



## **PROJETO DE GRADUAÇÃO**

# **AVALIAÇÃO DE RISCOS NA ELICITAÇÃO DE REQUISITOS DE SOFTWARE**

Por,

**Vinicius Ribeiro Coelho**

**Brasília, 14 de fevereiro de 2019**

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Departamento de Engenharia de Produção

## PROJETO DE GRADUAÇÃO

# AVALIAÇÃO DE RISCOS NA ELICITAÇÃO DE REQUISITOS DE *SOFTWARE*

POR,

**Vinícius Ribeiro Coelho**

Relatório submetido como requisito parcial  
para obtenção do grau de Engenheiro de Produção

### BANCA EXAMINADORA

Prof. Dr. Edgard Costa Oliveira, UnB/EPR (Orientador) \_\_\_\_\_

Prof. Dr. Ari Melo Mariano, PhD. \_\_\_\_\_

Brasília, 14 de fevereiro de 2019

“Tudo passa”

**Chico Xavier**

## **Dedicatória**

*Dedico esse trabalho primeiramente a Deus, pois sem ele nada seria. Dedico também aos meus pais João Gilberto e Fátima pelos conselhos, pela paciência e principalmente por todo amor que sempre tive. A minha vó que acendeu umas velas por mim. A meu irmão que, com seus métodos de tortura, me ensinou a ser forte e aguentar as pauladas que a UnB me deu.*

# AGRADECIMENTOS

Agradeço a Deus por ter me dado esse corpo físico e me colocado nessa família, sem Ele nada seria. Ao meus pais que me educaram de maneira a nunca parar e nem desistir, que me deram todo o apoio em todas as horas. Agradeço a dona Fátima em especial que mesmo com todos os percalços de sua vida só nos deu amor. E ao seu Gilberto por ter me ensinado minha primeira profissão aos 12 anos. Agradeço ao meu irmão Coelho por ter me ensinado a ser resiliente e ter me mostrado minha futura corporação.

Agradeço ao meu exímio orientador Professor Edgard Costa por toda paciência, conselhos e orientações que me deu durante esse período turbulento da minha vida e a todos os professores da engenharia de produção, sendo os mais notáveis Prof. Sérgio, Profa. Simone e Profa. Martha. Agradeço ao meu amigo Wilker Mesquita que me salvou nessa turbulência. Agradeço também a minha namorada Ana Maria por ter me acalmado em momentos de crise, e ter me auxiliado veemente na realização deste trabalho. Muito obrigado mestre amigo, por ter me concedido as realizações dos meus maiores sonhos.

Vinícius Ribeiro Coelho

---

## RESUMO

Os programadores têm tendência a negligenciar a parte de avaliação de riscos na fase de elicitação de requisitos do processo de desenvolvimento do *software*. Com isso, as empresas tendem a gastar 60% a mais em tempo e dinheiro quando realizam uma má elicitação de requisitos. Esses problemas se agravam ainda mais em projetos pequenos realizados por empresas que estão entrando no mercado. O presente trabalho tem como objetivo desenvolver uma ferramenta de avaliação dos riscos envolvidos na elicitação de requisitos de um contexto de projeto de desenvolvimento de *software*. Os objetivos específicos são descrever o projeto de desenvolvimento de *software* e a ferramenta de *software* desenvolvida pela empresa em questão, pesquisar os riscos do processo de elicitação de requisitos de projetos de *software* de pequeno porte e identificar métodos correntes de avaliação de riscos do desenvolvimento de *software*. Por fim, objetiva-se construir e testar a ferramenta de avaliação de riscos na elicitação de requisitos no projeto de desenvolvimento de *software* da empresa estudada. A metodologia utilizada consistiu em reuniões técnicas com a equipe de projeto, levantamento dos métodos de análise de risco, definição dos critérios de análise de riscos, desenho do processo de elicitação de requisitos do cliente, desenho do processo de desenvolvimento de *software*, construção das tabelas e aplicação da ferramenta junto ao cliente. Como resultado elaborou-se a construção de uma ferramenta de avaliação de riscos da elicitação de requisitos, voltada para pequenas equipes de desenvolvimento de *software*, baseada numa lista de ameaças previamente criada. A ferramenta foi testada e o projeto da empresa voluntária foi documentado. Após o processo de avaliação de riscos da elicitação de requisitos com essa ferramenta, os objetivos do projeto foram alcançados e concluiu-se que a avaliação de risco por parte das pequenas empresas desenvolvedoras de *software* torna a produção desses produtos menos dispendiosas em tempo e retrabalho.

**Palavras-Chave:** Risco, Desenvolvimento de *software*, Análise de Risco, Avaliação de Risco, Elicitação de Risco.

---

## ABSTRACT

Programmers tend to neglect risk assessment in elicitation phase during software requirements of the software development process. Companies tend to spend 60% more on time and money when poorly they perform requirements. These problems worsen even more in small projects carried out by companies that are starting. The present work aims to develop a tool to evaluate the risks involved in the elicitation of requirements of a software development project context. The specific objectives are to describe the software development project and the software tool developed by the company, to investigate the risks of elicitation processes of small software project requirements and to identify current methods of risk assessment of software development, in order to construct and test the risk assessment tool in the requirement elicitation a software development project of the studied company. The methodology used consisted of technical meetings with the project team, survey of risk analysis methods, definition of risk analysis criteria, design of the client's requirements elicitation process, design of the software development process, construction of the tables and application of the tool with the client. As a result, it was developed a requirements elicitation risk assessment tool for small software development teams, based on a previously created threats list. The tool was tested, and the project of the voluntary company was documented. After the requirements elicitation risk assessment process with this tool, the present study objectives were achieved, and it was concluded that doing risk assessment by small software developers makes the production of these products less costly in time and rework.

**Key Words:** Risk assessment, Software Development, Risk Analysis, Risk Management, Requirement Elicitation.

# SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Objetivos .....	14
1.2	Metodologia.....	14
2	REVISÃO DE LITERATURA .....	17
2.1	Risco e incerteza.....	17
2.2	Engenharia de requisito .....	19
2.3	Definição de requisitos de <i>software</i> .....	20
2.4	Elicitação de requisitos.....	21
2.5	Avaliação de risco .....	22
2.6	Metodologia ágil.....	23
3	ANÁLISES DAS TÉCNICAS DE AVALIAÇÃO DE RISCOS .....	25
3.1	Técnicas de avaliação de riscos utilizadas no mercado.....	25
3.2	Técnicas de avaliação de riscos do processo de desenvolvimento de <i>software</i> .....	31
4	ESTABELECIMENTO DO CONTEXTO DA PESQUISA.....	36
4.1	Contexto Interno.....	36
4.2	Contexto Externo.....	36
4.3	Desenho do processo de desenvolvimento do <i>software</i> utilizado no contexto.....	37
4.4	Descrição das atividades de elicitação de requisitos .....	38
4.5	Descrição das <i>features</i> do <i>software</i> .....	40
4.6	Funcionamento do <i>software</i> .....	40
5	DEFINIÇÃO DOS CRITÉRIOS DE AVALIAÇÃO DOS RISCOS .....	46
5.1	Compilação da lista de riscos na elicitação de requisitos.....	46
5.2	Critério de análise de risco .....	47
5.3	Seleção dos métodos de avaliação de riscos.....	51
5.4	Resultado da primeira análise e avaliação dos riscos .....	53
6	DESENVOLVIMENTO DA FERRAMENTA DE AVALIAÇÃO DE RISCOS .....	55
6.1	Tela de análise da ferramenta .....	55
6.2	Tela da curva ABC .....	57
6.3	Tela do quadro de avaliação .....	59
7	CONSIDERAÇÕES FINAIS .....	70
8	REFERÊNCIAS .....	72
9	ANEXO .....	74



# LISTA DE FIGURAS

1. Figura 1. <i>Breakdown of Topics for the Software Requirements KA SWEBOK</i> .....	21
2. Figura 2. Contribuição do processo de avaliação de riscos para o processo de gestão de riscos (ABNT/ISO 31010, 2012) .....	26
3. Figura 1 Processo de desenvolvimento de software do contexto estudado .....	39
4. Figura 4. Funções da ferramenta .....	56
5. Figura 5. Tela inicial da ferramenta Risky_TOOL_2000_V1 .....	57
6. Figura 6. Cálculo ABC .....	58

# LISTA DE TABELAS

1.	Tabela 1. Primeira avaliação método 1 .....	49
2.	Tabela 2. Avaliação e impacto .....	50
3.	Tabela 3. Primeira avaliação método 2 .....	50
4.	Tabela 4. <i>Ranking</i> de riscos .....	52
5.	Tabela 5. Quadro avaliação primeira análise .....	53

# LISTA DE GRÁFICOS

1. Gráfico 1. Gráfico ABC primeira avaliação .....	54
2. Gráfico 2. Curva ABC da ferramenta .....	59
3. Gráfico 3. Curva ABC da ferramenta colorida .....	59
4. Gráfico 4. Curva ABC segunda análise .....	68
5. Gráfico 5. Curva ABC sugestão .....	68

# LISTA DE QUADROS

1.	Quadro 1. Quadro de tipo de riscos .....	18
2.	Quadro 2. Conceitualizações de incerteza (LIPSHITZ & STRAUSS, 1997) adaptado .....	19
3.	Quadro 3. Formulário “o que aconteceria se” .....	29
4.	Quadro 4. Critério de Análise para Severidade das Falhas Adaptado da <i>FORD MOTOR COMPANY</i> (1988) .....	30
5.	Quadro 5. Critério de Análise para Índice de Ocorrência das Falhas Adaptado de <i>FORD MOTOR COMPANY</i> (1988) .....	31
6.	Quadro 6. Critério de Análise para Índice de Detecção das Falhas Adaptado da <i>FORD MOTOR COMPANY</i> (1988) .....	32
7.	Quadro 7. Probabilidade .....	34
8.	Quadro 8. Impacto do risco .....	34
9.	Quadro 9. probabilidade VS impacto retirado do PMI (2013) .....	36
10.	Quadro 10. Avaliação probabilidade vs impacto retirado do PMI (2013) .....	36
11.	Quadro 11. Lista final de riscos .....	49
12.	Quadro 12. Comparação de posição de riscos .....	51
13.	Quadro 13. Comparação de produção entre análises .....	66

# 1 INTRODUÇÃO

Neste capítulo são apresentadas as considerações preliminares sobre o tema do projeto. A contextualização, os objetivos e a estrutura do trabalho.

Atualmente, os *softwares* estão inseridos em todas as nossas atividades cotidianas como assistir televisão, ir à geladeira e principalmente usar o celular. Por conseguinte, existe uma grande demanda de desenvolvimento de *software* em todas as regiões do globo. Segundo pesquisa realizada no meio eletrônico *ZDNET* (2016), mais de 68% dos projetos de desenvolvimentos tecnológicos tendem a falir. Ainda segundo o site, outro dado importante é que aproximadamente 41% dos recursos totais destinados ao desenvolvimento de *software* serão consumidos devido a elicitação de requisitos malfeita. Ademais, as empresas tendem a gastar 60% a mais em tempo e dinheiro quando realizam uma má elicitação de requisitos.

A empresa em que foi realizado o contexto de desenvolvimento é um negócio de pequeno porte que realiza trabalhos sob demanda de clientes. Os programadores usam da metodologia ágil para o desenvolvimento do projeto, de forma que o enfoque da ferramenta se deu na avaliação de riscos na elicitação de requisitos que utilizam de desenvolvimento ágil no desenvolvimento de *software*. Os produtos desenvolvidos pela empresa estudada são de pequeno porte e voltados para ambiente *web*.

Gilb (1988) afirma que se você não ataca os riscos de maneira ativa, eles irão te atacar de maneira ativa, o que demonstra que os riscos devem ser analisados e tratados para não impactar no projeto. Para Boehm (1998), existem quatro motivos principais para a implementação de um plano de gestão de ameaças, sendo elas:

- a) Evitar desastres de projetos de *software*, incluindo orçamentos e cronogramas, *softwares* defeituosos produtos e falhas operacionais;
- b) Evitar retrabalho causado por requisitos errados, ausentes ou ambíguos, design ou código, que normalmente consome 40-50% do custo total de desenvolvimento de *software*;
- c) Evitar excessos com técnicas de detecção e prevenção em áreas de risco mínimo ou nenhum risco;
- d) Estimular uma solução de *software* ganha-ganha, onde o cliente recebe o produto de que precisa e o fornecedor faz os lucros que eles esperam;

Nesse projeto de graduação foram tratados os problemas relacionados a riscos na elicitação de requisitos em empresas pequenas com projetos pequenos que usam metodologia ágil. A complexidade dos projetos de *software* faz com que grande parte desses projetos exceda o prazo e o orçamento previstos e não atenda às expectativas dos clientes em termos de funcionalidades e qualidade (ROCHA e BELCHIOR, 2004).

A elicitação de requisitos é uma das etapas do desenvolvimento do *software* que requer maior cuidado da equipe de projeto, porque é a fase em que o cliente conversa com o desenvolvedor para criar o seu projeto. Por se tratar de uma fase inicial, em que se busca levantar as necessidades do cliente, erros realizados nesse momento serão propagados ao decorrer do projeto, aumentando o custo de retrabalho.

Esses problemas se agravam ainda mais em projetos pequenos realizados por empresas que estão no seu começo, que não possuem muito pessoal e até mesmo não possuem uma métrica de elicitação de requisitos, que é o caso da empresa estudada nesse projeto. Assim, existem riscos nos projetos, gerando maiores chances de ocorrerem eventos que os atrasem, ou até mesmo que os impeçam de continuar. Foi evidenciado no contexto de desenvolvimento a ocorrência de vários dos riscos listados na elicitação de requisitos e a necessidade de uma avaliação dos riscos sistematizada.

Existe uma grande correlação entre o sucesso do projeto e um bom levantamento dos requisitos. Por isso, as avaliações de riscos devem ser realizadas e tratadas de forma eficaz. Destarte, o projeto abordará somente os riscos envolvidos na fase de elicitação, não sendo seu escopo os outros riscos envolvidos.

## **1.1 Objetivos**

### **1.1.1 Objetivo geral**

Desenvolver uma ferramenta de avaliação dos riscos envolvidos na elicitação de requisitos de um contexto de projeto de desenvolvimento de *software*.

### **1.1.2 Objetivos específicos**

- a) Descrever o projeto de desenvolvimento de *software* e a ferramenta de *software* desenvolvida pela empresa;
- b) Pesquisar os riscos do processo de elicitação de requisitos de projetos de *software* de pequeno porte e identificar métodos correntes de avaliação de riscos do desenvolvimento de *software*;
- c) Construir a ferramenta avaliação de riscos na elicitação de requisitos no projeto de desenvolvimento de *software* da empresa estudada.
- d) Testar a ferramenta avaliação de riscos na elicitação de requisitos no projeto de desenvolvimento de *software* da empresa estudada.

## **2 METODOLOGIA**

Para alcançar o primeiro objetivo de descrever o projeto de desenvolvimento de *software* e a ferramenta de *software* desenvolvida pela empresa, foi elaborada a descrição do contexto de desenvolvimento, atingida a partir de reuniões diárias realizadas no ambiente de trabalho da empresa, no período de um mês. As pautas das reuniões eram definidas no dia anterior. Durante os encontros eram realizadas anotações para a futura descrição. O desenho do processo de desenvolvimento de

*software* foi feito em conjunto com a equipe de projeto e foi criado a partir de um arquivo em branco, elaborando as etapas de entrada, preparação, execução, validação e liberação e todos seus detalhamentos conforme indicação da equipe de projeto. Foi utilizada a ferramenta da Adobe para edição do processo.

