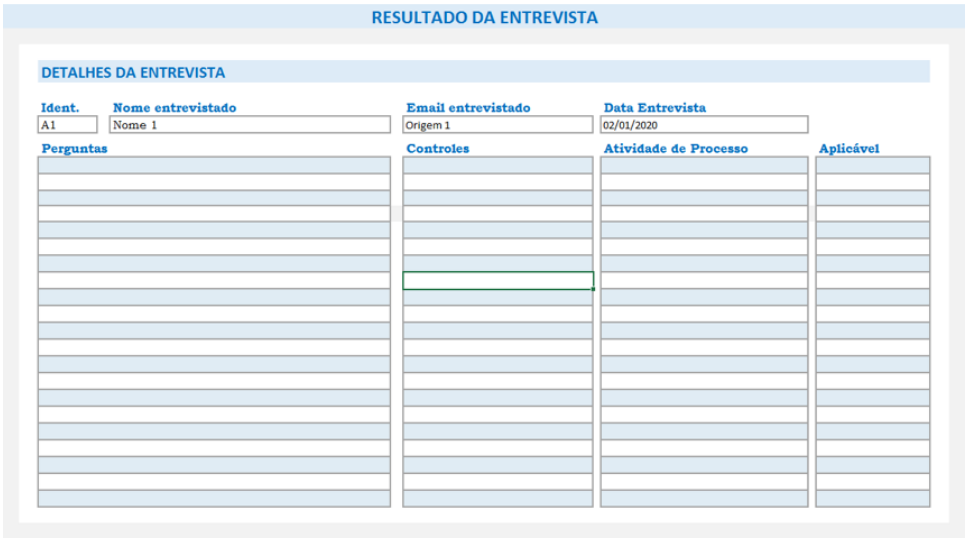


Figura 28 Protótipo não funcional da tela resposta de entrevistas. Fonte: Elaboração própria

A tela de consulta de entrevista, conforme figura 6.5, são apresentados os campos de identificação da entrevista, nome do entrevistado, e-mail e data da entrevista.

Ela também é composta por uma *grid* contendo as perguntas, seus respectivos controles, atividade ao qual a pergunta se encaixa e a resposta do entrevistado.

6.2.4 Protótipo não funcional da tela de análise de riscos

A quarta tela representa o protótipo da funcionalidade de análise dos riscos, sendo representada pela figura 6.6:

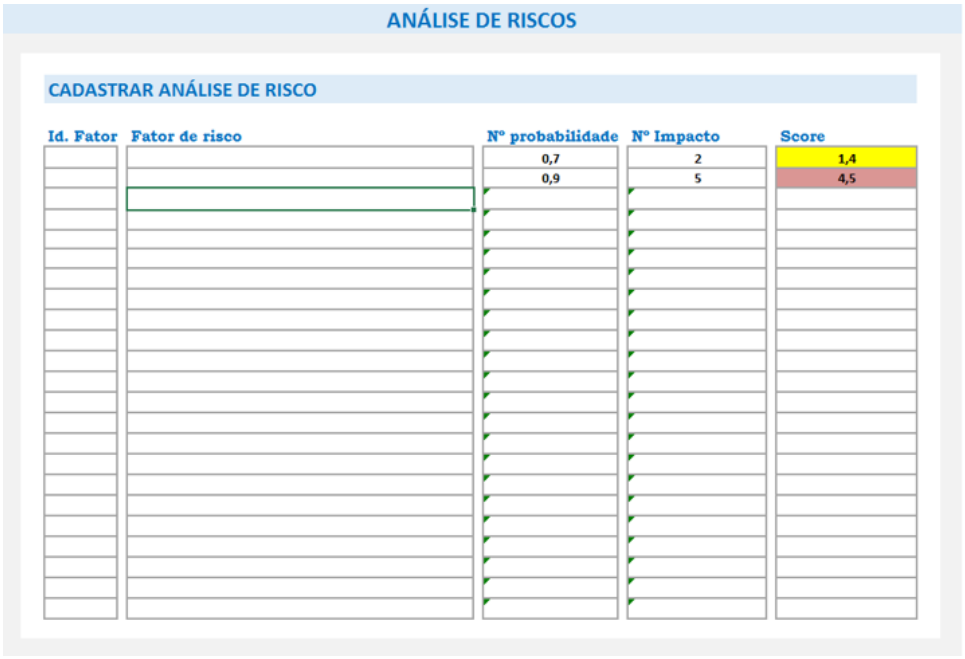


Figura 29 Protótipo não funcional da tela de análise de riscos. Fonte: Elaboração Própria

Os requisitos utilizados para o desenvolvimento dessa tela abrangem as necessidades de negócio identificadas na reunião de levantamento de requisitos, opinião de especialista e a indicação da matriz de probabilidade de impacto proposta pela ISO 31010.

O protótipo é composto por um campo de identificação dos riscos, fator de risco, o grau de probabilidade e o grau de impacto.

Como regra, os campos de probabilidade e impacto são multiplicados entre si para apresentar um valor de risco indicado no campo score.

A proposta geral do protótipo não funcional foi utilizada como um documento inicial para o *Brainstorming* de funcionalidades com o gestor técnico do produto de benefícios da previdência social, além das possíveis necessidades identificadas través do software "DemandasTI".

Em posse desse primeiro protótipo, foi realizado um *Brainstorming* de funcionalidades com um dos gestores técnicos que será apresentado no tópico seguinte.

6.2.5 *Brainstorming* de funcionalidades

O protótipo não funcional da tela de controle, serviu de insumo para debate na identificação das demais funcionalidades.

Após realizar uma entrevista inicial com o gestor técnico, modelar o uso da ferramenta e criar um protótipo inicial, foi criada uma apresentação em powerpoint sobre o uso da proposta de instrumento computacional para gestão de riscos no processo de reengenharia de software e a importância dessa prática.

O intuito da apresentação é fazer uma introdução ao assunto do *Brainstorming* que

envolveu o gestor técnico do produto de benefícios da previdência social, dois desenvolvedores e dois analistas de sistemas, selecionados pelo próprio gestor técnico.

Em posse dos documentos iniciais e após apresentação da proposta de instrumento computacional, foi dado início ao *Brainstorming* onde os participantes tiveram a oportunidade de expor as suas ideias sobre o uso da ferramenta no dia a dia do processo de reengenharia de software e quais seriam as principais funcionalidades.

Através deste *Brainstorming*, foi identificada a necessidade de funcionalidades:

- Consultar controles;
- Consultar riscos;
- Consultar entrevistas
- Registrar análise de riscos
- Consultar indicadores de grau dos possíveis riscos identificados

Após a realização do *Brainstorming*, os requisitos foram anotados e sintetizados na renovação do primeiro protótipo apresentado.

Dessa forma, foi criada uma segunda proposta de protótipo de instrumento computacional para gestão de riscos no processo de reengenharia de software que será descrita no próximo tópico

6.3. Elaboração de protótipo funcional

Nessa sessão estão descritas as telas prototipadas e principais requisitos identificados.

Em posse dos requisitos identificados no *Brainstorming* de funcionalidades e avaliando o documento de padrão de design das funcionalidades da CredTec, foi criada uma segunda proposta de protótipo de instrumento computacional para gestão de riscos no processo de reengenharia de software.

Esse protótipo levou em consideração, além dos itens citados anteriormente, o desenho do fluxo de uso da ferramenta e opinião de especialista em gestão de riscos.

Como forma de preservar a privacidade da instituição, a proposta de instrumento computacional foi denominada “Risk Tools”.

Com o uso da ferramenta Excel, foram criadas todas as telas que poderiam ser utilizadas na gestão de riscos aplicada ao processo de reengenharia de software.

6.3.1 Tela inicial

Inicialmente foi desenhada uma tela com os menus para as demais funcionalidades, sendo essa denominada "Home".