Ademais, para a descrever o produto desenvolvido pela equipe, ocorreram encontros também diários, realizados após a descrição do processo de desenvolvimento de *software*. Nessas reuniões, um dos desenvolvedores relatava oralmente as *features* da ferramenta e desse modo eram realizadas anotações, no Microsoft Word, para a descrição do *software*. Após essa etapa, o outro método realizado para a descrição das funcionalidades da ferramenta era a observação das partes já construídas pela equipe de projeto, assim eram descritas conforme a visualização de cada parte do *software*.

O segundo objetivo era pesquisar os riscos do processo de elicitação de requisitos de projetos de *software* de pequeno porte e identificar métodos correntes de avaliação de riscos do desenvolvimento de *software*. Para atingir a primeira parte de pesquisa dos riscos foram buscados na plataforma de pesquisa do Scholar Google as palavras chaves: avaliação de risco na elicitação de requisitos, *risk assessment in requirement engineer*, gerenciamento de riscos na elicitação de requisitos de *software*, *risk management in requirement engineer* e análise de risco na elicitação de requisitos do processo de desenvolvimento de *software*. Foram selecionados 44 artigos que pareciam se assimilar melhor ao objetivo deste projeto. A partir dessa lista, ocorreu a leitura dos artigos e identificação e seleção dos riscos relacionados a elicitação de riscos citados pelos autores dos artigos. Porém, após a leitura, só foram utilizados 9 desses artigos. Assim, foi construída uma lista bruta com diversos riscos relacionados a elicitação de requisitos. O próximo passo foi a filtração dos riscos citados na primeira lista, agrupando riscos similares e eliminando os iguais, para então elaborar a lista final de riscos de elicitação de requisitos.

Além do mais, para identificar os métodos correntes de avaliação de riscos do desenvolvimento de *software* foram utilizadas pesquisas semelhantes a anterior. Foram procurados métodos de avaliação de riscos nos livros de engenharia de *software* dos autores Pressman e Sommerville e também nas ABNT/ISO 31000 (2009) e ABNT/ISO 31010 (2012). Para complementar a identificação de métodos as seguintes palavras foram utilizadas na plataforma do Google: métodos de avaliação de risco na elicitação de requisitos, métodos de avaliação de risco, métodos de gerenciamento de risco no desenvolvimento de *software*, *methods of risk assessment* e *methods of risk management*.

Para lograr o último objetivo de construir e testar a ferramenta de avaliação de riscos na elicitação de requisitos no projeto de desenvolvimento de *software* da empresa estudada. O desenvolvimento da ferramenta utilizou de programação VBA no Microsoft Excel, automatizando os resultados das planilhas. Para a construção da ferramenta foi desenvolvida uma planilha inicial contendo a classificação do risco, o próprio risco, a ocorrência, a probabilidade, o impacto e a multiplicação da probabilidade pelo impacto. O método de análise utilizado é a multiplicação da probabilidade pelo impacto, esses variando de 1 a 5. Em seguida, a ferramenta classifica os riscos do maior valor para a menor.

A próxima planilha pega os resultados de cada risco inseridos na tela inicial e os dispõem individualmente em um quadro de avaliação. Esse quadro está disposto de maneira que as colunas representam a probabilidade e as linhas, o impacto, variando de muito baixo a muito alto, e cada área do quadro representa uma sugestão de avaliação do risco. A seguir, os dados da ferramenta são retirados da planilha inicial e apresentam uma curva ABC com a divisão de cores por tipo de risco - sendo verde para os riscos A, amarelo para os riscos B e laranja para os riscos tipo C. Além disso, o teste da ferramenta se deu com os desenvolvedores utilizando-a em computadores próprios e analisando os riscos valendo-se da ferramenta.



## 3 REVISÃO DE LITERATURA

### 3.1 Risco e incerteza

Este capítulo tem como objetivo trazer os conceitos necessários a compreensão do leitor sobre o projeto. O conceito de risco está delimitado em vários documentos, desse modo os conceitos mais relevantes serão citados de forma a elucidar o caminho para compreensão íntegra dos riscos na obtenção da elicitação de requisitos de *software*.

Risco e incerteza são conceitos relacionados, mas distintos. Incerteza resulta da falta de informação. O risco é caracterizado pela probabilidade de um evento que resultará em um impacto negativo. O risco é frequentemente o resultado da incerteza. O inverso do risco é a oportunidade, que é caracterizado pela probabilidade de que um evento com um resultado positivo possa ocorrer. SWEBOK (2014, pg 31).

Pressman (2016) traz que o conceito de risco no *software* possui duas características: a incerteza e a perda. Sendo que essa nunca terá uma probabilidade de 100%, pois isso a tornaria uma restrição no desenvolvimento do *software*. Já aquela são as consequências malquistas da ocorrência do risco. O autor divide primordialmente risco em três grandes categorias, sendo elas: riscos de projeto, riscos técnicos e riscos de negócio.

A primeira das três partes revela as ameaças ao desenvolvimento do projeto. Está relacionado intrinsecamente com a parte de orçamento, cronograma, desenvolvedores e principalmente requisitos. Os riscos técnicos englobam a qualidade do desenvolvimento do *software*, “Riscos técnicos ocorrem porque o problema é mais difícil de resolver do que se pensava” (PRESSMAN 2016 pg 649). O último tipo de risco trata-se de perigos na viabilidade do desenvolvimento do *software*, esse geralmente associado a estratégia de negócio adotado pelos desenvolvedores.

“Você pode pensar em um risco como algo que você preferiria que não acontecesse. Os riscos podem ameaçar o projeto, o *software* que está sendo desenvolvido ou a organização” (SOMMERVILLE 2011 pg 415). Ademais, o autor divide riscos em três partes. O autor acredita que existem riscos de projeto, riscos de produto e risco de negócio. O quadro 01 apresenta essas definições segundo o autor. Esses riscos não são totalmente independentes, podendo um sobrepor ao outro, sendo capaz de um risco de projeto tornar-se um risco de negócio.

Quadro 1. Quadro de tipo de riscos adaptado Sommerville

Tipo de risco	Definição
Riscos de projeto	Riscos que afetam o cronograma ou os recursos de projeto. Um exemplo de um risco de projeto é a perda de um projetista experiente.
Riscos de produto	Riscos que afetam a qualidade ou o desempenho do <i>software</i> que está sendo desenvolvido. Um exemplo de um risco de produto é a falha de um componente comprado para o desempenho esperado.
Risco de negócio	Os riscos que afetam a organização que desenvolve ou adquire o <i>software</i> . Por exemplo, um concorrente que introduz um novo produto é um risco empresarial.

Utilizando-se de metodologias como dados pregressos e conhecimento técnico de especialistas, é viável constatar variáveis de riscos e mensurar a gravidade e ocorrência em certos casos. Quando se tem incerteza, não é sabido quais são os eventos que podem ir contra os objetivos traçados. A diferença entre risco e incerteza é definida, primordialmente, pelo grau de conhecimento que se tem sobre o fato.

Lipshitz & Strauss (1997) trazem em seu livro várias formas de como é abordado o risco, tanto em sua definição quanto como nomeação. O quadro 2 abaixo reproduz algumas dessas definições:

Quadro 2. Conceitualizações de incerteza (LIPSHITZ & STRAUSS, 1997) adaptado

<b>Autores</b>	<b>Termo</b>	<b>Conceitualização</b>
1. Anderson et al. (1981)	Incerteza	Uma situação na qual não se tem o conhecimento a respeito de vários estados da natureza que ocorreram ou que vão ocorrer.
2. Anderson et al. (1981)	Incerteza	Uma situação em que se sabe apenas a probabilidade de ocorrência dos vários estados da natureza que ocorreram ou que irão ocorrer.
3. Humphreys & Berkeley (1985)	Risco	A falta de habilidade para afirmar com certeza um dos itens seguintes: (a)seqüências de atos-eventos; (b) seqüências evento-evento; (c)valor das conseqüências; (d)processo de decisão apropriado; (e)preferências e ações futuras; (f)habilidade atual de afetar eventos futuros.
4. Anderson et al. (1981)	Risco	Mesmo de (1)
5. Anderson et al. (1981)	Risco	Mesmo de (2)
6. MacCrimmon & Wehrung (1986)	Risco	Exposição à chance de perda em uma situação de escolha.
7. Arrow (1965)	Risco	Uma função positiva da variância da distribuição de probabilidade dos resultados positivos e negativos esperados.
8. Hogarth (1987)	Ambiguidade	Falta de conhecimento precABNT/ISO a respeito da probabilidade de eventos (probabilidade de segunda ordem).

Em conclusão, existem diversas formas de se definir risco, mas todas encaminham-se para o mesmo destino: riscos são ocorrências negativas que afetam os projetos e se tem informação a respeito, já incertezas são ocorrências sem informações, que impactam negativamente o andamento do projeto.

### **3.2 Engenharia de requisito**

Este capítulo tem como objetivo explicar sobre engenharia de requisito. O processo de engenharia de requisitos dá-se após reunião com os clientes, onde discute-se todas as necessidades. Dessa maneira são estabelecidos os requisitos do projeto. Dependendo da aplicação no projeto, esses requisitos podem ser funcionais ou não funcionais. Com isso a posto, far-se-á uma lista com as prioridades após detalhados estudos dos requisitos. Isto deve ser documentado, o que usualmente é a parte mais complexa para os desenvolvedores. Não obstante, uma parte extremamente importante para o funcionamento do *software*.

Para Sommerville (2001) esta área é responsável por obter descrições do que o sistema deve fazer, os serviços que oferecem e as restrições no seu funcionamento. Esses requisitos devem refletir as

necessidades dos clientes para o projeto requerido, controlar um dispositivo, colocar um pedido ou encontrar informações.

“ área de conhecimento de Requisitos de *Software* (KA) é responsável pela elicitação, análise, especificação e validação de requisitos de *software*, bem como o gerenciamento de requisitos durante todo o ciclo de vida do produto de *software*. É amplamente reconhecido entre os pesquisadores e profissionais da indústria que os projetos de *software* ficam particularmente vulneráveis quando as atividades relacionadas aos requisitos são mal executadas. Os requisitos de *software* expressam as necessidades e restrições impostas a um produto de *software* que contribuem para a solução de algum problema real. SWEBOK, (2014, pg 32)

[...] a identificação dos objetivos a serem atingidos pelo futuro sistema, a operacionalização 3 de tais objetivos em serviços e restrições, e a atribuição de responsabilidades pelos requisitos resultantes a agentes humanos, dispositivos e *software*. LAMSWEERDE (2000, pg 5).

Em suma, os objetivos da engenharia de requisitos são, não só detectar as reais necessidades do cliente, mas também descobrir as possíveis restrições do desenvolvimento do *software*. Reafirmando, essa é a parte do desenvolvimento do *software* responsável pela comunicação entre o cliente e a equipe de projeto.

### **3.3 Definição de requisitos de *software***

O objetivo principal desta parte é esclarecer a definição de requisitos de *software*. Requisitos tem como função a determinação das necessidades dos *stakeholders*. Podemos estruturar de forma simplória requisitos em partes: requisito para atingir um objetivo ou para solucionar uma adversidade para o usuário, sendo isso uma condição ou capacidade do projeto. Tem-se também atendimento a documentos formalmente impostos e contrato entre tantos documentos, sendo isso também capacidade ou condição do projeto, de forma que podemos decompor as atividades de requisitos em avaliação, especificação, análise e elicitação.

Em sua forma mais básica, um requisito de *software* é uma propriedade do sistema, que deve ser exibida por algo para resolver algum problema no mundo real. Pode ter como objetivo automatizar parte de uma tarefa para alguém apoiar os processos de negócios de uma organização, corrigir falhas do *software* existente ou controlar um dispositivo. As formas pelas quais usuários, processos de negócios e dispositivos funcionam normalmente são complexas. Portanto, os requisitos sobre um determinado *software* normalmente são uma combinação complexa de várias pessoas em diferentes níveis de uma organização e que, de alguma forma, estão envolvidas ou conectadas com esse recurso do ambiente no qual o *software* operará. SWEBOK (2014, pg 33)

Beatty (2013) traz em seu livro “*software requirements*” um conceito muito interessante sobre requisitos. Eles tratam requisitos como sendo especificações a serem implementadas. Quer dizer, são descrições de como o programa deve funcionar, ou uma propriedade do sistema, e podem ser considerados também como atributos. Ainda são possíveis limitações para o desenvolvimento do

sistema. Outro tema abordado pelos autores sobre requisitos, foi que eles possuem dimensão no tempo, podendo ser de curto, médio ou longo período cada um com suas prioridades.

A figura 1 abaixo detalha como os requisitos de *software* são divididos. É perceptível a complexidade, demandando tempo e recursos, para obter essa fase do desenvolvimento de programas.

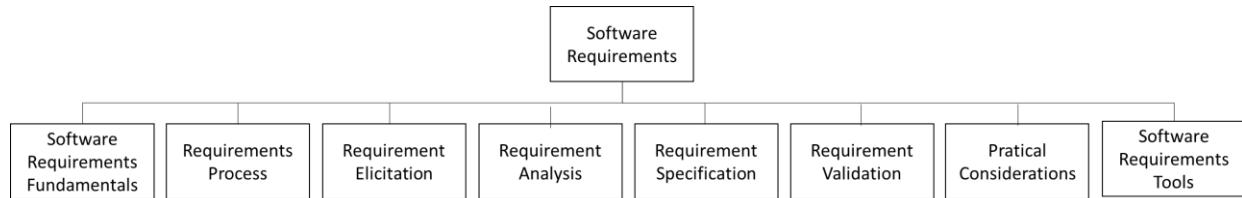


Figura 1. *Breakdown of Topics for the Software Requirements* adaptado do *SWEBOK*

É possível concluir que o conceito de requisitos abrange uma grande variedade de aspectos no desenvolvimento de *software*. Além do mais, que os requisitos são os conceitos iniciais fundamentais ao desenvolvimento do projeto. Desse modo, sem a elucidação do conceito de requisitos torna-se árduo entender o desenvolvimento deste projeto.

### 3.4 Elicitação de requisitos

Esta etapa demonstra os conceitos necessários a compreensão da elicitação de requisitos. O início do desenvolvimento de *software* é um momento de muitos percalços e com poucas informações. A elicitação de requisitos é a primeira parte na construção de um entendimento do problema que o *software* precisa solucionar. Nesse momento busca-se constatar os requisitos necessários para o sistema.

No primeiro instante os desenvolvedores unem-se aos usuários para em conjunto delimitar o problema a ser abordado, com foco principal nas funcionalidades que os sistemas devem dispor. Contudo, os usuários em sua maioria não possuem total entendimento do que desejam e como desejam. Por conseguinte, isso pode fazer com que alguns requisitos que foram definidos não solucionem as necessidades dos usuários em sua totalidade.

Kotonya (1998) expõe a elicitação de requisitos por ser um dos estágios com maior relevância, que envolve mais do que o diálogo sobre as necessidades/desejos entre usuário e desenvolvedor. Ocorre uma análise cautelosa da organização, de domínio da aplicação e dos processos de negócio onde o *software* será implementado. Além do mais, esse autor dividiu a elicitação de requisitos em partes, sendo o primeiro entendimento do domínio da aplicação, do problema, do negócio e, por fim, das necessidades e restrições das pessoas envolvidas no sistema. O primeiro é no sentido de se ter uma percepção holística de onde será inserido o sistema. Logo após, tem-se o intuito de conhecer os detalhes próprios do problema. A terceira parte é do entendimento do modo como o *software* inserido irá relacionar-se com a organização, de quais formas irá afetar os negócios.