A figura a seguir mostra o protótipo da tela inicial da ferramenta conforme figura 30:

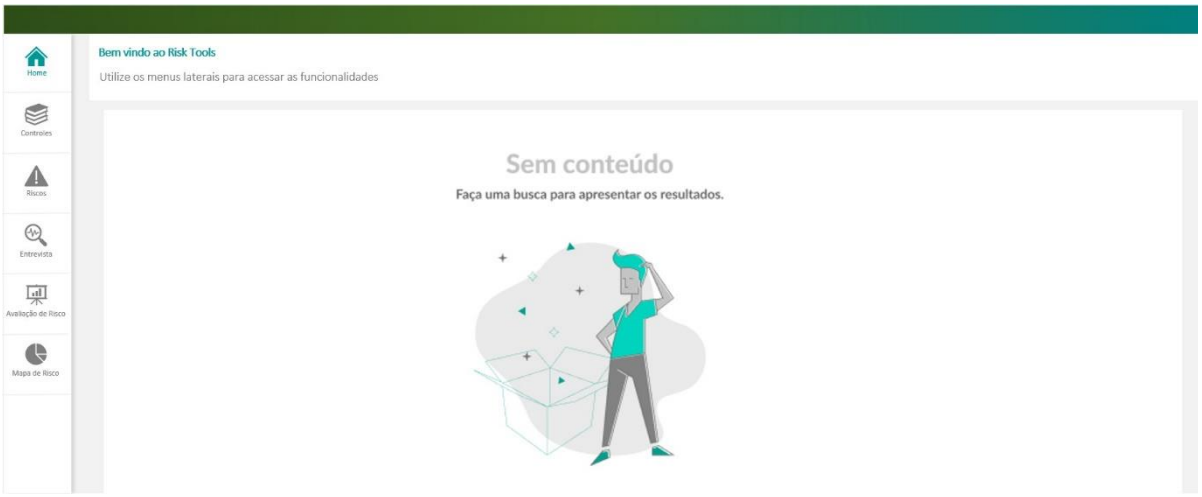


Figura 30 Protótipo Funcional da tela Home da RiskTolls. Fonte: Elaboração própria

Essa tela inicial apresenta apenas os menus das demais telas que serão utilizadas que são: Menu Inicial, Controles, Riscos, Entrevistas, Avaliação de Riscos e Mapa de Riscos.

6.3.2 Controles

A funcionalidade de controle foi dividida em duas partes, uma contendo a consulta dos controles já existentes na base de dados e outra para inclusão dos novos controles.

Inicialmente foi definida como persona, ou usuário da tela de controles, o próprio gestor de riscos que é responsável pelo levantamento do conhecimento com os controles internos e externos, como também o registro deles na RiskTolls.

A figura 31 apresenta o protótipo da tela para consulta de controles, onde apresenta dados mascarados apenas para validação do protótipo:

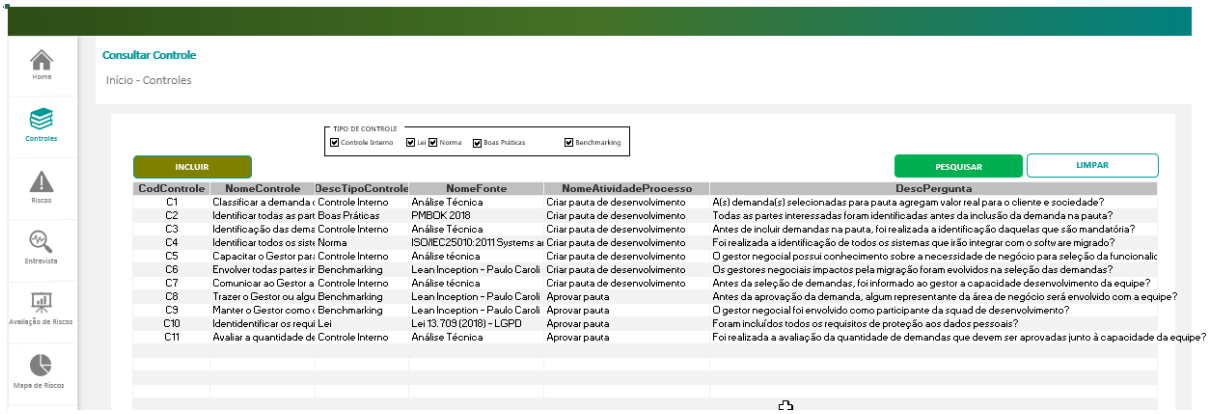


Figura 31 - Tela de consulta de controles do protótipo funcional. Fonte: Elaboração própria

Essa tela permite a consulta de controle existentes através dos filtros de tipo de controle que são: “Controles Internos, “Leis”, “Normas”, “Boas Práticas” e “*Benchmarking*”. Além desses filtros, é possível realizar a inclusão de um novo controle acionando a opção “Incluir”.

A figura 32 apresenta a tela do protótipo de inclusão de controle:

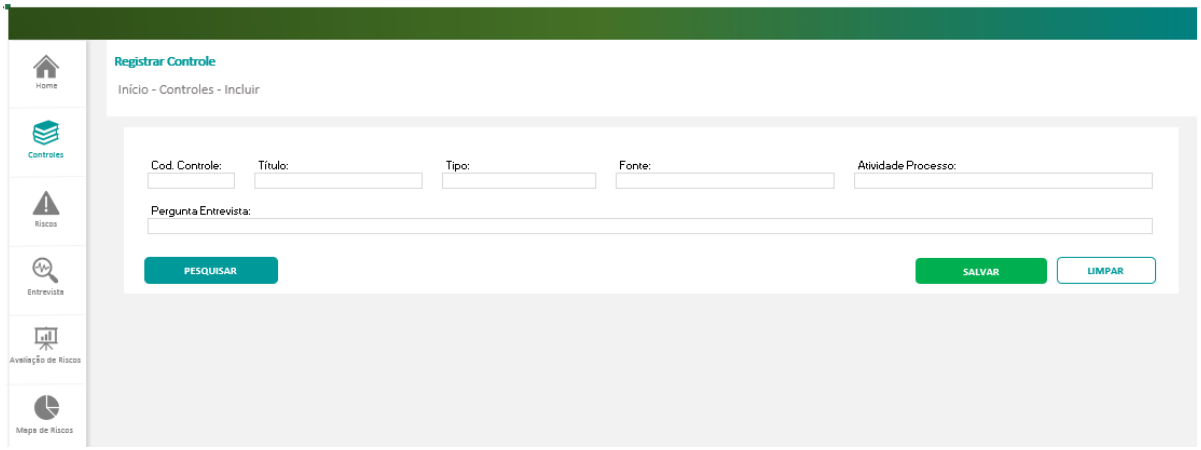


Figura 32 - Tela de inclusão de controle do protótipo funciona. Fonte: Elaboração própria

A proposta de tela de inclusão de controle é bem simples, permitindo ao operador realizar o registro de um novo controle na base de dados da ferramenta incluindo os dados de código do controle, título, tipo, fonte, atividades de processo ao qual ele está associado e pergunta que pode ser utilizada para elaboração do questionário (será abordado mais à frente).

Decidiu-se que os dados de controle serão mantidos via registro manual pelo analista de riscos, tendo em vista que o perfil de gestão técnica terá acesso apenas a visualização desses dados.

6.3.3 Riscos

Após o cadastro de controle, a próxima funcionalidade proposta é a de registro de riscos, sendo possível realizar a consulta dos riscos existentes na base de dados da RiskTolls como também a inclusão de um novo risco.

A próxima tela apresenta a proposta de funcionalidade de consulta de risco, conforme figura 33:

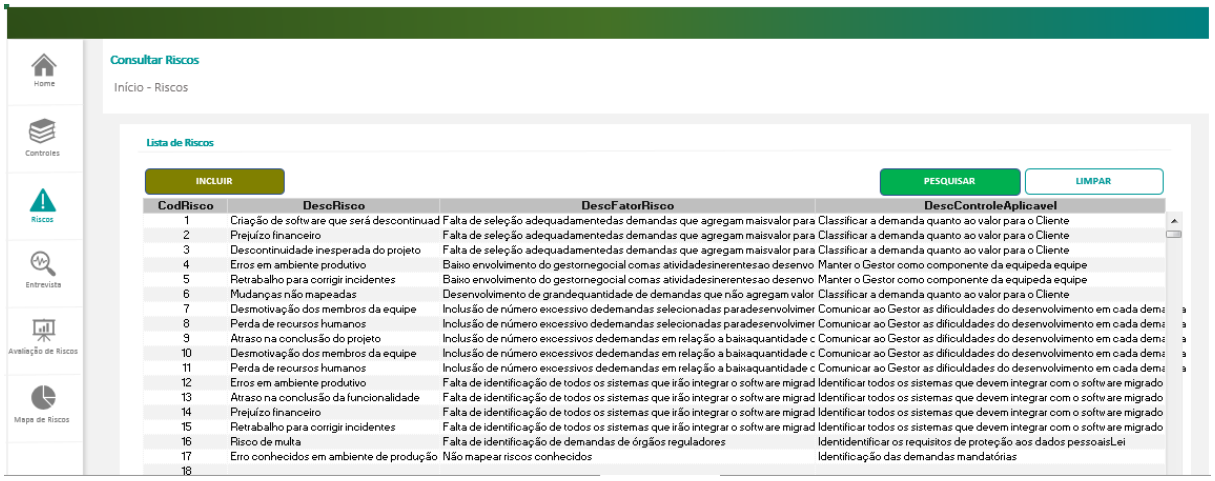


Figura 33 - Tela de consulta de riscos do protótipo não funcional. Fonte: Elaboração própria

Na tela de consulta de risco, o usuário poderá pesquisar de forma macro por todos os riscos existentes na base de dados, sem a necessidade de informar um determinado filtro.

Para inserir um novo risco, foi incluindo um botão de “incluir” que ao ser acionado direciona o operador do Risk Tolls para a tela de cadastro de riscos apresentada na figura 34:

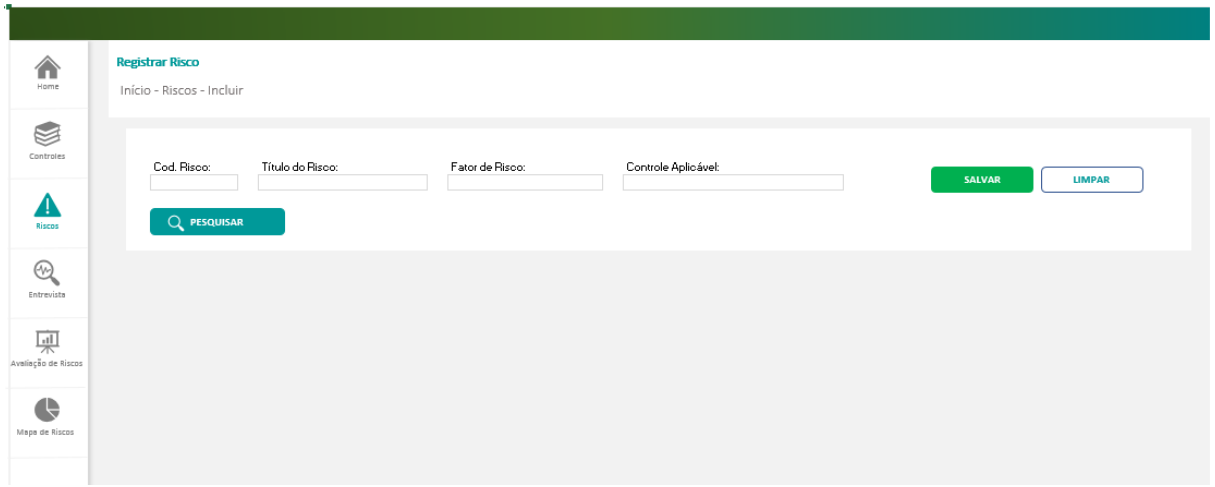


Figura 34 - Tela de inclusão de riscos do protótipo funciona. Fonte: Elaboração própria

Na inclusão de riscos, a proposta é que o operador consiga, de forma simples, inserir um novo risco para registro na base de dados da Risk Tolls, informando os campos: código do risco, título do risco, fator de risco e possíveis controles aplicáveis.

No momento do registro, foi decido que será exibido os controles aplicáveis registrados anteriormente, permitindo ao operador selecionar o controle desejado.

6.3.4 Entrevistas

Após realizar o cadastro dos riscos, a próxima funcionalidade prototipada foram as telas entrevista, dívidas nas funcionalidades de cadastro e consulta de entrevista.

Conforme os requisitos, a tela de consulta de entrevista deve permitir ao operador consultar as perguntas de uma determinada entrevista respondida e já cadastrada na base de dados da Risk Tolls, sendo possível filtrar pelo perfil de quem respondeu o questionário, conforme figura 35:

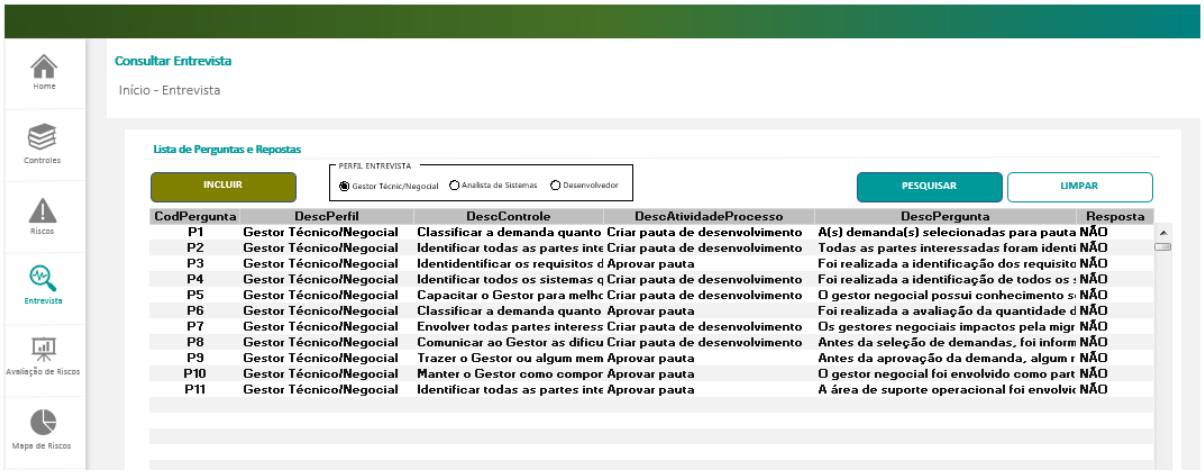


Figura 35 - Tela de consulta de entrevista do protótipo funcional. Fonte: Elaboração própria

Além dos campos de pesquisa, conforme elicitação de requisitos, foi desenhada uma *grid* em que operador pudesse ver as colunas de código da pergunta, descrição do perfil, descrição do controle, atividade do processo ao qual foi aplicada, a descrição da pergunta e a resposta de quem respondeu o questionário informando se determinado controle foi ou não aplicado a sua atividade do processo de reengenharia de software.

Conforme necessidades mapeadas, foi desenhado um botão de “incluir” que a ser acionado, direciona o usuário/operador para a tela de cadastro de pergunta de entrevista, apresentada na figura 36:

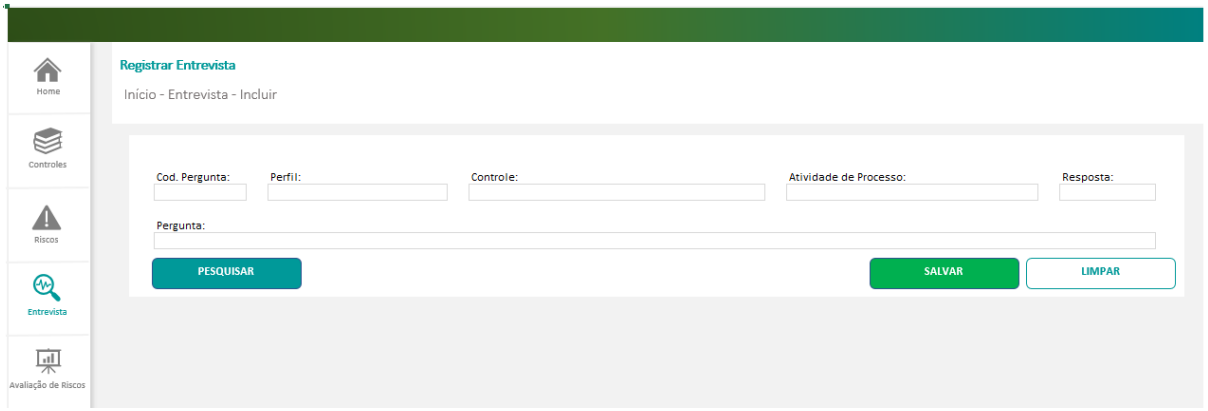


Figura 36 - Tela do protótipo funcional de inclusão de pergunta de entrevista. Fonte: elaboração própria