Para Sommerville (2011) o processo de elicitação de requisitos se dá após o estudo de viabilidade inicial. É nesse momento que a equipe de projeto trabalha junto aos clientes e usuários finais do *software* para descobrir suas funções, os serviços que o sistema deve prestar, o desempenho exigido do *software*, as restrições de *hardware* e assim por diante. Diz ainda, que a elicitação e as análises de requisitos podem envolver diferentes tipos de pessoas dentro das organizações. Qualquer um que teve influência direta ou indireta no requisito do sistema é tratado como *stakeholder* para o autor.

Sommerville (2011) divide a elicitação de requisitos em quatro partes:

- a) Descoberta de requisitos: Este é o processo de interação com os *stakeholders* para descobrir seus requisitos. Os requisitos de domínio das partes interessadas e da documentação também são descobertos durante essa atividade. Existem várias técnicas complementares que podem ser usadas para a descoberta de requisitos.
- b) Classificação e organização de requisitos: Esta atividade coleta a coleção não estruturada de requisitos, agrupa os requisitos relacionados e os organiza em *clusters* coerentes. A maneira mais comum de agrupar requisitos é usar um modelo da arquitetura do sistema para identificar subsistemas e associar os requisitos a cada subsistema. Na prática, engenharia de requisitos e projeto arquitetônico não podem ser atividades completamente separadas.
- c) Priorização e negociação de requisitos: Inevitavelmente, quando várias partes interessadas estão envolvidas, os requisitos entrarão em conflito. Esta atividade está relacionada à priorização de requisitos e à descoberta e resolução de conflitos de requisitos por meio de negociação. Normalmente, as partes interessadas precisam se reunir para resolver as diferenças e concordar com os requisitos de compromisso.
- d) Especificação de requisitos: Os requisitos são registrados em documentos, podendo ser esses documentos formais ou informais.

Entretanto, o Swebok (2014) afirma que deve ocorrer uma comunicação contínua, junto ao cliente, durante todo o processo de desenvolvimento de *software*. Além disso, afirma que há a possibilidade de criação de um canal para essa comunicação. A elaboração do escopo é uma parte crítica da elicitação, uma vez que isso envolve fornecer descrições detalhadas do *software*, e sua finalidade é priorizar os produtos que serão entregues. Todos esses processos são para garantir que o cliente tenha suas necessidades mais importantes atendidas primeiro, minimizando assim o risco de os programadores desperdiçarem tempo com requisitos de baixa importância.

### **3.5 Avaliação de risco**

Segundo a ABNT/ISO 31000 (2009), gestão de riscos, a probabilidade de ocorrência de uma consequência do risco, seja ela positiva ou negativa, deve ter suas causas apreciadas. Isso envolve o processo de análise de risco. Deve ser realizada uma descoberta dos fatores que afetam a probabilidade e consequência. “Um evento pode ter várias consequências e pode afetar vários objetivos.” (ABNT/ISO 31.000 2009)

A análise de riscos envolve desenvolver a compreensão dos riscos. A análise de riscos fornece uma entrada para a avaliação de riscos e para as decisões sobre a necessidade de os riscos serem tratados, e sobre as estratégias e métodos mais adequados de tratamento de riscos. A análise de riscos também pode fornecer uma entrada para a tomada de decisões em que escolhas precisam ser feitas e as opções envolvem diferentes tipos e níveis de risco. ABNT/ISO 31.0000 (2009, pg 26)

A ABNT/ABNT/ISO 31010 (2012), técnicas para processo de avaliação de risco, trata sobre a avaliação de riscos e como esse processo de avaliação permite ao gerente o entendimento do risco em vários aspectos, tais como causa, consequência e probabilidade. Esse entendimento das ameaças permite ao responsável pelo tratamento do risco tomar decisões sobre numerosos aspectos tais como se é conveniente que a atividade seja feita, se o risco precisa ser tratado, escolher entre os riscos, a priorização sobre o tratamento dos riscos etc.

Ainda segundo a ABNT/ISO 31010 (2012), é importante que os responsáveis da avaliação de riscos tenham informações suficientes sobre diversos tópicos como contexto, objetivos da organização, extensão, tipo de risco que são aceitos e os riscos que devem ser evitados. Devem conhecer também como é realizado o processo de integração entre a avaliação de risco e os processos organizacionais, quais são as técnicas a serem utilizados no processo de avaliação de riscos e como é feita a contribuição para o processo de gestão de risco. Outros pontos de destaque, seriam conhecer quais são os recursos que estão dispostos para o processo de avaliação de risco e como o processo de avaliação de risco será analisado.

### **3.6 Metodologia ágil**

Este capítulo tem como intuito mostrar conceitos de metodologia ágil. Segundo Middleton (2012) a metodologia ágil iniciou-se no começo dos anos 1990 em um cenário de grandes mudanças tecnológicas, cenário esse, cuja as metodologias tradicionais perdiam força devido a sua má capacidade de adaptação às mudanças de constantes de cenário. A abordagem proposta baseia-se no modelo *lean*, criado pela Toyota, em 1950, para descrever seus processos automotivos e caracterizado pela utilização de menos espaço, mão de obra, materiais e tempo para produzir seus produtos.

Highsmith (2001) conta que em fevereiro de 2001, foi realizada uma reunião com participação de 17 profissionais reconhecidos por utilizar métodos inovadores no gerenciamento de projetos. Essa reunião tinha não somente como intuito debater alternativas as grandes quantidades de processos falhos na indústria de *software*, mas também levantar as semelhanças de projetos que foram concluídos com sucesso. Dessa forma, foi criado o Manifesto Ágil, composto por um conjunto de princípios e valores baseados no modelo *lean*.

O Manifesto Ágil tem como pilar quatro valores fundamentais:

I. Indivíduos e interações mais que processos e ferramentas: esse primeiro aspecto tem como intenção priorizar as atividades humanas e suas interações diante dos processos burocráticos, ou ferramentas utilizadas durante o desenvolvimento do projeto;

II. *Software* em funcionamento mais que documentação abrangente: a premissa principal é que o cliente anseia por resultados e não burocracia, desse modo a solução deve ser mais importante do que uma documentação robusta. O método ágil não sugere o descarte da documentação, e sim a elaboração de uma fundamentação mínima necessária que dê suporte e agregue valor à solução desenvolvida;

III. Colaboração com o cliente mais que negociação de contratos: esse pilar consiste no trabalho em equipe e contato diário com o cliente, assim as decisões tomadas devem ser realizadas em conjunto, com *feedbacks* constantes;

IV. Responder a mudanças mais que seguir um plano: O Manifesto Ágil diz que o alinhamento da equipe em prazos curtos, com a finalidade de debater sobre a necessidade de atualização do planejamento e então fazer as mudanças propostas na reunião. segundo com o manifesto, essa é a ação considerada mais relevante, mais do que percorrer o planejamento inicial com êxito e disponibilizar uma solução divergente da expectativa do cliente.

Schuh (2005) define ainda que metodologia ágil é um método de desenvolvimento, *banana*, de *software* que atua por meio do empoderamento e confiança de pessoas, reconhecendo a mudança como norma e promovendo *feedbacks* constantes.

O *scrum* é uma metodologia ágil que tem como intuito a gestão e o planejamento de *software* tem como pilar as melhores práticas no desenvolvimento do projeto. Ela é usada no desenvolvimento de sistemas de maneira incremental, isto é, os requisitos sofrem mudanças constantes durante o desenvolvimento do *software* “resultando em uma abordagem que reintroduz as ideias de flexibilidade, adaptabilidade e produtividade” (SCHWABER & BEEDLE, 2002).

Segundo Sutherlnad (2009) *scrum* não é um processo ou técnica para o desenvolvimento de produtos, é na verdade um *framework* que possibilita empregar diversos processo e técnicas. O *scrum* é uma metodologia focada no gerenciamento de pessoas, mais preciso em equipes, e preocupa-se com a organização de processos e na maneira como as atividades devem ser executadas.

Segundo exposto por Leitão (2010) o desenvolvimento do *scrum* é dividido em *sprints*, ou seja, iterações que geralmente tem entre 1 e 4 semanas. Dessa maneira o *scrum* utiliza-se do conceito de atividade *time-boxes*, assim cada uma tem um tempo delimitado que não pode ser estendido, a fim de criar uma regularidade no ciclo de produção.

Para Sutherland (2009) o uso da metodologia pode ser usado nos diversos ambientes podendo ser eles equipes de projetos grandes ou pequenas, porém o tamanho considerado ideal pelo autor é de 5 a 9 pessoas. Além do mais, quando existem menos de 5 participantes no projeto, há uma menor interação entre os membros, resultando em menor ganho de produtividade. Podendo, também, ter dificuldade durante os *sprints* e atrapalhar no andamento do desenvolvimento de *software*. Outro aspecto relevante é que equipes com mais de 9 integrantes tem muita complexidade para a gestão do desenvolvimento de *software*.



## 4 ANÁLISES DAS TÉCNICAS DE AVALIAÇÃO DE RISCOS

O presente capítulo tem como objetivo apresentar e analisar técnicas de avaliação de riscos, que englobam indústrias e também as técnicas desenvolvidas para avaliação de riscos de *software*.

### 4.1 Técnicas de avaliação de riscos utilizadas no mercado

O processo de avaliação de riscos segundo a ABNT/ISO 31010 (2012) ocorre segundo a figura 2 abaixo. Ele é feito para dar aos tomadores de decisões conhecimento sobre os riscos e suas consequências. O processo de avaliação de risco se dá primordialmente com a identificação do risco, análise do risco e avaliação do risco. O modo pelo qual esse processo é feito depende tanto do contexto, quanto das técnicas utilizadas na avaliação de riscos

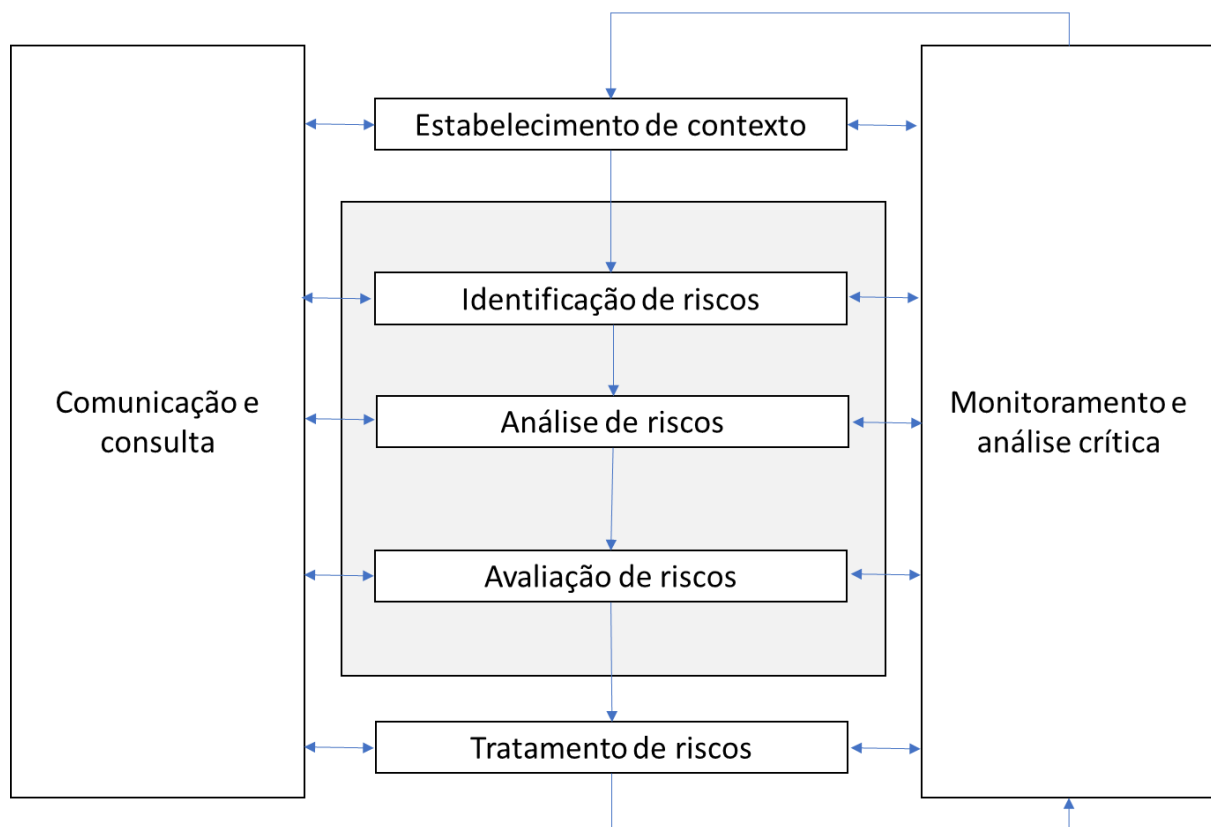


Figura 2. Contribuição do processo de avaliação de riscos para o processo de gestão de riscos adaptado ABNT/ISSO 31010

A ABNT/ISO 31000 (2009) propõe que a análise de risco é a determinação das consequências e suas probabilidades para cada risco previamente identificado, contando com a presença de métodos de controle e da eficácia dos mesmos. Relata também em seu texto que um evento pode ter impacto em mais de um objetivo e ter mais de uma consequência. Ademais, o texto segue dizendo que os métodos de análise podem ser qualitativos, semi-quantitativos ou quantitativos, a depender do projeto que será implementado.

Em seu texto a ABNT/ISO 31010 (2012) traz também informações sobre avaliação de riscos. Para a norma, avaliação é baseada na comparação entre os níveis estimados de risco, com os critérios de riscos definidos quando o contexto é estabelecido. Ainda de acordo com a ABNT/ISO, é no momento de análise do risco que se obtém a compreensão do risco, para assim tomar as atitudes futuras em relação aos riscos. Algumas dessas decisões podem ser se o risco deve ser tratado, se deve ser aceito, qual a prioridade do tratamento do risco, e se mesmo com o risco a atividade deve ser realizada e quais medidas adversas para a mitigação do risco.

“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 necessita. Isso fornece resultado atrativamente simples.” (ABNT/ISO 31010, 2012, pg 14)

Ainda sobre a norma, o texto traz uma abordagem onde a avaliação é dividida em três faixas. A primeira delas é uma faixa superior, nessa parte o risco é tido como intolerável e o tratamento é essencial qualquer que seja o custo. A parte intermediária, ou como dito área cinzenta, é onde os custos e os benefícios são tomados em consideração, realizando um *trade off* entre oportunidades e consequências. Por fim, a faixa inferior é onde o risco é considerado como desprezível, não requerendo nenhum tipo de tratamento.

A ABNT/ISO 31010 (2012) traz também em seu texto técnicas utilizadas na avaliação de risco e complementa dizendo que em alguns casos é necessário, inclusive, a utilização combinada de duas técnicas ou mais. Além disso, é relatado que a técnica deve ser apropriada a situação, que os resultados propiciem melhor entendimento sobre o risco e suas consequências e que seja utilizada de forma repetível e verificável. A seleção das técnicas deve ser escolhida segundo alguns fatores como: a necessidade da equipe de projeto, o tipo e a gama de riscos a serem analisadas, as consequências dos riscos, o potencial alcance das consequências, o nível de conhecimento especializado, os recursos humanos, a disponibilidade dos dados e informações e quaisquer outros requisitos.