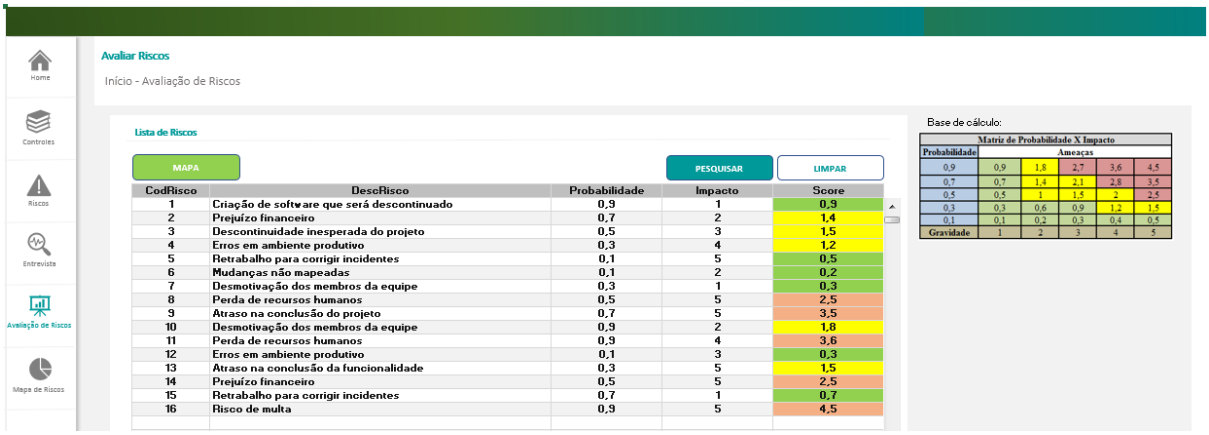
Foi identificada a necessidade de dispor na tela de inclusão de pergunta de entrevista, os mesmos campos exibidos na tela de consulta.

6.3.5 Avaliação de riscos

Conforme elicitação de requisitos, a penúltima tela desenhada no protótipo funcional é de Avaliação dos riscos.

Essa funcionalidade exibe todos os riscos identificados através do questionário disparado para o entrevistado. Uma vez que a resposta for negativa, ou seja, o entrevistado responder como “NÃO” e após o registro da pergunta na ferramenta, o risco será exibido na tela de avaliação de riscos.

Uma vez os dados de riscos sendo exibidos, operador com perfil de gestor poderá indicar o grau de riscos de cada registro através de valores associados à probabilidade e impacto, conforme figura 37:



CodRisco	DescRisco	Probabilidade	Impacto	Score
1	Criação de software que será descontinuado	0,9	1	0,9
2	Prejuízo financeiro	0,7	2	1,4
3	Descontinuidade inesperada do projeto	0,5	3	1,5
4	Erros em ambiente produtivo	0,3	4	1,2
5	Retrabalho para corrigir incidentes	0,1	5	0,5
6	Mudanças não mapeadas	0,1	2	0,2
7	Desmotivação dos membros da equipe	0,3	1	0,3
8	Perda de recursos humanos	0,5	5	2,5
9	Atraso na conclusão do projeto	0,7	5	3,5
10	Desmotivação dos membros da equipe	0,9	2	1,8
11	Perda de recursos humanos	0,9	4	3,6
12	Erros em ambiente produtivo	0,1	3	0,3
13	Atraso na conclusão da funcionalidade	0,3	5	1,5
14	Prejuízo financeiro	0,5	5	2,5
15	Retrabalho para corrigir incidentes	0,7	1	0,7
16	Risco de multa	0,9	5	4,5

Probabilidade	Ameaças				
0,9	0,9	1,8	2,7	3,6	4,5
0,7	0,7	1,4	2,1	2,8	3,5
0,5	0,5	1	1,5	2	2,5
0,3	0,3	0,6	0,9	1,2	1,5
0,1	0,1	0,2	0,3	0,4	0,5
Gravidade	1	2	3	4	5

Figura 37 - Tela do protótipo funcional de avaliação de riscos. Fonte: Elaboração própria.

Conforme identificação das necessidades dos usuários e opinião de especialista, a tela de avaliação de riscos irá exibir a listagem dos riscos e como forma de apoio, será disposta a matriz de probabilidade indicada pelo PMBoK® e ISO 31010 como ferramenta para avaliação de riscos.

Para fins de cálculo do campo score, os valores de probabilidade e impacto são multiplicados entre si, exibindo então o grau do risco.

6.3.6 Mapa de riscos

O último protótipo desenhado é o de mapa de risco que, conforme necessidades dos usuários identificadas e a opinião dos especialistas, deve exibir um mapa dos riscos após suas avaliações pelo gestor, em um mapa, conforme a figura 38:

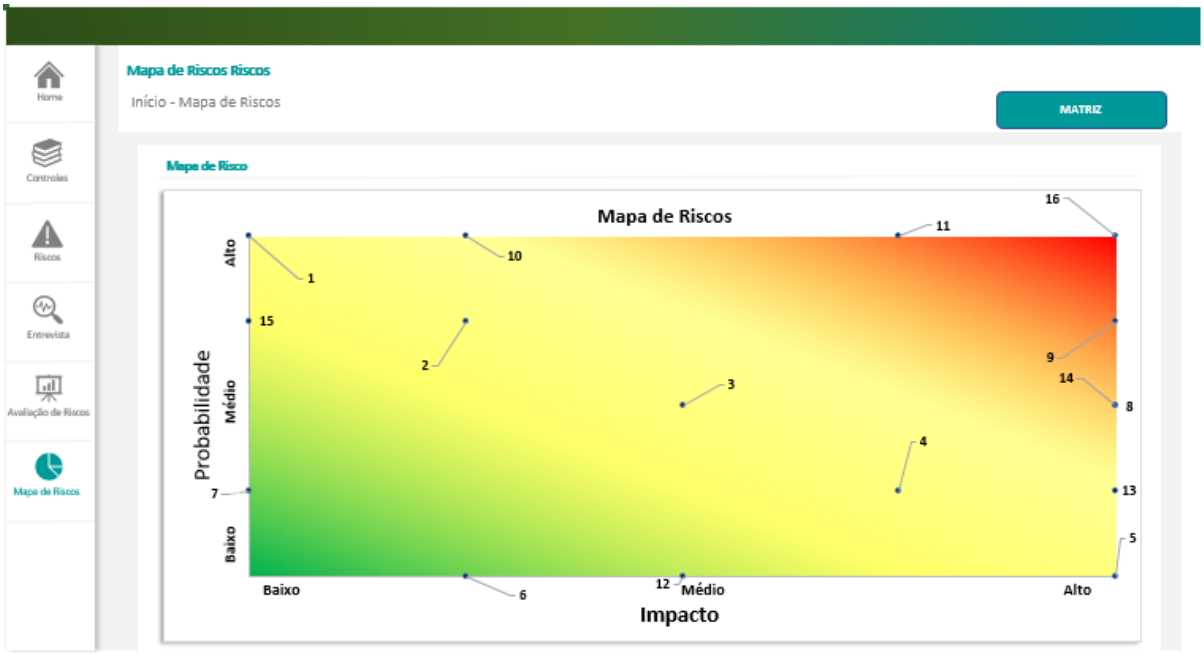


Figura 38 - Tela do protótipo funcional contendo o mapa de riscos. Fonte: Elaboração própria

Conforme as partes interessadas, a indicação dessa tela é que sejam exibidos todos os riscos que foram avaliados e sua disposição no mapa de riscos pelos graus: baixo indicado pela cor verde, médio indicado pela cor amarela e alto indicado pela cor vermelha.

As partes interessadas informaram que a programação dessa tela de mapa de riscos deve indicar a posição exata do risco inerente ao grau de risco em si.

Após a construção de todo o protótipo funcional, foi realizada a sua validação, conforme apresentação na próxima sessão.

6.4. Validação do protótipo funcional

Após elaboração do protótipo funcional contendo a aplicação das regras identificadas no primeiro levantamento de requisitos, foi realizada uma nova reunião com o gestor técnico e especialista em gestão de riscos para realizar uma nova validação das telas e aplicação de regras.

O intuito dessa nova validação do protótipo seria gerar o primeiro produto mínimo viável de proposta da ferramenta para gestão de riscos aplicada ao processo de reengenharia de software.