Brown (2004) elenca alguns tipos de avaliação de riscos voltados para área industrial, projetos esses também encontrados na ABNT/ISO 31010 (2012), que são passíveis de serem aplicadas na etapa de elicitação de requisitos no processo de desenvolvimento de *software*. Sendo essas “*What-if*” (e - se) e FMEA (“*Fail Mode & Effect Analysis*”). Explica também que para a utilização e implementação dessas práticas é recomendado ter o comando de um guia, esse de preferência com vasto conhecimento sobre o processo de desenvolvimento de *software*.

#### **4.1.1 “*What-if*” (o que aconteceria se)**

A primeira das técnicas apresentada por Anthony E.P. Brown (2004) é chamada de “*What-If*”, ela é uma metodologia voltada para a revisão de riscos, esses são gerados a partir de assembleia da equipe de projeto de desenvolvimento de *software*. A essência dessa ferramenta está em analisar o risco por meio de exames sistemáticos onde o grupo pergunta “o que aconteceria se ...”, por conseguinte

estimulando a interação entre os integrantes favorecendo a troca de ideias. Esse modelo tem como objetivo a identificação dos possíveis riscos que não foram notados em outras etapas.

O uso do “*What-if*” é bem simples, além disso já foi aplicada de forma eficiente em diversos projetos. Os resultados gerados pelo método são de maneira geral imediatos para aplicação. Outro aspecto forte desta ferramenta é que pode ser usada em qualquer etapa do processo de desenvolvimento de *software*. Não obstante, o lado negativo é que por ser uma ferramenta extremamente intuitiva conta com o erro humano dos integrantes e gera uma grande dependência da experiência dos participantes da equipe.

Segundo Dougherty (1999) os procedimentos adotados nesse modelo utilizam um formulário similar ao mostrado no quadro 3. Geralmente este formulário é preenchido por todas as partes envolvidas no processo de desenvolvimento de *software*. A cada etapa as perguntas e respostas são feitas, “o que aconteceria se ...”, para que os problemas potenciais não sejam negligenciados. Após essa etapa, a probabilidade e o impacto são julgados por uma equipe de revisão das respostas realizadas. Por fim, concluída a análise, prioriza-se os riscos.

Quadro 3. Formulário “o que aconteceria se”

“o que aconteceria se ...”	Resposta	Probabilidade	Impacto	Medidas

A primeira parte da etapa de análise é selecionar a equipe envolvida, os limites e, principalmente, as informações. Quanto maior o escopo da revisão menos detalhada é a análise, para uma revisão efetiva é necessária uma boa definição dos limites da análise.

O equilíbrio na hora da escolha do time é essencial, deve-se buscar pessoas experientes com um elevado saber nos projetos e também novos integrantes com visão disruptiva para as análises de risco. Posteriormente, é realizado a coleta de dados e informações necessárias ao projeto e, depois de coletado, é necessário o agrupamento dessas informações.

A primeira pergunta a se responder para preencher o Formulário é “o que aconteceria se ...”. Elas podem ser desenvolvidas a partir de erros prévios, por meio de dados históricos. É de suma importância para o desenvolvimento da ferramenta que os membros da equipe tenham o ensejo de acrescentar possíveis erros ou falhas concomitantes com a elaboração das perguntas. Não se deve responder as perguntas ao passo que elas são concebidas, uma vez que existe a possibilidade de não percorrer todas as soluções possíveis.

A segunda etapa do processo é responder as perguntas previamente criadas. Essas perguntas só podem ser realizadas ao término por completo da elaboração das perguntas, onde todas as possíveis situações foram exaustivamente levantadas e revisadas. Assim a equipe deve responder: Qual seria o

possível resultado dessa situação? Essa análise tem o intuito de apontar as possíveis melhorias do projeto, outrossim minimizar problemas de qualidade.

Em seguida tem-se o preenchimento das partes de probabilidade e impacto, a missão da equipe é opinar sobre a aceitabilidade desses riscos separadamente. Ademais, a equipe deve fazer as recomendações das medidas a serem tomadas para cada circunstância, estas geralmente surgem de forma natural ao decorrer das discussões.

#### 4.1.2 FMEA (“*Fail Mode & Effect Analysis*”)

O *Fail Mode & Effect Analysis* é uma ferramenta largamente usada nas mais diversas funções, passando por todos os tipos de indústrias entre outros modelos de empresa. Ademais, Bonanomi(2012) usa o método de Vanderbrande para a aplicação da lógica da ferramenta de avaliação de risco com o intuito de prevenção dos riscos do projeto em desenvolvimento de *software*, que pode ser aplicada em qualquer etapa do processo de desenvolvimento de *software*. Por conseguinte, esse método é hábil para ser usado na fase de elicitação de requisitos.

Bonanomi (2012) aponta em seu texto como o método opera. Avalia-se as possíveis formas pelas quais um sistema ou processo está mais propício às falhas, de maneira lógica, sequencial e sistemática. Existem três variáveis que são analisadas e que antecedem ao surgimento das reclamações dos clientes: severidade das falhas, como poderiam ser detectadas e frequência da ocorrência das falhas. Sendo embasado nesses índices, o FMEA prioriza os modos de falha do produto que tem o potencial de causar um maior risco ao projeto e aos clientes.

Os quadros abaixo demonstram como as variáveis severidade, ocorrência e detecção podem ser classificadas e relacionadas a cada projeto, produto e indústria. Esse modelo serve como uma exemplificação de aplicação, isto é, ilustrando a maneira a qual o FMEA é implementado. O padrão do Quadro 4 é a classificação usada pela *Ford Motor Company*. Sendo 1 como o menos temerário e o 10 mais perigoso.

Quadro 4. Critério de Análise para Severidade das Falhas Adaptado da *FORD MOTOR COMPANY* (1988).

Severidade	Critério: Severidade do efeito	Índice de Falha
Perigoso	É muito perigosa, risco a vida ou não conformidade com legislação. Sem avABNT/ISO prévio	10
Sério	É muito perigosa, risco a vida ou não complacente com legislação. Com avABNT/ISO prévio	9
Importante	Produto inoperável, com perdas das funções básicas	8
Impactante	Desempenho do produto sofre impacto, podendo não operar	7

Significativo	Desempenho do produto é degradado. Funções podem não operar.	6
Moderado	Moderado efeito sobre o produto, requerendo reparos	5
Baixo	Pequeno defeito no desempenho do produto, não necessitando de reparos	4
Insignificante	Efeito insignificante no produto, não necessitando de reparos	3
Desprezível	Efeito insignificante no produto, não necessitando de reparos, notado pelo cliente	2
Muito desprezível	Efeito desprezível no produto, não necessitando de reparos	1

Já o Quadro 5 trata sobre a ocorrência da falha ou risco, classificando as de 1 a 10. Sendo 1 o menos perigoso e 10 a ocorrência com maior impacto para a empresa. Sendo 1 em 20 um índice alto, isto é uma taxa considerável, e 1 a cada 150.000 um índice baixo.

Quadro 5. Critério de Análise para Índice de Ocorrência das Falhas Adaptado de *FORD MOTOR COMPANY* (1988)

Ocorrência de falha	Taxas possíveis de falha	Índice de Ocorrência
Extremamente alta	$\geq 1$ em 2	10
Alta	1 em 3	9
	1 em 8	8
	1 em 20	7
Moderada	1 em 80	6
	1 em 400	5
Baixa	1 em 2.000	4
	1 em 15.000	3
	1 em 150.000	2
Quase impossível	1 em 1.500.000	1

O quadro 6 demonstra a categorização sobre a eventualidade de detectar uma falha ou risco. Assim como os demais, a detecção de falhas varia de 1 a 10, sendo 1 mais fácil de ser detectado e 10 mais difícil.

Quadro 6. Critério de Análise para Índice de Detecção das Falhas Adaptado da *FORD MOTOR COMPANY* (1988).

Detecção	Possibilidade de detecção da falhas	Índice de Falha
Totalmente incerta	Quase impossível detectar a falha	10
Muito remota	Chance muito remota de detectar a falha	9
Remota	Chance remota de detectar a falha	8
Muito baixa	Chance muito baixa de detectar a falha	7
Baixa	Chance baixa de detectar a falha	6
Moderada	Há moderada chance de detectar a falha	5
Moderadamente alta	Moderadamente alta de se detectar a falha	4
Alta	Alta chance de se detectar a falha	3
Muito alta	Há chance muito alta de se detectar a falha	2
Quase certa	É quase certo que a falha será detectada	1

A fórmula abaixo representa a forma pela qual o risco é calculado. Esta escala é chamada em inglês de *Risk Priority Number* (RPN) ou, em português, número da prioridade do risco (NPR). Será adotada a sigla NPR, por ser do idioma brasileiro, por meio da expressão (1).

$$\text{NPR} = \text{O} \times \text{S} \times \text{D} \quad (1)$$

Onde:

- a) RPN: *Risk Priority Number*;
- b) O: ocorrência;
- c) S: severidade;
- d) D: detecção.

Destrinchando a fórmula, temos que a multiplicação das pontuações apanhados na categorização da ocorrência da falha ou risco (O), severidade das falhas ou risco (S) e possibilidade de detectar uma falha ou risco (D). Esta é a técnica mais empregada para mensurar o risco relacionado à cada falha. A escala deste método varia de 1 até 1000 pontos, onde este é um risco altamente danoso e aquele um baixíssimo risco. Portanto, para um risco ser tido como alto é necessária uma alta probabilidade de ocorrência, um vultoso impacto no produto e pouca capacidade de ser detectado.

## 4.2 Técnicas de avaliação de riscos do processo de desenvolvimento de *software*

Esta parte do projeto de graduação tem como objetivo esclarecer sobre as técnicas de avaliação de riscos do processo de desenvolvimento de *software*. Para PRESSMAN (2016), a análise começa enumerando todos os possíveis riscos, mesmo que esses sejam improváveis de acontecer. Os riscos são dispostos na primeira coluna de uma tabela que será desenvolvida para categorização das ameaças.

Esses riscos podem ser listados por meio de ferramentas como o *brainstorming*, experiência anterior ou com uma lista de riscos previamente concebida por outros autores e/ou desenvolvedores. Decomposição, exame de tomadores de decisão, comparação com a experiência e *checklist* padronizados disponíveis são possíveis técnicas de identificação de variáveis de risco (BOEHM, 1991). A etapa seguinte é inserir nas colunas adjacentes à de riscos, a probabilidade e o impacto por elas causadas, que serão dispostos para cada risco e serão estimadas por meio de experiência anterior da equipe de desenvolvimento. Após essa etapa deve-se ordenar a tabela por multiplicação da probabilidade por impacto.

Para o Leopoldino (2004), a análise dos riscos relacionados ao processo de desenvolvimento de *software* é cíclica, isso se dá devido ao ambiente dinâmico em que vivemos. Esse ambiente pode gerar ou desaparecer com fatores malquistos quanto ao grau de probabilidade e impacto. SWEEBOK (2014) afirma que avaliação é identificar os riscos mais críticos, para isso se faz necessário constatar as possíveis ameaças que estão inseridas nos projetos internos e externos do desenvolvimento de *software*. Deve se definir um *ranking* dos riscos que causam mais impacto.

Leopoldino(2004) ainda escreveu que o processo de avaliação de risco deve ser contínuo e cíclico, pois os ambientes tendem a surgir e desaparecer com novas ameaças. Como exemplificação disso, temos os riscos temporais: na medida em que o desenvolvimento do *software* se aproxima de seu fim existe menos tempo disponível da equipe.

Para a realização de uma boa análise de risco, segundo Sommerville (2011), é necessário conceber juízo de valor referente ao impacto e a probabilidade dos riscos na elicitação de requisitos. Para o autor, isso é feito baseando-se no próprio julgamento e na experiência adquirida ao decorrer dos projetos pretéritos. Ele diz também que não há uma forma correta de definir tais limites e que não há uma avaliação totalmente acurada da probabilidade e gravidade dos riscos. No quadro 7, Sommerville (2011) demonstra uma divisão de probabilidade e avaliação do risco.

Realizar a análise dos riscos obtidos é o primeiro passo, deve-se fazer isso mensurando os resultados do método e inserindo em uma tabela ordenada de acordo com o efeito e probabilidade do risco. É nessa etapa que acontece a avaliação arbitrária do risco, isso é feito para o impacto e a probabilidade. Para essa avaliação faz-se necessária a busca de informações prévias e detalhadas sobre as etapas de desenvolvimento e seus atores.

Quadro 7. Probabilidade

Probabilidade do risco	Avaliação
< 10%	Muito baixa
10 a 25%	Baixa
25 a 50%	Moderada
50 a 75%	Alta
> 75%	Muito alta

Já o quadro 8, dispõe como seus efeitos podem ser avaliados na visão do autor.

Quadro 8. Impacto do risco

Efeito do risco	Avaliação
Catastróficos	Ameaçam a sobrevivência do projeto
Graves	Causariam grandes atrasos
Toleráveis	os atrasos estão dentro da contingência permitida
Insignificantes	Autor não cita nada sobre os riscos insignificantes

Uma vez que os riscos foram analisados e classificados, você deve avaliar quais desses riscos são mais significativos. Seu julgamento deve depender de uma combinação da probabilidade de ocorrência de risco e dos efeitos desse risco. Em geral, os riscos catastróficos devem ser sempre considerados, bem como os riscos graves que têm mais do que uma probabilidade moderada de ocorrência. SOMMERVILLE (2011, pg 416).

Pressman (2016) diz que se deve traçar uma linha de corte, o que implicará que apenas os riscos que estiverem sobre a linha receberão maior atenção por parte do desenvolvedor, e para os demais riscos deverá ser feita uma reavaliação, ou seja, uma priorização de segunda ordem.

Mateus Macedo e Eduardo Salgado (2005) trazem na revista eletrônica “Sistemas e Gestão” diferentes conceitos na análise qualitativa do risco. O primeiro desses conceitos é o entendimento onde:

“Os riscos são analisados primordialmente através da probabilidade e impacto. A probabilidade é a possibilidade de ocorrência, que é a razão do número de chances pela qual um evento de risco pode acontecer (ou não acontecer) pela soma das chances do acontecimento (ou não acontecimento). O impacto pode ser positivo ou negativo



nos objetivos planejados dependendo do evento. A análise de impacto inclui a quantificação de suas respectivas severidades de impacto, probabilidade e sensibilidade de mudanças. (CHARETTE, 2005, SCHUYLER, 2001)

Para o Sweebok(2014), identificar os mais críticos consiste na avaliação dos riscos. Para qualificar os riscos basta identificar os perigos que fazem parte dos ambientes interno e externo ao *software* que está sendo aplicado e após isso definir uma lista ordenada de prioridade dos riscos mais relevantes.

Assim como a análise de risco, a avaliação requer tempo para a aplicação e experiência de projeto. A avaliação de riscos é o passo seguinte a análise e a priorização dos riscos feita anteriormente. Essa etapa de planejamento das ameaças tem como intuito a concepção de estratégias para avaliar cada risco identificado individualmente. “Selecionar a opção mais adequada de tratamento de riscos envolve equilibrar, de um lado, os custos e os esforços de implementação e, de outro, os benefícios decorrentes” (ABNT/ISO 31.00, 2009, pg 19)

A finalidade da avaliação de riscos é auxiliar na tomada de decisões com base nos resultados da análise de riscos, sobre quais riscos necessitam de tratamento e a prioridade para a implementação do tratamento. ABNT/ISO 31.000 (2009, pg 18)

Segundo Sommerville (2011), caso ocorra um problema já identificado no risco, é necessário que se tome ações capazes de reduzir ao máximo as interrupções do desenvolvimento de *software*, esse tratamento deve ser feito para cada risco individualmente.

Você também precisa pensar sobre as informações que pode precisar coletar durante a monitoração do projeto de maneira que os problemas possam ser antecipados. Não existe um processo simples que possa ser seguido para o planejamento de contingência. Ele usa o julgamento e a experiência do gerente de projeto. SOMMERVILLE (2011, pg 420).

Já Prado Felipe, *et all* (2009) aborda no texto “Análise de Risco em Projetos de *Software*” duas formas de se tratar os riscos do desenvolvimento de *software*: por métodos qualitativos e quantitativos. A primeira etapa é avaliar quais são as probabilidades e os impactos que são gerados por cada risco identificado. Posteriormente, faz-se uma matriz de probabilidade vs. impacto assemelhado ao quadro 9 abaixo. Esse quadro é feito para a realizar-se a avaliação dos riscos. O próximo passo, é a análise quantitativa. Alguns dos motivos principais para a realização dessa análise quantitativa são listados abaixo:

- Quantificar as probabilidades e os possíveis resultados do projeto;
- a) Realizar uma avaliação sobre os possíveis resultados de atingir objetivos específicos do projeto;
  - b) Realizar a identificação dos riscos que requerem um maior cuidado, para mensurar sua contribuição ao risco total;

- c) Definir as melhores escolhas de avaliação do projeto quando houver condições ou resultados com um certo grau de incerteza.

O quadro 9 abaixo, matriz de probabilidade e impacto, serve para avaliar as consequências dos riscos para o projeto. Esse quadro, proposto pelo PMI (2013), assemelha-se com o relatado pelo autor Prado Felipe anteriormente. Essa técnica auxilia de forma visual o gerente de projeto a fazer uma avaliação dos riscos que perpassam o desenvolvimento do *software*. Na parte superior os riscos devem ser evitados, já na parte central os riscos devem ser mitigados, na parte esquerda inferior é o local onde os riscos devem ser aceitos. Por se tratar de uma forma visual de avaliação de risco existe uma facilidade na compreensão do impacto do risco para o andamento do projeto.

Quadro 9. probabilidade VS impacto retirado do PMI (2013)

PROBABILIDADE					
Muito alta					
Alta					Evitar
Média	Aceitar				
Baixa				Mitigar	
Remota					
	Muito baixo	Baixo	Médio	Alto	Muito alto
GRAU DE IMPACTO					

O valor das probabilidades e dos impactos variam de 1 a 5, o quadro 10 abaixo representa a avaliação da probabilidade e do impacto segundo o modelo proposto. Assim como Pressman, a avaliação do impacto possui apenas 5 aspectos de avaliação.

Quadro 10. Avaliação probabilidade vs impacto retirado do PMI (2013)

Valor da probabilidade/impacto	Avaliação Impacto	Avaliação Probabilidade
1	Muito baixa	Remota
2	Baixa	Baixa
3	Moderada	Média
4	Alta	Alta
5	Muito alta	Muito alta

Existem três tipos de estratégia principais para o tratamento do risco segundo SOMMERVILLE (2011), podendo-se criar estratégias para prevenir de que aconteçam, criar também estratégias que

possam minimizar o seu acontecimento e por último criar planos de contingência do risco. O quadro 10 é uma ilustração de como proceder, esse modelo foi retirado e proposto por SOMMERVILLE (2011) em seu livro.

Obviamente, é melhor usar uma estratégia que evite o risco. Se isso não for possível, você deve usar uma estratégia que reduza as chances de riscos com efeitos graves. Finalmente, você deve ter estratégias preparadas para lidar com os riscos caso eles apareçam. Elas devem reduzir o impacto global de um risco relativo ao projeto ou produto. SOMMERVILLE (2011, pg 420)

Entende-se que existem diversas formas de avaliação de risco que podem ser aplicadas separadamente para todas as etapas do desenvolvimento de *software*. Esse, permeia por diversos riscos, principalmente no momento de elicitação de requisitos, porém essas técnicas de avaliação estão dispostas para permitir que os projetos corram de forma linear com o menor nível de impactos negativos para o bom andamento do projeto possível.

## 5 ESTABELECIMENTO DO CONTEXTO DA PESQUISA

Neste capítulo será apresentado o contexto de desenvolvimento baseado na ABNT/ISO 31000 (2009) e tem-se o objetivo de documentar processo de desenvolvimento de *software* e o *software* da empresa desenvolvedora do *software*.

### 5.1 Contexto Interno

De acordo com a ABNT/ISO 31000 (2009), para se caracterizar o contexto interno de uma empresa é necessário o entendimento sobre as capacidades da organização, de recurso e de conhecimentos, das partes internas, dos objetivos e das estratégias que estão em vigor, dos valores e cultura, das políticas e processos etc.

A empresa de desenvolvimento é focada em desenvolvimento de *software*, teve seu início no ano de 2017, e conta com dois desenvolvedores principais e alguns *freelancers*, quando necessário. É um negócio pequeno, voltado para o desenvolvimento de *softwares* de médio e pequeno porte, como a criação de aplicativos para *mobile* e *websites*. Os projetos realizados pela empresa em sua maioria são ágeis, ou seja, requerem pouca documentação. Os projetos são desenvolvidos sem documentação, todo o processo de desenvolvimento de *software* dá-se de maneira informal, devido à falta de tempo para realizar serviços burocráticos de documentação da equipe.

A empresa tem como objetivos principais de curto prazo, a permanência no mercado e difusão do nome da empresa, a fim de angariar recursos e expandir o negócio. É uma empresa jovem com uma cultura organizacional voltada para pessoas, a empresa dá atenção aos talentos individuais e valoriza as ideias, com intuito de fortalecer os laços de amizade entre os colegas e fornecedores da empresa. Os valores da empresa são voltados para simplicidade e progresso, pois os desenvolvedores têm a crença de que a simplicidade e evolução constante levam o negócio a frente.

### 5.2 Contexto Externo

Ainda segundo a ABNT/ISO 31000 (2009), para realizar a caracterização do contexto externo da empresa, requer-se o conhecimento sobre fatores culturais, políticos, legais, fatores chave e tendências que tenham impacto sobre a organização e percepção do mercado. O mercado atual de plataformas online para concursos está em pleno desenvolvimento, isso se dá principalmente com a alta de concursos públicos nos últimos anos. Essa alta procura dá-se por parte dos estudantes e dos governos federal e regionais, pois buscam alternativas as altas taxas de desemprego formal no mercado privado, aproximadamente 14% somente no ano de 2018 segundo pesquisas do IBGE (2018). Uma alternativa para a população jovem está sendo estudar para os concursos públicos que auferem a eles grandes vantagens financeiras.

Por conseguinte, existe uma necessidade latente de interação digital dos professores com o estudante por parte dos coachings e cursinhos voltados para concursos públicos. Porque nem todos os

estudantes têm acesso a aulas presenciais, dessa forma um website tem a possibilidade de alcançar mais clientes e suprir essa demanda. Ademais, o mercado de concurso público é bastante competitivo, desse modo, as empresas precisam agregar valor aos seus produtos, para obterem uma vantagem competitiva sobre os demais concorrentes. Uma boa estratégia é a implementação de um website de fácil uso e com bons materiais.

A proposta inicial do *software* a ser desenvolvido pela equipe de projeto é de um sítio eletrônico (*website*) e um aplicativo para celulares (*mobile app*) voltado para estudantes de concurso público. O website trará conteúdos de julgados comentados por professores, fornecendo material de estudo para estudantes. A equipe é composta por dois desenvolvedores formados em engenharia de *software* pela Universidade de Brasília. Este projeto foi proposto à equipe por uma juíza (*Project Owner*), um auditor do Tribunal de Contas da União (*Project Management*) e outro cliente. Eles possuem uma empresa de coaching para concursos públicos. Este projeto tinha com o intuito inicial suprir as demandas de estudantes, porém com o grande volume de requisição por parte dos alunos, os empresários precisam expandir seu negócio, para aumentar seu *marketing share*, e dessa forma aumentar sua receita.

### **5.3 Desenho do processo de desenvolvimento do *software* utilizado no contexto**

Essa parte tem por objetivo explicar como foi realizada a documentação do processo de desenvolvimento do produto e, também, documentar o mesmo. Para elaborar a documentação da metodologia utilizada, optou-se pela inserção no ambiente de desenvolvimento de *software* do contexto do desenvolvimento em questão. Essa inserção ocorreu no período de aproximadamente 1 mês, com reuniões diárias. Para coletar as informações, foi utilizado o método de entrevista não estruturada. Não se usou nenhum questionário formal para a realização das perguntas. A documentação se deu por meio de um desenvolvedor explicando como ocorria o processo de desenvolvimento de *software* e, principalmente, por observação das atividades.

Como visto, os programadores optaram por usar a metodologia ágil em seu processo de desenvolvimento de *software* no desenvolvimento do sítio eletrônico com conteúdo voltado para concursos. Um dos motivos para os desenvolvedores adotarem essa metodologia foi o baixo valor agregado para o produto e para o cliente nas metodologias tradicionais. Esses métodos tradicionais requerem uma grande documentação por parte dos desenvolvedores, desse modo toma-se tempo crucial de desenvolvimento do *software*, que ao final será pouco útil para os mesmos. Por se tratar de duas pessoas desenvolvendo a parte dos códigos do projeto, os métodos ágeis auxiliam na evolução mais rápida do produto com uma entrega com menor prazo do que as metodologias tradicionais.

Outra razão pela escolha dessa opção foi, não somente de ser mais flexível a mudança e assim tornar-se mais viável economicamente para o cliente, mas também minimizar os riscos envolvidos, principalmente na parte de engenharia de requisitos. Projetos robustos, principalmente na área de tecnologia, acarretam maiores probabilidades de acontecerem os riscos já identificados nas fases primordiais do desenvolvimento de *software*, e com isso surgir impactos extremos para o

desenvolvimento produto. Por esse motivo, existe pouca documentação por parte dos desenvolvedores de todo o processo de desenvolvimento do *software*.

Nessa etapa, foi gerado em conjunto com os desenvolvedores a figura 3 abaixo. Essa representa o processo de desenvolvimento de *software* do contexto de desenvolvimento exposto neste trabalho. Para a edição visual dessa ferramenta usou-se o Adobe Photoshop, criando um esboço do processo de desenvolvimento de *software*. Existem 5 fases principais neste projeto, sendo elas entrada, preparação, execução, validação e liberação. Cada uma exerce uma parte muito fundamental para o desenvolvimento do *software*, porém para o trabalho as fases que mais afetam os riscos envolvidos na elicitação de requisitos estão inseridas na parte de preparação e na parte de desenvolvimento.

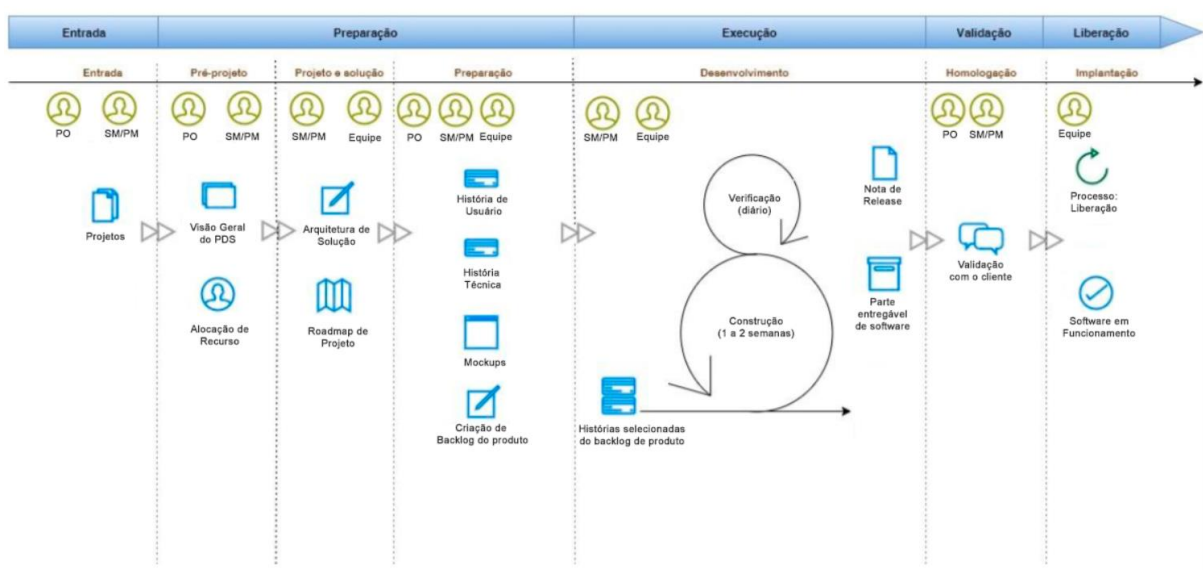


Figura 2 Processo de desenvolvimento de *software* do contexto estudado. Figura autoral.

Como entrada, tem-se o projeto, que foi a proposição realizada pelo *Project Owner* (PO) e *Project Manager* (PM). A seguir, o pré-projeto foi uma visão geral do processo de desenvolvimento de *software*, e como seriam alocados os recursos financeiros e humanos. O passo seguinte é o desenvolvimento de uma arquitetura de solução e o *roadmap* do projeto, isso é, uma visão de como estaria o produto em suas etapas seguintes.

A parte mais importante para este trabalho foi a preparação onde equipe, PO e PM se reuniram para captar os requisitos criando a história de usuário. Esse é o principal documento de elicitação de requisitos. Na outra etapa, os requisitos levantados eram refinados semanalmente pelas histórias selecionadas de backlog do produto, onde se entregava uma parte do *software* ao PO que o validava na etapa de homologação do produto.

#### 5.4 Descrição das atividades de elicitação de requisitos

Para gerar a documentação do processo de elicitação de requisitos do contexto da empresa de desenvolvimento de *software*, foram realizadas diversas reuniões. O principal método de documentação da elicitação de requisitos se deu por meio de observação nas reuniões semanais. Contudo, as etapas

anteriores às documentações tiveram de ser explicadas pelos desenvolvedores. Para documentar usou-se a ferramenta do *Microsoft Word* para editar os tópicos no momento da entrevista, após isso, elaborou-se a documentação formal, a partir dos tópicos criados no momento da entrevista.

Para a primeira parte da documentação, o desenvolvedor explicou que o levantamento de requisitos teve sua primeira fase na reunião inicial, onde estavam presentes os desenvolvedores, o auditor do TCU, que é o *Project Manager* do projeto, e a juíza, que é o *Project Owner* do projeto. Nessa reunião foi definido o *Product Backlog*, abrangendo todos os épicos do produto - uma lista contendo todas as funcionalidades desejadas para o produto. Os épicos definidos relataram como seria a área administrativa, um pouco sobre a parte do cliente e, o mais importante para os clientes, os métodos de buscas. Esses épicos foram definidos pela juíza, porém não foram exploradas todas as funcionalidades na primeira reunião.