O especialista em gestão de riscos solicitou uma amostragem do uso da ferramenta já utilizando algumas atividades de processos que tiveram seus riscos identificados no Capítulo 5.

As próximas sessões discorrem sobre a validação prática da proposta de ferramenta computacional.

Para essas etapas, foi utilizado também como parâmetro de entrada o desenho do uso da ferramenta realizado apresentado na figura 24.

6.4.1 Criação de questionário

Nessa sessão estão contidos todos os passos realizados para geração do questionário para identificação das vulnerabilidades no processo de reengenharia de software.

Seleção de atividade de processo

Para amostragem de funcionamento e validação da proposta de ferramenta computacional, foi selecionado pelo especialista em gestão de riscos três atividades de processo AS IS de reengenharia de software relacionadas ao perfil de gestor técnico/negocial, conforme figura:

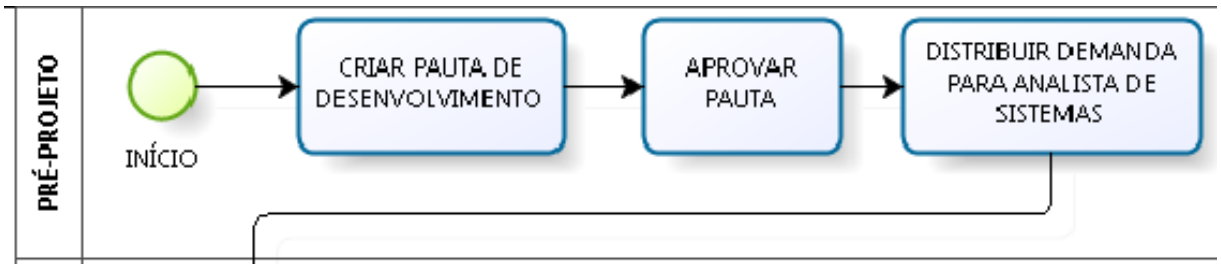


Figura 39 Atividade de Processos Selecionadas para Validação da Ferramenta.

Como as atividades são comuns entre os projetos de reengenharia de software, apesar de serem as mesmas atividades, foi selecionado outro gestor técnico para responder a pesquisa, só que dessa vez associado ao produto de conta corrente.

A proposta foi avaliar a experiência com uso da ferramenta alinhada ao processo de gestão de riscos na reengenharia de software.

A seleção das atividades de processo foi realizada após avaliação crítica do gestor técnico, elencando as mais relevantes para o processo de reengenharia de software.

Seleção de controles

A seleção dos controles foi baseada no levantamento realizado no processo de gestão de riscos apresentado no capítulo 6 dessa pesquisa.

Foram elencados todos os controles associados as atividades de "Criação de pauta de desenvolvimento", "Aprovar pauta" e "Distribuir demanda para analista de sistemas".

Como um primeiro passo, os controles foram registrados de forma manual pelo analista de riscos na proposta de ferramenta computacional e associado a eles, foi indicada uma pergunta para cada um que serviu de insumo para geração do questionário, conforme exemplo do quadro 11:

Quadro 11 Seleção do controle e associação de questionário. Fonte: Elaboração própria.

Cód.	**Controle**	**Fatores de Riscos**	**Riscos**	***Questionário***
C01	Identificar demandas que mais agregam valor para o cliente e sociedade.	F02 - Falta de seleção adequadamente das demandas que agregam maisvalor para o negócio. F03 Gestor Negocial influenciado negativamentepelas partes interessadas.	R01: Descontinuidade inexperada do projeto; R02: Erros em ambiente produtivo; R03: Atraso na conclusão do projeto. R04: Criação de softwareque será descontinuado; R05: Prejuízo financeiro; R06: Desmotivação dos membros da equipe.	A(s) demanda(s) selecionadas para pauta agregam valor real para o cliente e sociedade?
C10	Identificar todas as necessidades, premissas, restrições e requisitos iniciaisdo projeto.	F01 Falta de envolvimento das partes interessadas na definição das demandas que serão incluidas na Pauta.	R01: Descontinuidade inexperada do projeto; R03: Atraso na conclusão do projeto. R04: Criação de softwareque será descontinuado; R05: Prejuízo financeiro;	Todas as partes interessadas foram identificadas antes da inclusão da demanda na pauta?

O questionário foi criado pelo próprio analista de riscos, com apoio do especialista em gestão de riscos, sendo esse relacionado diretamente ao controle em questão.

À parti da documentação do questionário e associação deles aos controles, foi possível gerar um questionário que contempla a identificação de vulnerabilidades no processo de reengenharia de software conforme resposta do entrevistado.

Elaboração da pesquisa

Para elaboração da pesquisa, foi utilizada a ferramenta "GoogleForms" e diferentemente da pesquisa anterior, os questionários já possuíam os controles identificados e associados a cada pergunta.

Com apoio do especialista em gestão de riscos, foi elaborado um questionário contendo os controles aplicados às atividades de processo de reengenharia de software selecionadas pelo gestor técnico.

A seguinte figura 40 apresenta o modelo da pesquisa gerado:

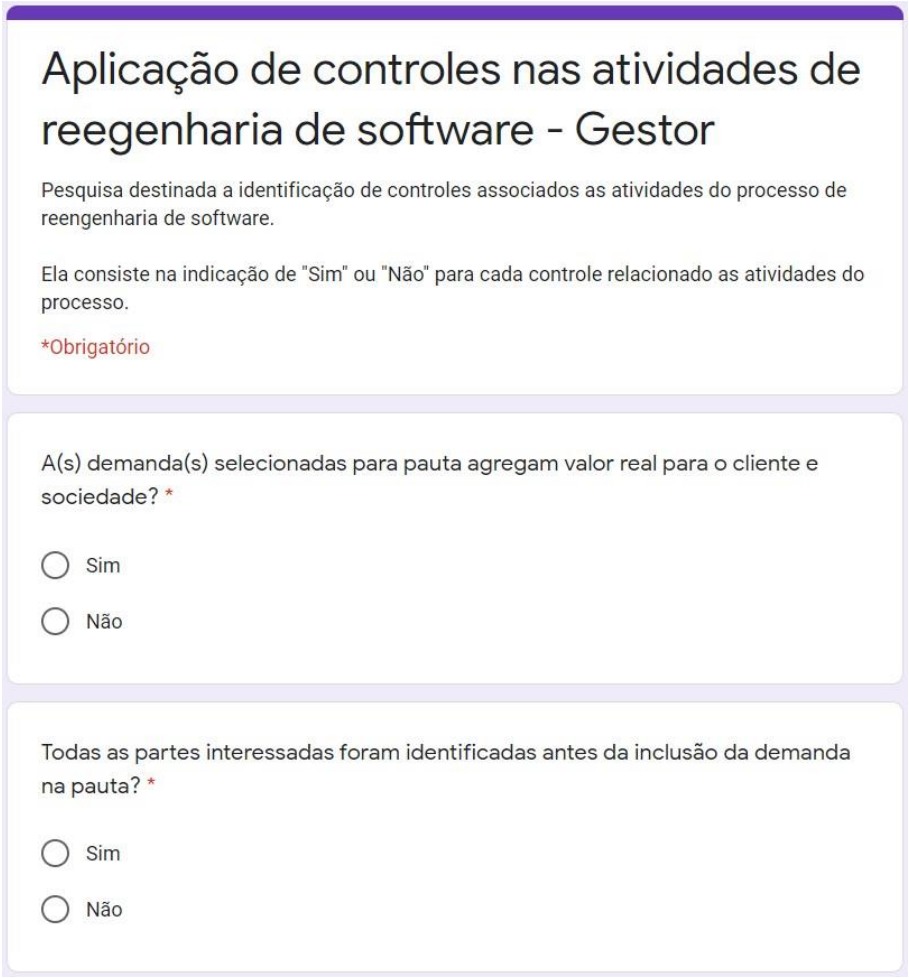


Figura 40 Figura 6.14: Modelo de Pesquisa. Fonte: Elaboração própria

O modelo de pesquisado foi criado tendo como base a flexibilização de sua aplicação em outros projetos.

O questionário criado buscou apenas identificar se um determinado controle foi aplicado ou não às atividades de processo de reengenharia de software associadas ao perfil do gestor.