Os passos seguintes são encontros, que de maneira geral, ocorrem mensalmente entre o PO e o PM, a fim de levantar e refinar as necessidades do cliente. Isso se dá porque o auditor possui habilidades de intermediário, entre a juíza e a equipe de desenvolvimento, por possuir conhecimentos em desenvolvimento de *software* e também das características dos produtos e das informações contidas nele.

Por meio de observação, viu-se que depois das reuniões o auditor se encontra semanalmente com a equipe de desenvolvimento, a fim de validar o que foi realizado. Também para repassar os requisitos levantados e assim criar o *sprint backlog*, isto é a lista de tarefas que os programadores se comprometem a realizar no *Sprint* daquela semana. Essas exigências são refinadas e detalhadas, as definições dos requisitos são bem flexíveis, podendo ser alteradas por ideias propostas pelos desenvolvedores e reconhecidas pelo *Project Owner*.

Por fim, ocorre uma reunião semanal entre todas as partes envolvidas. Nesse compromisso, os desenvolvedores apresentam a juíza e ao auditor os requisitos previamente levantados que já foram desenvolvidos com o intuito de validar novamente e atestar se as exigências foram atendidas conforme as expectativas, ou se precisam ser alteradas por superveniência de novas necessidades do cliente. Eles também refinam os requisitos que por algum acaso não foram bem compreendidos, ou que precisam ser alterados por uma das partes, tanto para adequar a novas necessidades dos clientes quanto por impossibilidade de desenvolver tal quesito.

A criação da documentação nessa parte, foi de grande valia para o sucesso do desenvolvimento deste projeto de graduação. Porque permitiu uma maior compreensão sobre os aspectos da elicitação de requisitos do contexto da empresa estudada. O método de observação foi necessário, porque o desenvolvedor fantasiava sobre o processo de elicitação por medo de parecer simples. Assim, os métodos utilizados foram extremamente importantes para a real documentação da elicitação de requisitos.

## 5.5 Descrição das *features* do *software*

Neste item, apresentam-se os resultados coletados nas reuniões. A plataforma *web* proposta consiste de duas áreas: a primeira, uma região administrativa onde os donos da empresa de coaching e seus funcionários irão cadastrar os insumos de estudo e insumos do cliente, e a segunda, compreende um espaço no qual os estudantes irão acessar as informações empregadas por parte da equipe. Essas informações serão disponibilizadas sobre demanda para eles. Existem três unidades organizacionais envolvidas na concepção do *software*, são elas: cliente, layout e equipe de desenvolvimento.

Os insumos a serem cadastrados no sítio eletrônico, os insumos de estudos que se constituem de jurisprudências de tribunais, serão tratados pelos professores e funcionários. O tratamento desses julgados, constitui-se de uma seleção dos mais importantes para os estudantes, a juíza, posteriormente, realiza um resumo e cadastra o julgado. Dessa forma, torna essas informações finais em documentos. Esses documentos possuem como informação: órgão Julgador (ex: Supremo Tribunal Federal, Supremo Tribunal de Justiça), ano (ex: 2018), numeração, texto livre, palavra-chave, resumo e ementa sem formatação.

As informações dos documentos serão concebidas, a partir dos dados dos informativos, esses são cedidas pelos tribunais. Cada Informativo, possui ao menos uma jurisprudência, podendo dispor de 'n' julgados, com cada um possuindo seu resumo e sua avaliação por parte do professor. Ele será categorizado em disciplinas pautadas nos ramos do direito. Essas disciplinas apresentaram ramos e os ramos deterão de tópicos e os tópicos portarão subtópicos, com o intuito de caracterizar os julgados já cadastrados.

A outra área administrativa que requer a inserção de dados por parte da consultoria é a do cadastro dos alunos. Para isso o cadastro do estudante no portal irá recrutar os dados: nome completo, e-mail, telefone, endereço, data de nascimento, sexo, CPF e dados do cartão de crédito.

## 5.6 Funcionamento do *software*

Esta é a parte da documentação responsável pelo funcionamento do *software*. A documentação foi elaborada, a partir de reuniões, e por meio de observação das telas do *software*. O produto em questão possui ao todo 8 partes principais, que serão descritas, uma a uma, ao longo deste estudo. Todas as funcionalidades apresentadas são requisitos que foram levantados ou estão sendo gerados pelo cliente. As partes do processo são:

- a) *Landing page*;
- b) *Sign up*;
- c) *Login*;
- d) Área administrativa;
- e) Área do cliente;
- f) Resultado da pesquisa;
- g) Caixa do julgado;



h) Minha conta.

Todos as partes foram documentadas de maneira descritiva nos tópicos a seguir, sendo de suma importância para a compreensão do funcionamento do *software*. As descrições foram obtidas ou por meio de descrição verbal pela equipe de projeto, ou por meio de anotações realizadas a partir das conversas tidas com a equipe de projeto durante as reuniões.

### 5.6.1 “Landing page”

Nesta atividade, a partir das reuniões, percebeu-se que a primeira das partes é a “*landing page*”, ela é a página de apresentação. O usuário tem a oportunidade de ver informações genéricas sobre os conteúdos do site, a fim de converter o visitante em cliente. Na *landing page* existem vídeos informativos dos conteúdos dos julgados, esses curta metragens são atrativos para a conversão do visitante.

*Link* que dá acesso ao *blog*, esse blog contém informações da pessoa guia do julgado com todas as informações da carreira, que dá credibilidade às informações por eles postadas. Há também alguns comentários sobre leis e algumas informações extras voltadas ao universo do concurso.

Há também o contato para equipe, esse contato inclui *e-mail* e direcionamento para o *blog* pessoal. Outra parte da página inicial são os links que direcionam o usuário para o *Sign up* e, se já for cadastrado, para o login.

### 5.6.2 “Sign up”

Essa é a parte de cadastramento do usuário, documentada a partir das entrevistas. Podendo ser o usuário estudante ou professor. Os processos de preenchimento do formulário serão similares, com algumas diferenças que serão explicadas a seguir. Para o cadastro do estudante serão necessários: os dados de Nome completo, E-mail válido para envio de informação, Telefone, Endereço, Data de nascimento, Sexo, CPF e Dados do cartão de crédito. Já na parte do professor excluem-se apenas os dados do cartão de crédito, já que são desnecessários, porém necessita-se dos dados do *blog* pessoal, que são relacionadas a carreira acadêmica e profissional.

Após o preenchimento completo do formulário, o sistema pede ao usuário a criação de um login e senha para futuros acessos. Após, há a validação da forma de pagamento e, por fim, uma confirmação final da conta. O formulário é submetido para o sistema que, como passo imediato, exhibe uma tela de confirmação, isso vai requerer uma confirmação no *e-mail* por parte do usuário cadastrado, assim finalizando a tarefa de inscrição do usuário.

### 5.6.3 “Login”

O modo de acesso pelo usuário às regiões restritas, acontece quando o estudante ou professor, previamente cadastrados, inserem seus *login* e senha em caixas de texto. Essa é a etapa que requer a autenticação do usuário já cadastrado no sistema, que dá acesso às duas partes principais do sistema: área administrativa e área do cliente. A área administrativa é restrita somente a professores e

funcionários da empresa, desse modo estudantes não terão acesso a esse ambiente. A área do cliente pode ser acessada por ambas as partes, permitindo assim acesso dos professores a esse ambiente.

#### **5.6.4 Área administrativa**

Essa região como dito anteriormente é restrita para certos perfis. Ela possui cadastro de informativos, cadastro de julgados, cadastro de disciplina, cadastro de ramos, cadastro de tópicos, cadastro de órgão julgador e cadastro de subtópicos abaixo é detalhado cada área.

##### **5.6.4.1 Cadastro informativo**

O cadastro do informativo é realizado pelos funcionários da consultoria. Essa parte do *software* contém informações como as quantidades de informativos já existentes no sistema, quantidade de julgados já inseridos no sistema anteriormente por outros usuários colaboradores. Ademais, possui os julgados de forma sequenciada em uma caixa que diz se o julgado está publicado ou não para o estudante. É nessa parte que o professor cadastra no sítio eletrônico as informações sobre o informativo. Para o cadastro, o sistema requer do responsável por inserir informações cruciais para a categorização do informativo, essas informações estão dispostas abaixo:

- a) Número do informativo;
- b) Ano;
- c) Publicado ou não para o estudante;
- d) Quantidade de julgados.

Os dois primeiros dados estão dispostos em forma de texto, sendo o primeiro deles sequencial e progressivo, isto é os informativos cadastrados representaram uma sequência cronológica de inserção dos mesmo no website. Já o segundo, é um dado booleano, isso significa que possui apenas duas possíveis informações sim ou não. Outra informação necessária para o entendimento, é entender que o número do informativo é lançado pelo órgão julgador e esse informativo possui o número do órgão julgador. Todos esses dados podem ser transformados nas quatro operações básicas *Create, Read, Update e Delete* (CRUD).

##### **5.6.4.2 Cadastrar julgado**

Essa é uma das fases mais importantes requisitadas pelo cliente, os funcionários ou professores são responsáveis nesse momento por cadastrar os julgados dispostos nos informativos cedidos pelos tribunais. Para a inserção dos julgados é necessário registrar no banco de dados em forma de caixa de texto as informações a seguir:

- a) Texto livre;
- b) Palavra chave;
- c) Resumo;
- d) Ementa sem formatação.

Texto livre é a parte na qual o professor agrega valor ao julgado, nessa parte ele insere no sistema comentários dos julgados e faz o juízo de valor sobre os conteúdos presentes na jurisprudência. A palavra-chave auxilia na busca por parte dos estudantes aos conteúdos dispostos no website. O resumo é onde o professor sintetiza as informações dispostas sem realizar julgamentos.

A próxima etapa é selecionar as disciplinas que os julgados pertencem, deve-se também escolher os ramos que são um nível abaixo das disciplinas. Em seguida o operador escolhe o tópico baseado nos ramos das matérias selecionadas, a seguir são escolhidos os subtópicos – que tem como critério base os tópicos anteriormente selecionados. Essas escolhas são hierárquicas e sucessivas, os ramos, tópicos e subtópicos foram previamente cadastrados pelo usuário.

Os próximos campos a serem preenchidos são booleanos, com respostas sim ou não. Devem ser assinalados os campos de repercussão geral, repetitivo, súmula vinculante, incidente de assunção de competência e relevância. Todas as informações podem ser operadas pelo CRUD.

#### **5.6.4.3 Cadastro de disciplina**

Nessa etapa o operador tem a possibilidade de cadastrar as disciplinas, essas matérias estão relacionadas com as grandes áreas do direito. Outro fator que deve ser realizado pelo professor é a descrição da disciplina, que é obrigatória para a criação da mesma. O sítio já possui algumas disciplinas criadas pelos desenvolvedores. O usuário tem o poder de criar novas matérias, ler quais estão disponíveis, atualizar e, por fim, deletar as que não forem mais necessárias.

#### **5.6.4.4 Cadastro de ramos**

O papel dos ramos, assim como o das disciplinas, é de facilitar a busca. Eles fazem isso criando filtros para a pesquisa realizada pelo estudante. Os ramos podem ser gerenciados pelas operações CRUD. O usuário insere o nome do ramo no sistema e realiza uma breve descrição dos mesmos, essas duas operações são obrigatórias para a gênese do ramo. O próximo passo é colocar a disciplina característica daquele ramo.

#### **5.6.4.5 Cadastrar tópicos**

A operação de cadastramento dos tópicos é bem similar a realizada nos ramos, o professor insere o nome do tópico e atrela esse a um ramo. Não é necessário escolher a disciplina, pois o ID salvo do ramo já está ligado a uma disciplina. O usuário pode usar as quatro operações básicas nessa etapa.

#### **5.6.4.6 Cadastrar subtópico**

Esse é o último nível de filtro dos julgados, nesta etapa o professor insere o nome do subtópico no sistema e atrela esse a um tópico. Não necessita-se atrelar a um ramo, ou disciplina pois obedece a uma ordem hierárquica. Todas as informações podem ser operadas pelo CRUD. Nessa etapa termina o cadastramento de informativo.

### **5.6.1 Área do cliente**

Essa é a região na qual o estudante, ou seja, o cliente, da empresa contratante tem acesso a todas as informações cadastradas pela juíza ou pelos professores. Essa parte do *software* possui a área de pesquisa, resultado da pesquisa, caixa do julgado, minha conta e tipo de estudo.

#### **5.6.1.1 Área de pesquisa**

É nessa etapa que o aluno tem acesso a pesquisa dos julgados, existem 3 abas de pesquisa sendo a primeira a busca por julgado, a segunda busca por informativo e, por fim, a busca avançada - que é um pouco mais refinada que as anteriores. Todas essas abas foram necessidades dos donos da empresa de coaching e propostas aos desenvolvedores.

Na primeira das abas, o estudante tem a possibilidade de escolher disciplina, ramo, tópico e subtópico. Já na busca por informativos, o usuário tem como possíveis filtros a escolha de número do informativo, ano, tribunal e palavra-chave. Na terceira parte, os filtros são mais numerosos começando por: tribunal, ano, número do informativo, órgão julgador, disciplina, ramo, tópico, subtópico, palavra-chave e repercussão geral.

#### **5.6.1.2 Resultado da pesquisa**

Essa etapa é a sucessora da busca anterior, nesse momento o estudante tem acesso a todos os julgados compatíveis a procura realizada. A tela do resultado de pesquisa contempla a quantidade de julgados que a busca retornou e o julgado em destaque – que é escolhido pelo professor na hora do cadastro. Contempla também o número do julgado. Outros resultados que retornam dos critérios inseridos são ano, número e órgão do informativo. Também retorna palavras-chave e resumo. O estudante pode realizar certas operações nesse momento, entre elas gerar um PDF da pesquisa, editar a última procura e criar um plano de estudo a partir da consulta.

Os julgados são apresentados em caixas que, quando o estudante clica nelas, direcionam a uma outra área onde esse julgado está mais detalhado e contém todas as informações relativas a jurisprudência escolhida, o estudante tem a possibilidade de saber as palavras chaves disposta, a ementa sem formatação e o resumo cadastrado pelo professor. Essa caixa possui um *checkbox* onde o usuário pode definir se já estudou ou não a matéria, mas para realizar tal operação é necessário estar logado no sítio eletrônico.

#### **5.6.2 Minha conta**

O usuário tem acesso a todas suas informações cadastrais previamente inseridas no sistema na hora de adentrar a região restrita a estudantes pela primeira vez. Há também a possibilidade de atualizar e deletar tais informações, é lá que o estudante pode recomendar o sítio eletrônico a outras pessoas. Também pode alterar suas informações de estudos nas ferramentas: tipo de estudo, meus estudos e preciso reler.

As informações contidas na aba preciso reler, são os julgados que já foram lidos, porém foram marcados como leitura posterior pelo aluno. Nos tipos de estudo, o cliente pode selecionar os estudos

por disciplina, ramo, tópico e subtópico e assim gerar um caderno para estudos futuros. As informações do caderno são armazenadas na região dos meus estudos, onde o estudante tem acesso a todas as informações armazenadas. Ele pode apenas deletar o caderno, não podendo atualizar. A atualização do caderno é um requisito pedido pela juíza, mas que está em desenvolvimento ainda.

Por fim, essa etapa do contexto de desenvolvimento traz uma visão global sobre a empresa, seu cliente e o *software* desenvolvido. Essa fase é muito importante para entender o contexto no qual a ferramenta será aplicada, pois somente assim têm-se a percepção das dimensões dos riscos, das probabilidades e, principalmente, do impacto que esses geram no desenvolvimento do *software* e de como a ferramenta auxilia na avaliação desses riscos.