A intensão do mapeamento desses dados é identificar as possíveis vulnerabilidades nas atividades do processo executadas pelo entrevistado.

Disponibilização da pesquisa

A pesquisa foi disponibilizada via link gerado pela ferramenta "Google Forms"e destinada a um dos gestores técnicos do projeto CCO, responsável pelo processo de reengenharia de software.

Foi realizada uma orientação ao gestor técnico quanto às respostas do questionário e solicitado que fosse baseada em sua experiência atual.

Registro da Pesquisa

Para o registro da pesquisa, foi gerada a exportação das respostas em ".csv" da ferramenta "Google Forms".

Como mapeado anteriormente, o registro das entrevistas seria realizo de forma manual pelo analista de riscos e tratada posteriormente.

Os dados foram tratados e as respostas consolidadas para registro na proposta de ferramenta computacional dessa pesquisa.

Como houve a documentação da associação dos controles e riscos, através da resposta

da pesquisa, foi possível identificar todos os riscos relacionado a cada vulnerabilidade nas atividades do processo de reengenharia de software conforme perfil do entrevistado.

Na próxima sessão é apresentada a análise e avaliação dos riscos com o uso da ferramenta.

6.4.2 Análise e avaliação dos riscos na ferramenta

A partir da experiência no capítulo anterior, foi solicitado que o gestor técnico do produto conta corrente, realizasse a análise e avaliação dos riscos na ferramenta proposta.

Uma vez que os riscos foram identificados, foi disponibilizada uma listagem deles na própria ferramenta para que o gestor, com apoio do especialista em gestão de riscos, pudesse incluir um grau de probabilidade e impacto para cada um desses riscos.

Para auxiliar no processo de julgamento do grau de impacto, foi disponibilizado para o gestor técnico o registro dos riscos na própria ferramenta.

A partir dessa análise, foi possível que o gestor técnico tivesse a visão geral dos riscos associados as suas atividades do processo de reengenharia de software, possibilitando que ele pudesse elencar os riscos mais prioritários para tratamento.

Capítulo 7

Conclusões finais

A complexidade do cenário atual de tecnologia da informação tem evoluído constantemente, o que demanda o aumento da qualidade software e melhoria da experiência de usuários perante os aplicativos. Para acompanhar a realidade desse cenário, as organizações buscam realizar a reengenharia de software para migrar os seus sistemas legados, adequando-os às novas tecnologias.

Todavia, apesar dos esforços empenhados para garantir uma qualidade aceitável desses aplicativos em ambiente produtivo, existem vários riscos que permeiam o processo de reengenharia de software.

Nesse contexto, a CredTec também realiza a migração de seus sistemas legados para novas linguagens, banco de dados e plataformas, de maneira a garantir que seu negócio evolua, mantendo as regras negociais preservadas. O maior objetivo da CredTec nesse sentido é garantir que os sistemas legados que estão sendo migrados, apresentem um grau satisfatório de qualidade em ambiente produtivo.

Dessa forma, o objetivo principal dessa pesquisa foi realizar a análise e melhoria do processo de reengenharia de software, tendo como apoio um instrumento computacional para gerenciamento dos riscos do processo.

Concluímos neste trabalho que foi possível realizar o gerenciamento dos riscos no processo de reengenharia de software, por meio do instrumento computacional aqui proposto, plicando essa ferramenta nos projetos existentes de reengenharia de software da CredTec. Para atingir esse objetivo geral, realizamos a modelagem do processo atual (AS-IS) de reengenharia de software da CredTec para identificação e documentação das reais atividades do processo. Por meio da modelagem do processo AS-IS, a CredTec formalizou o esse fluxo de trabalho, que antes acontecia de maneira informal.

Com um processo AS-IS de reengenharia de software formalizado, foram evidenciados pontos críticos no processo, indicando então uma necessidade expressiva de gerenciamento de riscos.

Diante disso, realizamos a identificação e aplicação de modelo para gestão de riscos no processo de reengenharia de software. Para tanto, foram utilizadas ferramentas de

identificação dos riscos como entrevista, e opinião de especialistas da empresa, resultando na identificação dos riscos do processo de reengenharia de software.

Como forma de apoio para a CredTec na gestão de riscos, seguiu-se com a identificação e documentação da base de conhecimento com controles aplicados aos riscos no processo de reengenharia de software daquele contexto.

Assim foi realizada a identificação de possíveis ações contingenciais que pudessem servir como controles internos, junto à equipe de desenvolvimento dos projetos de reengenharia de software, composta por membros chaves selecionados pelos gestores técnicos. Em seguida realizamos a identificação e documentação dos controles externos aplicáveis aos riscos identificados no processo de reengenharia de software como: leis, normas, boas práticas e *Benchmarking*.

Após alcançar o primeiro objetivo específico com modelagem do processo AS-IS, e após a identificação dos riscos relacionados a reengenharia de software e possíveis controles aplicáveis, foram propostos métodos de apoio aos gestores dos projetos de reengenharia para análise e avaliação dos riscos, para o segundo objetivo.

Foi realizada então uma revisão bibliográfica para identificar dentro da literatura possíveis métodos de análise e avaliação de riscos passíveis de serem aproveitados em projetos de reengenharia de software. Além dessa pesquisa, foi utilizada a opinião de especialistas, convergindo então para métodos de análise de avaliação de riscos, como matriz de probabilidade e impacto.

A partir da aplicação do modelo para gestão de riscos na reengenharia de software, foi possível validar que a CredTec ganhou maturidade nas suas ações contingenciais, permitindo evidenciar, analisar e tratar possíveis riscos do processo.

Como forma de resposta aos riscos identificados no processo AS-IS, identificou-se a necessidade de redesenhar um processo melhorado TO-BE.

Diante do cenário da CredTec e a partir da gestão dos riscos de reengenharia de software, sugeriu-se a remodelagem do processo AS-IS, utilizando a engenharia ágil de software com abordagens como Kanban, Scrum e Lean Inception.

Vale ressaltar que a necessidade da organização é cíclica e evolutiva, sendo que cada projeto de reengenharia possui uma especificidade. Dessa maneira, o redesenho do processo pode ser monitorado, avaliado e evoluído constantemente.

Após o redesenho do processo TO-BE, verificou-se junto à área gestora a necessidade de desenvolver um instrumento computacional que pudesse auxiliar a CredTec na automação e organização sistematiza do gerenciamento de riscos do processo de

reengenharia. Nesse contexto, foram realizadas reuniões de levantamento de requisitos para identificar as necessidades de negócios e possíveis ferramentas já existentes para gerenciamento de riscos dentro da CredTec.

Concluímos que não existe uma ferramenta especifica o para gerenciamento dos riscos na reengenharia de software, indicando a necessidade de se desenvolver um instrumento computacional para essa finalidade.

Para tanto, foram utilizas técnicas de identificação de requisitos como entrevista, e análise documental, buscando gerar insumos para criação do instrumento computacional para gerenciamento de riscos no processo de reengenharia de software. Isso foi feito por meio de reuniões e coleta de opinião de especialistas, desenho de um diagrama contendo o esquema de funcionamento da ferramenta, servindo como insumo para identificação dos requisitos funcionais.

A partir das reuniões com especialistas, análise documental e aplicando o modelo de gestão de riscos obtido no segundo objetivo específico, foi proposto um primeiro protótipo não funcional, contendo quatro telas que contemplavam as funcionalidades de cadastro de controles, riscos, entrevista e análise dos riscos. Esse protótipo não funcional foi apresentado para as partes interessadas, que apoiaram no processo de desenvolvimento do instrumento computacional, objetivando a validação dos requisitos iniciar e identificar outras funcionalidades possíveis.

Com a validação do protótipo não funcionais e suas regras agregadas, utilizando para tanto do Excel, foi gerado um protótipo funcional e navegável com a proposta do instrumento computacional para gerenciamento dos riscos do processo de reengenharia. Esse segundo protótipo foi apresentado novamente para as partes interessadas e continha cincos telas sendo elas: consulta de controles, consulta de riscos, consulta de entrevistas, registro de análise de riscos e consulta de indicadores de grau dos possíveis riscos identificados.

As partes interessadas avaliaram o protótipo por meio de análise crítica e comentários sobre a dinâmica do instrumento e das telas apresentadas, sugerindo o refinamento de alguns requisitos. Com isso, geramos a versão final do instrumento computacional contemplando os ajustes, sendo composto por nove telas: Home, consulta e cadastro de controles, consulta e cadastro de riscos, consulta e cadastro de entrevistas, avaliação dos riscos e mapa de riscos.