## 6 DEFINIÇÃO DOS CRITÉRIOS DE AVALIAÇÃO DOS RISCOS

Este capítulo tem por objetivo apresentar uma lista dos riscos envolvidos na elicitação de requisitos. Além disso, tem como objetivo editar a ferramenta de análise e avaliação de risco. A primeira das etapas da análise e avaliação de riscos da empresa estudada ocorreu concomitantemente com o desenvolvimento da ferramenta.

### 6.1 Compilação da lista de riscos na elicitação de requisitos

Tem-se o objetivo de compilar em uma lista os riscos envolvidos na elicitação de requisitos. Para realizar avaliação de riscos requer-se uma identificação prévia dos mesmos. Existem diversas formas de realizar essa identificação, tais como entrevista com especialistas, *brainstorming*, técnica *delphi*, técnica de grupo nominal, *slip de Crawford*, analogia, *checklist*, questionários e modelos de identificações, como disse Engolhm (2013).

Para a criação da lista de riscos do contexto estudado, foi adotada a técnica de analogia, sendo o primeiro passo a definição do escopo, identificando e selecionando somente os riscos relacionados a etapa de elicitação de requisitos, focando principalmente em metodologias ágeis. Após isso definiu-se uma lista de artigos acadêmicos encontrados em plataformas de buscas na web, cujo tema envolvia riscos no desenvolvimento de *software*. De 44 artigos com temas próximos ao do estudo, porém, apenas 9 artigos foram de valia para o andamento do projeto. A fim de encontrar os riscos relacionados ao tema proposto, foi feita leitura dessa base de artigos. De maneira geral os artigos continham informação sobre todos os tipos de riscos, assim a leitura e seleção dos riscos foi de extrema importância para a correta triagem das ameaças.

Foi concebida uma lista com 133 riscos, que está anexa a este projeto de graduação. Essa lista contém todos os riscos relacionados a elicitação de requisitos retirados de forma bruta dos artigos, ou seja, sem tratamento. O passo seguinte foi analisar a primeira lista e realizar a triagem, a análise feita baseava-se na leitura de todos os riscos listados nos artigos. Depois da leitura, as ameaças listadas que eram similares ou muito próximas foram agrupadas. Dessa forma, escolhendo apenas um desses riscos ou unindo-os, reduzimos a um único risco. Outro tratamento dado foi o agrupamento dos riscos que eram similares, porém em línguas diferentes, e, da mesma forma explicada anteriormente, escolhia-se um risco para permanecer na lista final.

Essa análise foi de extrema importância pois reduziu de 133 riscos para apenas 15, uma diminuição de mais de 8 vezes. Isso foi veemente importante para atender aos desejos dos desenvolvedores de economia de tempo. O quadro 11 a seguir demonstra a lista dos riscos depois do tratamento realizado.

Quadro 11. Lista final de riscos

Risco
Adoção de novo método/tecnologia
Crescimento de requisitos não previstos
Crítérios de sucesso indefinidos ou mal definidos
Decisões tomadas já foram feitas em arquiteturas anteriores
Dependência de fornecedores externos
Escopo genérico
Especificações de requisitos/objetivos mal definidos ou não identificados
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto.
Fugir do escopo
Gerenciamento inadequado de mudanças
Grande número de stakeholders
Mudança de requisito
Projeto extenso/complexo
Requisitos mal interpretados

Pode-se concluir que o uso da lista foi de fundamental importância, pois ela abrange os principais critérios de riscos envolvidos na elicitação de requisitos em processos que utilizam de metodologia ágil. Uma lista prévia facilita a análise dos riscos envolvidos, porém restringe a análise aos riscos elencados.

## 6.2 Critério de análise de risco

Foram propostos dois métodos de análise dos riscos aos desenvolvedores envolvidos na fase de elicitação de requisitos do processo de desenvolvimento de *software*. Sugeridas também, duas metodologias diferentes de avaliação de risco do processo de desenvolvimento de *software*. Foram analisados ocorrência, probabilidade e impacto dos riscos. O Excel foi a plataforma utilizada para a construção das tabelas e realização dos cálculos.

A análise dos riscos em ambos os métodos se dá pela multiplicação da probabilidade pelo impacto representado na fórmula 1 abaixo, o passo seguinte é classificar os riscos a partir da multiplicação realizada. A categorização é feita do maior para o menor. Isso é feito para descobrir os riscos que geram maior dano ao desenvolvimento do projeto.

$$PxI = P \times I \quad (1)$$

Onde:

P = Probabilidade

I = Impacto

Em reunião com a equipe de projeto do contexto de desenvolvimento, no ambiente de trabalho deles, foram apresentados os dois métodos propostos e a maneira pela qual deveriam ser analisados e valorados. Os desenvolvedores tinham que discutir cada risco, um a um, com enfoque sobre probabilidade, impacto e ocorrência desses riscos. Após esse debate, deveriam chegar em um consenso de valor, sendo esse inserido para o posterior cálculo de probabilidade versus impacto. Eram dois métodos para serem avaliados: o primeiro descrito pela tabela 1 e o outro método pela tabela 3.

O método 1 é uma análise que foi proposta por Presmman (2016) e também por Sommerville (2011), onde a probabilidade varia de 0% a 100%, e o impacto varia de 1 a 4, sendo 1 para impactos insignificantes, 2 para impactos que podem ser tolerados, 3 para impactos graves e 4 para impactos catastróficos.

A tabela 1 representa a análise dos riscos na primeira etapa segundo os valores inseridos pela equipe de projeto do desenvolvimento de *software* exposto acima. O “O” é ocorrência, “P%” é a probabilidade em porcentagem e “I” é impacto.

Tabela 1. Primeira avaliação método 1

<b>Risco Método 1</b>	<b>O</b>	<b>P %</b>	<b>I</b>	<b>PxI</b>
Dependência de fornecedores externos	Sim	78	4	312
Projeto extenso/complexo	Sim	90	2	180
Fugir do escopo	Sim	76	2	152
Requisitos mal interpretados	Sim	53	2	106
Critérios de sucesso indefinidos ou mal definidos	Sim	51	2	102
Especificações de requisitos/objetivos mal definidos ou não identificados	Sim	51	2	102
Crescimento de requisitos não previstos	Sim	97	1	97
Volatilidade nos requisitos	Sim	96	1	96
Mudança de requisito	Sim	95	1	95
Adoção de novo método/tecnologia	Não	46	2	92
Escopo genérico	Sim	92	1	92
Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	55	1	55
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	7	3	21
Gerenciamento inadequado de mudanças	Não	13	1	13
Grande número de stakeholders	Sim	9	1	9

O método 2 de análise realizada foi desenvolvido a fim de facilitar a vida dos desenvolvedores limitando de 100 alternativas de probabilidade para 5. Segundo a equipe de projeto, por meio de *feedback* verbal, esse método facilita o uso da ferramenta e diminui o tempo que os programadores gastam para definir o valor da probabilidade. E para o impacto, houve um acréscimo ínfimo de uma possibilidade a mais de avaliação, isso foi feito para auxiliar o usuário a manter um padrão de escolhas a fim de facilitar o uso da mesma. O valor das probabilidades e dos impactos variam de 1 a 5, a tabela 2 representa a avaliação da probabilidade e do impacto segundo o modelo proposto.



Tabela 2. Avaliação e impacto

Valor da probabilidade/impacto	Avaliação Impacto	Avaliação Probabilidade
1	Muito baixa	Remota
2	Baixa	Baixa
3	Moderada	Média
4	Alta	Alta
5	Muito alta	Muito alta

A tabela 3 apresenta os valores escolhidos pelos desenvolvedores no processo de desenvolvimento proposto para os desenvolvedores. Esses valores foram escolhidos no mesmo período que os valores escolhidos na tabela 1 anterior, ou seja, não houve nenhuma alteração do risco no período, apenas a troca de valores.

Tabela 3. Primeira avaliação método 2

Risco Método 2	O	P	I	PxI
Dependência de fornecedores externos	Sim	4	5	20
Projeto extenso/complexo	Sim	5	3	15
Fugir do escopo	Sim	4	3	12
Adoção de novo método/tecnologia	Não	3	3	9
Critérios de sucesso indefinidos ou mal definidos	Sim	3	3	9
Especificações de requisitos/objetivos mal definidos ou não identificados	Sim	3	3	9
Requisitos mal interpretados	Sim	3	3	9
Crescimento de requisitos não previstos	Sim	5	1	5
Escopo genérico	Sim	5	1	5
Mudança de requisito	Sim	5	1	5
Volatilidade nos requisitos	Sim	5	1	5
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	1	3	3
Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	3	1	3
Gerenciamento inadequado de mudanças	Não	1	1	1
Grande número de stakeholders	Sim	1	1	1

Abaixo temos o comparativo entre a classificação dos riscos dos dois métodos apresentados, lembrando que o método 2 usa valores de 1 a 5 tanto para probabilidade quanto para impacto.

Quadro 12. Comparação de posição de riscos

<b>Método 2</b>	<b>Método 1</b>
Dependência de fornecedores externos	Dependência de fornecedores externos
Projeto extenso/complexo	Projeto extenso/complexo
Fugir do escopo	Fugir do escopo
Adoção de novo método/tecnologia	Requisitos mal interpretados
Critérios de sucesso indefinidos ou mal definidos	Critérios de sucesso indefinidos ou mal definidos
Especificações de requisitos/objetivos mal definidos ou não identificados	Especificações de requisitos/objetivos mal definidos ou não identificados
Requisitos mal interpretados	Crescimento de requisitos não previstos
Crescimento de requisitos não previstos	Volatilidade nos requisitos
Escopo genérico	Mudança de requisito
Mudança de requisito	Adoção de novo método/tecnologia
Volatilidade nos requisitos	Escopo genérico
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Decisões tomadas já foram feitas em arquiteturas anteriores
Decisões tomadas já foram feitas em arquiteturas anteriores	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto
Gerenciamento inadequado de mudanças	Gerenciamento inadequado de mudanças
Grande número de stakeholders	Grande número de stakeholders

Como visto, existe uma variação ínfima das posições na comparação dos métodos. Por isso, foi selecionado o método 2 para a análise das ameaças. Esta decisão foi baseada não só na facilidade do uso da ferramenta pelo usuário, essa confirmada pelos desenvolvedores nas reuniões de *feedback*, mas também visando uma redução de tempo na utilização da ferramenta.

Por fim, a classificação final dos riscos a serem avaliados nessa etapa foi a da tabela 4.

Tabela 4. *Ranking* de riscos

Classificação	Risco
1	Dependência de fornecedores externos
2	Projeto extenso/complexo
3	Fugir do escopo
4	Adoção de novo método/tecnologia
5	Critérios de sucesso indefinidos ou mal definidos
6	Especificações de requisitos/objetivos mal definidos ou não identificados
7	Requisitos mal interpretados
8	Crescimento de requisitos não previstos
9	Escopo genérico
10	Mudança de requisito
11	Volatilidade nos requisitos
12	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto
13	Decisões tomadas já foram feitas em arquiteturas anteriores
14	Gerenciamento inadequado de mudanças
15	Grande número de stakeholders

Pode-se concluir com as análises e *feedback* da equipe de projeto, que ambos os métodos selecionados seriam de grande valia para o uso da ferramenta de análise. Mas com os *feedbacks* verbais ditos nas reuniões para definição do método, viu-se que o segundo era mais efetivo e menos complicado, assim tornando-se ideal para a análise de risco do projeto.

### 6.3 Seleção dos métodos de avaliação de riscos

Para auxiliar a selecionar os riscos que deveriam ser tratados, foram usados dois métodos de avaliação, um gráfico de Pareto e um método de avaliação proposto pelo PMI (2013) onde os riscos são colocados em uma tabela em que eles podem ser mitigados, aceitos ou evitados. Os fatores que influenciam são os de probabilidade e impacto, esse método também é exposto na ABNT/ISO 31010 (2012).

O preenchimento do quadro de avaliação de riscos foi realizado também no Excel. Cada risco nessa etapa foi inserido manualmente na região em que pertencia. Ele dá uma visão geral sobre o que deve ser feito com os riscos. A tabela 5 abaixo demonstra que os dois primeiros riscos devem ser

evitados. Ademais, os riscos de 3 a 7 devem ser mitigados. Os riscos 8,9,10 e 11, apesar de terem uma alta probabilidade de ocorrência, possuem um baixo impacto, assim segundo o modelo ser aceitos pelos desenvolvedores.

O décimo segundo risco deve ser aceito, porque possui médio impacto e probabilidade remota. O item 13 possui uma probabilidade média e um impacto mínimo, assim deve ser aceito porque não causa tanto impacto no andamento do projeto. Por fim, o décimo quarto e o décimo quinto possuem probabilidade remota e impacto muito baixo, por isso também devem ser aceitos, pois seu tratamento geraria um maior tempo gasto e um desperdício de recursos financeiros.

Tabela 5. Quadro avaliação primeira análise

PROBABILIDADE					
Muito alta	8,9,10,11		2		
Alta			3	Evitar	1
Média	13	Aceitar	4,5,6,7		
Baixa				Mitigar	
Remota	14,15		12		
	Muito baixo	Baixo	Médio	Alto	Muito alto
GRAU DE IMPACTO					

Para a realização do tratamento a fim de reduzir o tempo despendido na análise e avaliação das ameaças, escolheu-se um método experimental proposto nesse trabalho para selecionar os riscos que causavam mais impacto pela curva ABC a fim de tratá-los. O Excel foi a ferramenta utilizada para a construção do gráfico. Para gerar o gráfico pegou-se a multiplicação da probabilidade pelo impacto de cada risco e achou-se a porcentagem individual. Depois disso, os riscos foram organizados do maior para o menor. O próximo passo foi fazer a porcentagem acumulada para cada risco. As barras representam a porcentagem individual de cada risco e a curva a porcentagem acumulada dos riscos. O gráfico 1, a seguir, demonstra que apenas 7 riscos correspondem a 80% de todo o impacto. Assim reduz-se quase metade dos riscos necessários a serem tratados.