A CredTec validou o uso da ferramenta de maneira prática, aplicando o processo AS-IS de reengenharia de software do projeto de benefícios da previdência social. Para tanto

foram selecionadas pela equipe técnica as atividades do processo AS-IS de reengenharia de software mais críticas que contemplavam a etapa do pré-projeto. os riscos previamente identificados e os possíveis controles aplicados a eles foram associados às atividades do processo.

Para obter uma amostragem dos riscos, os usuários responderam uma pesquisa sobre a aplicação dos controles associados às atividades selecionadas do processo de reengenharia de software.

Com isso, o instrumento computacional apresentou para os usuários todos os riscos associados ao processo de reengenharia de software de maneira que fosse possível realizar a análise e avaliação desses riscos.

Uma vez a análise e a validação dos riscos concluídas, o instrumento computacional permitiu a exibição de um mapa de riscos contendo a nível de cada um dos riscos avaliados.

O questionamento dessa pesquisa foi se o gerenciamento dos riscos no processo de reengenharia de software pode auxiliar os gestores dos projeto na identificação, análise e avaliação dos riscos no processo atual; a aplicação de controles relacionados a reengenharia de software é possível mitigar os riscos identificados e se é possível um processo de reengenharia melhorado já contemplando os controles aplicados a reengenharia de software?

Para respostas a essas questões foi utilizada a metodologia de natureza aplicada, abordagem qualitativa e exploratória, através de um estudo de caso in-loco.

O problema central dessa pesquisa foi de que os sistemas legados que estavam sendo migrados no processo de reengenharia de software pela CredTec, que demandavam mais tempo do que o planejado para sua execução, também apresentavam problemas como baixa qualidade em ambiente produtivo e erros desconhecidos.

Como resposta para esses problemas, os resultados obtidos a partir da aplicação de um instrumento computacional para gerenciamento de riscos no processo de reengenharia de software da CredTec, indicam que é possível gerenciar riscos no processo e garantir uma redução das incertezas. Todavia, destaca-se a importância de realizar uma checagem periódica de possíveis deficiências no processo, como previsto nas atividades de monitoramento e análise crítica do processo de gestão de riscos.

Percebe-se então que a sistematização do gerenciamento de riscos pôde contribuir para melhorar o nível de maturidade da CredTec no que se refere ao processo de reengenharia de software.

Um dos limitadores encontrados no decorrer do desenvolvimento dessa pesquisa é que a análise e avaliação dos riscos foi realizada dentro do fluxo de trabalho atual, sendo necessário unir as atividades exercidas com as questões do estudo. Outro limitador encontrado é que a pesquisa foi realizada dentro de um cenário pandêmico, influenciando então na dinâmica de trabalho, o que ameaçou sua conclusão. Também possível entender como limitador que o instrumento computacional desenvolvido não permite a automação da identificação dos riscos e geração das entrevistas.

Dessa forma recomendamos que, por se tratar de um protótipo, o instrumento computacional para gerenciamento de riscos no processo de reengenharia de software da CredTec deve ser constantemente reavaliado e evoluído para sua melhor adequação aos possíveis cenários da instituição, incluindo outras funcionalidades, como a automatização de identificação dos riscos e geração das entrevistas.

7.1 Sugestões de trabalhos futuros

Como proposta de melhoria dessa pesquisa sugere-se a aplicação da ferramenta proposta em outros projetos de reengenharia de software e a realização de uma pesquisa bibliográfica mais aprimorada para criar uma base de conhecimento mais sólida.

Além disso, sugere-se a avaliação de outras ferramentas de mercado para adequação da proposta em Excel para uma ferramenta robusta com tecnologias de mercado reconhecidas de forma a expandir o uso da ferramenta e aprimoração de sua base de dados.

Acredita-se que seja necessário aprimorar a proposta de ferramenta para que seja possível cadastrar os riscos e enviar as pesquisas diretamente da plataforma computacional. Será necessária a avaliação da ferramenta por outros usuários com perfis distintos.

Á partir dessa avaliação, será possível coletar sugestões de melhoria na estrutura da ferramenta para elaboração de um segundo MVP.

Sugere-se uma análise bibliográfica para identificar outros métodos possíveis de identificação, análise e avaliação dos riscos que podem ser utilizados com a mesma finalidade. Em posse dessa informação, será realizada a criação de uma ferramenta computacional identificação, análise e avaliação dos riscos no processo de reengenharia de software com maior qualidade.

Com uma base dados consolidada com os controles aplicados à vulnerabilidade em processo de reengenharia de software, como também a ferramenta para análise e

avaliação dos riscos na mudança de tecnologia dos aplicativos legados, será realizada a aplicação do instrumento desenvolvimento, documentado todos os resultados alcançados e como forma de resposta aos riscos, o TO-BE será atualizado com todos os riscos identificados, analisados e avaliados.

Referências

[1] R. F. a. o. Manuel, "Mudanças tecnológicas e o impacto nas organizações: o processo de comunicação no estudo de caso da empresa angolana 'Teleservice, SA'," 2017.

[2] I. Sommerville, Software engineering 9th edition, ISBN-10, 137035152, 2011.

[3] E. d. V. AGILAR, Uma abordagem orientada a serviços para a modernização de sistemas legados, 2016.

[4] R. M. Pressman, Engenharia de Software-8ª Edição, McGraw Hill Brasil, 2016.

[5] International Standardization Organization, "ISO/IEC 25010:2011 - Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models," Standardization International Organization - ISO, 2011.

[6] R. M. Pressman, Engenharia de software: Uma abordagem profissional. 7ª Edição, 2011.

[7] R. M. Pressman, Software engineering: a practitioner's approach, Palgrave macmillan, 2005.

[8] S. C. N. d. Dôres, Um modelo para estimativa de esforço em projetos de reengenharia de software, Rio Grande do Sul: Pontifícia Universidade Católica do Rio Grande do Sul, 2015.

[9] Associação Brasileira de Normas Técnicas, "ABNT ISO GUIA: Gestão de riscos – princípios e diretrizes," ABNT, Rio de Janeiro, 2009.

[10] Z. C. V. W. R. &. W. G. Alreemy, "Critical success factors (CSFs) for information technology governance (ITG)," International Journal of Information Management, pp. 36(6), 907-916, 2016.

[11] International Starndard Organization, "ISO/IEC 13335-1:2004 - Information technology – security techniques – management of information and communications technology security – part 1: Concepts and models for information and communications technology security management," ISO, 2004.

[12] Associação Brasileira de Normas Técnicas, "ABNT ISO GUIA: Gestão de riscos – técnicas para o processo de avaliação de riscos," ABNT, Rio de Janeiro, 2012.

[13] Tricentis, “Software fail watch 5th in review,” https://www.tricentis.com/resources/software-fail-watch-5th-edition, 2018.

[14] A. C. G. a. L. J. C. Trapp, “Avaliação e gerenciamento do risco operacional no Brasil: análise de caso de uma instituição financeira de grande porte,” Revista Contabilidade & Finanças, pp. 16(37), 24-36, 2005.

[15] P. a. R. B. Naur, “Software engineering Report of a conference sponsored,” NATO Science Committee, pp. 7th-11th, 1969.

[16] Institute of e Electronics Engineers Electrical, “IEEE software engineering standards collection,” IEEE, 1991.

[17] F. N. Rodrigues, Análise e Gestão de requsitos de Software: Onde nascem os sistemas, Saraiva Educação SA, 2018.

[18] D. A. Rezende, “Engenharia de software e sistemas de informação,” Brasport, p. 12, 2006.

[19] M. J. Z. d. Barros, “Engenharia de software,” p. 12, 2018.

[20] A. Schnell, “Uso do modelo cascata no desenvolvimento de um sistema para serviços de oficina mecânica automotiva,” Universidade Tecnológica Federal do Paraná, p. 12, 2018.

[21] R. D. Nunes, “A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming (xp): uma alternativa para pequenas empresas do setor de tecnologia da informação,” ForScience, p. 4(2), 2016.

[22] IBM International Business Machines Corporation, “RUP - Rational Unified Process®,” 2011.

[23] A. Anwar, “A review of rup (rational unified process),” International Journal of Software Engineering (IJSE), p. 5(2):12–19, 2014.

[24] Object Management Group OMG®, “Business Process Model and Notation (BPMN),” http://www.omg.org/spec/BPMN/2.0, 2010.

[25] A. e. R. M. Martínez, “Guía a rational unified process,” Escuela Politécnica Superior de Albacete–Universidad de Castilla la Mancha, 2014.

[26] P. Caroli, Direto ao ponto: criando produtos de forma enxuta, Editora Casa do Código, 2015.

[27] R. d. Oliveira, “História do Scrum,” 2020. [Online]. Available: https://blog.romarconsultoria.com.br/2015/09/a-historia-do-

scrum.html#:~:text=O%20Scrum%20foi%20criado%20por,Takeuchi%20e%20Nonaka%20em%201986..

[28] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland e D. Thomas, "Manifesto para Desenvolvimento Ágil de Software," 2001. [Online]. Available: https://agilemanifesto.org/iso/ptbr/manifesto.html.

[29] K. Beck, Extreme programming explained: embrace change, addison-wesley professional, 2000.

[30] State of Agile ™, "14 th Annual State of Agile™ Report," 2020. [Online]. Available: https://stateofagile.com/#ufh-i-615706098-14th-annual-state-of-agile-report/7027494.

[31] K. e. J. S. Schwaber, "The scrum guide," Scrum Alliance, 2011.

[32] K. e. M. B. Schwaber, Agile software development with Scrum (vol. 1), Upper Saddle River: Prentice Hall, 2002.

[33] H. e. I. N. Takeuchi, "The new new product development game," Harvard business review, p. 64(1):137–146, 1986.

[34] Scrum.org, "Scrum Framework," https://www.scrum.org/, 2020.

[35] L. Gonçalves, "Scrum," Controlling & Management Review, p. 62(4):40–42, 2018.

[36] M. I. C. e. al., "How to elicit and specify software requirements from bpmn diagrams?," Proceedings of the XIV Brazilian Symposium on Information Systems, pp. 1-8, 2018.

[37] H. J. M. e. O. G. Leopold, "Learning from quality issues of bpmn models from industry," IEEE software, p. 33(4):26–33, 2015.

[38] M. J. Lisboa, "A importância do gerenciamento de processos de negócios (bpm) na otimização e melhoria contínua de processos de ti," Governança de Tecnologia da Informação-Unisul Virtual, 2018.

[39] ABPMP International, "Guide to the Business Process Management Body of Knowledge (BPM CBOK®)," ABPMP, 2020.

[40] P. Forbrig, "Continuous software engineering with special emphasis on continuous business-process modeling and human-centered design,"

Proceedings of the 8th International Conference on Subject-oriented Business Process Management, pp. 1-4, 2016.

[41] E. M. M. e. Y. R. Alreshidi, "Cloud-based bim governance platform requirements and specifications: software engineering approach using bpmn and uml," Journal of Computing in Civil Engineering, p. 30(4):04015063, 2016.

[42] S. Oliveira, "A comparative analysis between bpmn and spem modeling standards in the software processes context," 2012.

[43] Project Management Institute, "Guia PMBoK® 6ª edição: Um guia do conhecimento em gerenciamento de projetos," PMI, 2018.

[44] E. C. d. Oliveira, "Proposta para análise de riscos no processo de planejamento da contratação de ti: um estudo exploratório para órgãos governamentais," iSys-Brazilian Journal of Information Systems, p. 9(1):168–186, 2016.

[45] G. E. Y. e. N. Z. Elahi, "A vulnerability-centric requirements engineering framework: analyzing security attacks, countermeasures, and requirements based on vulnerabilities," Requirements engineering, p. 15(1):41–62, 2010.

[46] Mistério de Ciência, Tecnologia, Inovação e Comunicação, "Inventário e mapeamento de ativos de informação," MCTIC, 2017.

[47] H. Y. Ching, "Contribuição das boas práticas do mercado para a eficiência na gestão de risco corporativo," REBRAE, p. 4(3):257–273, 2011.

[48] Tribunal de Contas da União, "Manual de gestão de riscos do TCU," https://portal.tcu.gov.br/lumis/portal/file/fileDownload.jsp?fileId=FF8080816364D79801641D7B3C7B355A, 2018.

[49] Ministério de Transparência e Controladoria Geração da União, "Metodologias de gestão de riscos," https://www.gov.br/cgu/pt-br/centrais-de-conteudo/publicacoes/institucionais/arquivos/cgu-metodologia-gestao-riscos-2018.pdf, 2018.

[50] G. Moraes, Sistema de Gestão de Riscos - ISO 31.000 Comentada e Ilustrada: ISO 31.000 - Princípios e Diretrizes (2º edição), GVC Editora, 2016.

[51] R. G. d. e. a. Araujo, "Utilização de metodologias de gestão de risco na implementação de um sistema integrado de gestão (erp): estudo de caso em uma empresa industrial de grande porte," Universidade Presbiteriana Mackenzie, 2013.

[52] C. M. d. S. Xavier, “Gerenciamento de Projetos: como definir e controlar o escopo do projeto,” Saraiva Educação SA,, 2009.

[53] V. R. Coelho, “Avaliação de riscos na elicitação de requisitos de software,” Universidade de Brasília, 2019.

[54] International Standard Organization, “ISO/IEC 27001: Information Security Management,” ISO, 2013.

[55] Presidência da República, “LGPD - Le Geral de Proteção aos Dados Pessoais nº 13.709,” http://www.planalto.gov.br/ccivil_ 03/_ato2015-2018/2018/lei/L13709.htm, 2018.

[56] S. S. e. al., “CMMI ® for development: guidelines for process integration and product improvement,” Pearson Education, 2011.

[57] G. H. T. e. al., “Mps.br: promovendo a adoção de boas práticas de engenharia de software pela indústria brasileira,” XIII Congreso Iberoamericano en" Software Engineering"(CIBSE), Cuenca, Equador, 2010.

[58] E. Ries, The lean startup: How today’s entrepreneurs use continuous innovation to create radically successful businesses, Currency, 2011.

[59] V. Fontanette, Uma abordagem para migração gradativa de aplicações legadas, São Carlos: Universidade Ferderal de São Carlos Departamento de Computação, 2005.

[60] B. R. D. e. al., Interpretive structural modelling of critical risk factors in software engineering project, Benchmarking: An International Journal, 2016.

[61] S. A. Ó. Pinto, Gerenciamento de projetos: análise dos fatores de risco que influenciam o sucesso de projetos de sistemas de informação, São Paulo, 2002.

[62] Z. M. J. e. al., “An industrial study on the risk of software changes,” Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, pp. 1-11, 2011.

[63] Y. S. e. al., “A software system for risk management of information systems,” 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), pp. 1-6, 2018.

[64] C. e. D. K. Y. Kumar, “A probabilistic software risk assessment and estimation model for software projects,” Procedia Computer Science, p. 54:353– 361, 2015.

[65] M. e. M. S. Sadiq, “A systematic approach for the estimation of software risk and cost using esrctool,” CSI transactions on ICT, p. 1(3):243–252, 2013.

[66] T. Kendrick, “Identifying and managing project risk: essential tools for failureproofing your project,” Amacom, 2015.

[67] C. C. e. E. C. d. F. Prodanov, Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico-2ª Edição, Editora Feevale, 2013.

[68] W. C. e. a. Rodrigues, Metodologia científica, Paracambi: Faetec/IST, 2007.

[69] T. E. a. S. D. T. Gerhardt, Métodos de Pesquisa, Plageder, 2009.

[70] G. C. R. F. da Silva, O método científico na psicologia: abordagem qualitativa e quantitativa, 2010.

[71] G. I. Engel, Pesquisação, SciELO Brasil, 2000.

[72] A. M. Ozaki, “Um estudo sobre a elaboração de roadmaps tecnológicos em empresas brasileiras de software.,” Universidade de São Paulo, São Paulo, 2014.

[73] F. L. A. d. Silva, “Análise do impacto do gerenciamento de riscos no sucesso de projetos: um estudo de caso em uma organização de desenvolvimento de software,” Universidade Federal de Pernambuco, 2017.

[74] R. V. d. M. e. al., “A utilização do scrum como recurso educacional no processo de aprendizagem em engenharia de software,” International Journal on Alive Engineering Education, p. 3(2):87–102, 2016.