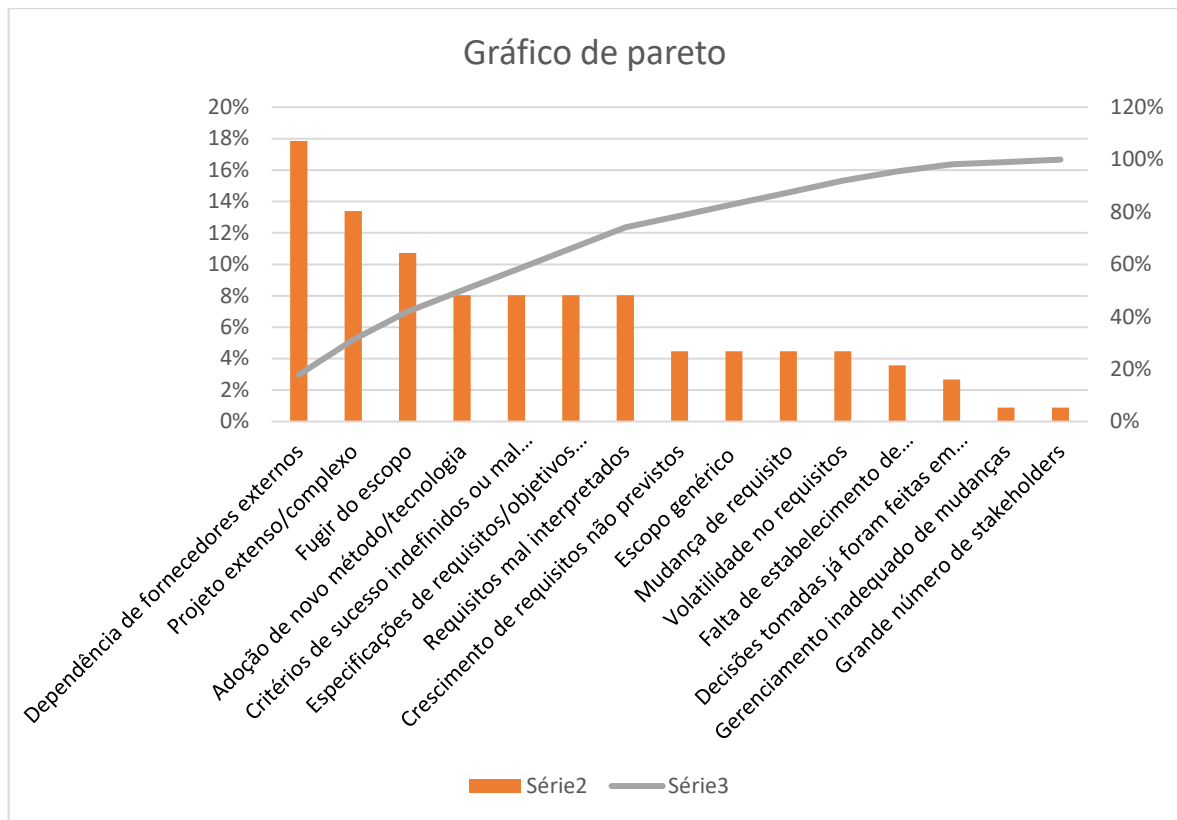


Gráfico 1. Gráfico ABC primeira avaliação

Observa-se que os dois métodos caminharam juntos no caso exposto, todos os riscos que precisavam de tratamento, seja para evitar ou para mitigar, estão expostos em ambos os métodos. O gráfico de Pareto é um método experimental proposto por este projeto a fim de facilitar o uso do usuário pela ferramenta, por evidenciar visualmente a diminuição de riscos a serem tratados.

#### 6.4 Resultado da primeira análise e avaliação dos riscos

O método trouxe como resultado dessa primeira etapa, o auxílio a tomada de decisão do desenvolvedor, porque a decisão final de se tratar ou não as ameaças estão na mão dos integrantes da equipe de projeto. Os sete principais riscos expostos pelos métodos adotados a serem tratados foram:

- a) Dependência de fornecedores externos
- b) Projeto extenso/complexo
- c) Fugir do escopo
- d) Adoção de novo método/tecnologia
- e) Critérios de sucesso indefinidos ou mal definidos
- f) Especificações de requisitos/objetivos mal definidos ou não identificados
- g) Requisitos mal interpretados

Os riscos de “dependência de fornecedores externos” e “projeto extenso” nesse produto são conflitantes devido ao baixo número de desenvolvedores, assim foram propostas duas alternativas para o tratamento desse risco. A primeira foi a de troca de fornecedor externo (o designer do *layout*), já a outra proposta feita foi que os próprios desenvolvedores criassem o design do *layout*.

A última opção deixaria o projeto mais longo, por isso a melhor alternativa foi a troca de fornecedor, haja vista, a ocorrência de atraso devido a dependência de fornecedores externos. O risco de “fugir do escopo” foi amenizado com a reavaliação periódica dos requisitos pela equipe de projeto. “Adoção de novo método/tecnologia” e “critérios de sucesso indefinidos ou mal definidos” foram riscos que não foram tratados.

“Especificações de requisitos/objetivos mal definidos ou não identificados” e “requisitos mal interpretados” foram amenizados juntamente com a revisão periódica e uma maior comunicação entre a equipe de projeto e os *Project Owner e Manager*. Foram criados meios menos formais de comunicação entre as partes a fim de facilitar a comunicação.

Para esses projetos que utilizam de metodologia ágil, as mudanças nos requisitos não são tão impactantes quanto nos projetos que não a utilizam, uma vez que a metodologia ágil abraça as mudanças de escopo de projeto. Por isso, o tipo de metodologia utilizada no desenvolvimento de *software* influencia muito na análise e avaliação de risco.

## 7 DESENVOLVIMENTO DA FERRAMENTA DE AVALIAÇÃO DE RISCOS

Esta parte do projeto tem por objetivo testar a ferramenta em um contexto de *software*. A ferramenta foi desenvolvida no programa Excel do pacote da *Microsoft office*, ela é uma planilha que possui os 15 riscos de elicitación de requisitos do processo de desenvolvimento de *software* já predefinidos.

O usuário deve inserir a probabilidade, o impacto dos riscos e se os riscos já ocorreram na fase atual do projeto ou não. Como resultado, a ferramenta retorna uma curva ABC simples e uma curva ABC (sugestão) onde irá descrever quais são os riscos A, B ou C. Outro resultado advindo dos dados inseridos é o quadro que mostrará se o risco deve ser mitigado, aceito ou evitado. Também mostrará o quadro sugestão onde indicará a criticidade do risco.

Para desenvolver a ferramenta foi necessário programar em linguagem de programação *visual basic*, a fim de torná-la automatizada. A ferramenta foi desenvolvida em conjunto com um estudante de pós-graduação. A ideia do que deveria ser feito foi cedida pelo autor ao pós-graduando, para que ele ficasse responsável por escrever os códigos em *visual basic*.

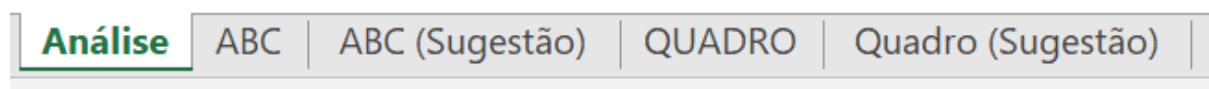


Figura 4. Funções da ferramenta

A ferramenta possui 5 abas como mostra a figura 4 acima, onde temos a curva ABC, a curva ABC com sugestão, o quadro e, por fim, o quadro com as sugestões de criticidade dos riscos. O usuário deve navegar entre as 4 últimas janelas para coletar o resultado dos dados inseridos na aba de análise.

### 7.1 Tela de análise da ferramenta

A figura 5 abaixo demonstra a tela inicial, a tela de análise, na qual o desenvolvedor do *software* que almeja fazer a análise e avaliação de risco vai interagir. Essa é a única parte na qual o usuário terá a interação de inserir dados na planilha. Nesse momento, o usuário insere os dados nas colunas de Ocorrência, Probabilidade e Impacto, e para gerar o resultado dos valores inseridos deve-se apertar atualizar. Dessa maneira mudará todas as partes da ferramenta, que irá gerar novas classificações, gráficos e quadros.

Método 1		O	P	I	PxI
1	Especificações de requisitos/objetivos mal definidos ou não identificados	Não	4	5	20
2	Dependência de fornecedores externos	Sim	5	3	15
3	Crítérios de sucesso indefinidos ou mal definidos	Não	4	3	12
4	Requisitos mal interpretados	Sim	3	3	9
5	Projeto extenso/complexo	Sim	3	3	9
6	Escopo genérico	Sim	3	3	9
7	Crescimento de requisitos não previstos	Sim	3	3	9
8	Adoção de novo método/tecnologia	Não	5	1	5
9	Mudança de requisito	Sim	5	1	5
10	Fugir do escopo	Sim	5	1	5
11	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	5	1	5
12	Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	1	3	3
13	Grande número de stakeholders	Sim	3	1	3
14	Volatilidade no requisitos	Sim	1	1	1
15	Gerenciamento inadequado de mudanças	Não	1	1	1

Atualizar

Figura 5. Tela inicial da ferramenta Risky\_TOOL\_2000\_V1

A primeira coluna é a classificação do risco na análise realizada. As classificações mudam com os valores da probabilidade e do impacto inseridos pelo usuário, ou seja, a cada rodada de análise as classificações podem ser diferentes. Tudo depende do valor inserido para a probabilidade e para o impacto. A ordem dos riscos decresce dos que geram mais dano ao projeto. Isso é calculado a partir da multiplicação da probabilidade pelo impacto, que gera um valor. Valor esse exposto na coluna PxI.

A coluna 2 é o risco a ser analisado, ela indica o que deve ser investigado pelo desenvolvedor. Têm-se no total 15 riscos a serem analisados, todos relacionados a elicitação de requisitos. Essa lista não pode ser mudada, ou seja, não se pode acrescentar novos riscos, apenas analisar os riscos envolvidos. Outro aspecto importante é que todos os riscos devem ser analisados e valorados, caso isso não ocorra a ferramenta não consegue gerar os resultados.

A coluna 3 com o símbolo “O” em cima é para analisar se já ocorreu aquele risco no processo de desenvolvimento até o momento da análise realizada. As duas únicas opções para marcar são “Sim” e “Não”. Esse diagnóstico serve para mostrar ao usuário se a avaliação de riscos está sendo efetiva até o momento, é um indicador de como anda o processo de avaliação dos riscos.

A quarta coluna é uma das duas opções que o usuário terá para analisar o risco, podendo inserir valores de 1 a 5 a depender da probabilidade de ocorrência do risco. Para inserir esses valores, os integrantes da equipe que estão analisando os riscos devem debater entre si e chegar a um consenso de qual é o valor que mais representa a probabilidade do risco. A coluna seguinte é a do impacto que também varia de 1 a 5, essa coluna segue o mesmo padrão da coluna anterior, isto é, os desenvolvedores precisam chegar a um acordo para a inserção dos dados.

As cores dos valores variam conforme a intensidade do número: 5 por ser um número alto apresenta um vermelho intenso, o número 1 a cor amarela predominante, já nos entremeios tem-se variações de laranja. A classificação por cor foi realizada para facilitar a visualização do usuário. O PxI é a coluna do cálculo da probabilidade vezes o impacto, é nessa coluna que a planilha puxa os valores para calcular as curvas ABC.



## 7.2 Tela da curva ABC

A figura 6 é uma captura de tela que mostra a tabela do gráfico de Pareto, a primeira coluna evidencia os riscos. A segunda coluna é a multiplicação da probabilidade pelo impacto, para gerar o gráfico, os valores dessa multiplicação são classificados do maior para o menor. A próxima coluna demonstra o percentual que cada multiplicação tem no risco total do projeto. A segunda é o percentual acumulado que serve como base para se achar os riscos que possuem 80% do risco total.

A curva ABC se mostrou como uma alternativa de avaliação de risco, pois foram testadas 33 combinações possíveis para auxiliar no tratamento do risco e, em todos eles, os riscos deveriam ser mitigados ou evitados. A priori, o gráfico de Pareto era só um auxílio visual para descobrir quais os riscos mais importantes, porém mostrou ser uma ótima ferramenta para a avaliação de riscos a fim de serem tratados.

<b>Método 1</b>	<b>Pxl</b>	<b>%</b>	<b>Acumu</b>
Especificações de requisitos/objetivos mal definidos ou não identificadas	20	18%	18%
Dependência de fornecedores externos	15	14%	32%
Critérios de sucesso indefinidos ou mal definidos	12	11%	42%
Requisitos mal interpretados	9	8%	50%
Projeto extenso/complexo	9	8%	59%
Escopo genérico	9	8%	67%
Crescimento de requisitos não previstos	9	8%	75%
Adoção de novo método/tecnologia	5	5%	79%
Mudança de requisito	5	5%	84%
Fugir do escopo	5	5%	88%
Falta de estabelecimento de processos/procedimentos/metodologia/plano	5	5%	93%
Decisões tomadas já foram feitas em arquiteturas anteriores	3	3%	95%
Grande número de stakeholders	3	3%	98%
Volatilidade no requisitos	1	1%	99%
Gerenciamento inadequado de mudanças	1	1%	100%
<b>SOMA</b>	<b>111</b>		

Figura 6. Cálculo ABC

Abaixo têm-se o gráfico 2 da curva ABC, que é uma representação gráfica da tabela acima. A tabela mostra na primeira coluna os riscos e na segunda coluna o valor a ser analisado. Esse valor é a multiplicação da probabilidade pelo impacto, a seguir soma todos os valores do risco e coloca na parte da soma. Para chegar ao resultado da terceira deve-se pegar o resultado risco e dividir pela soma, mostrando a porcentagem que cada risco corresponde no total. A última coluna é a porcentagem acumulada, para se chegar a isso deve-se somar as porcentagens dos riscos de forma iterada. O último risco sempre deve ser de 100%, pois é uma soma acumulada.

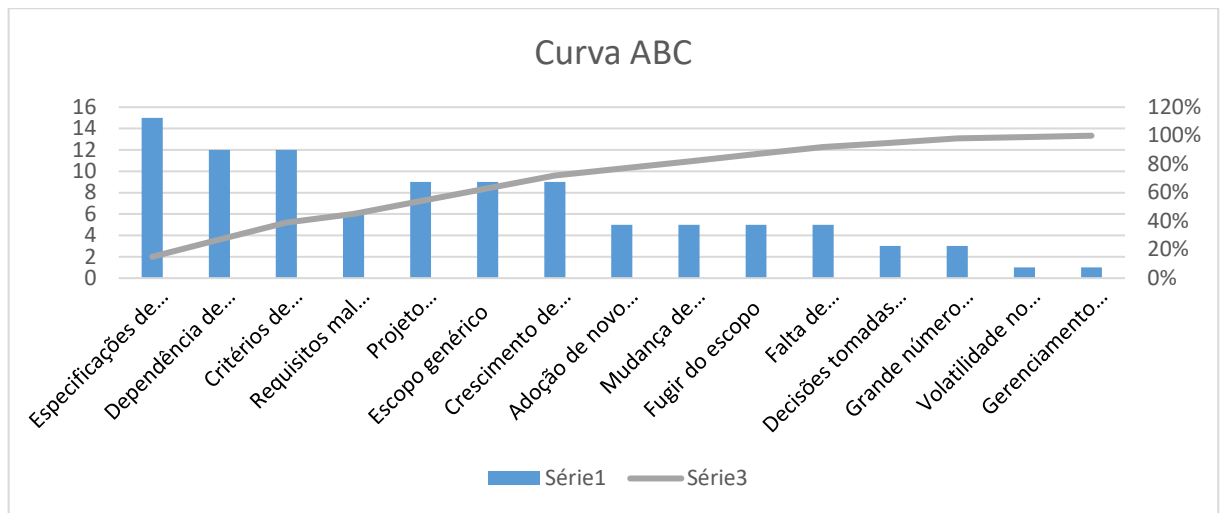


Gráfico 2. Curva ABC da ferramenta

Para a construção da curva ABC acima foi utilizada a tabela anterior, esse gráfico é gerado automaticamente por uma função do próprio Excel. As barras do gráfico representam os valores individuais dos riscos em porcentagem. O gráfico em curva representa os valores acumulados ponto a ponto dos riscos.

A ferramenta foi nomeada de Risky\_tool\_2000\_v1, para evidenciar sua funcionalidade. Traz como sugestão, para facilitar seu uso, indicações dos riscos que devem ser tratados. O gráfico abaixo demonstra isso. Os riscos a serem tratados são os do tipo “A”, pois estes representam aproximadamente 80% de todos os riscos. O gráfico 3 representa como será apresentado ao usuário a ferramenta.

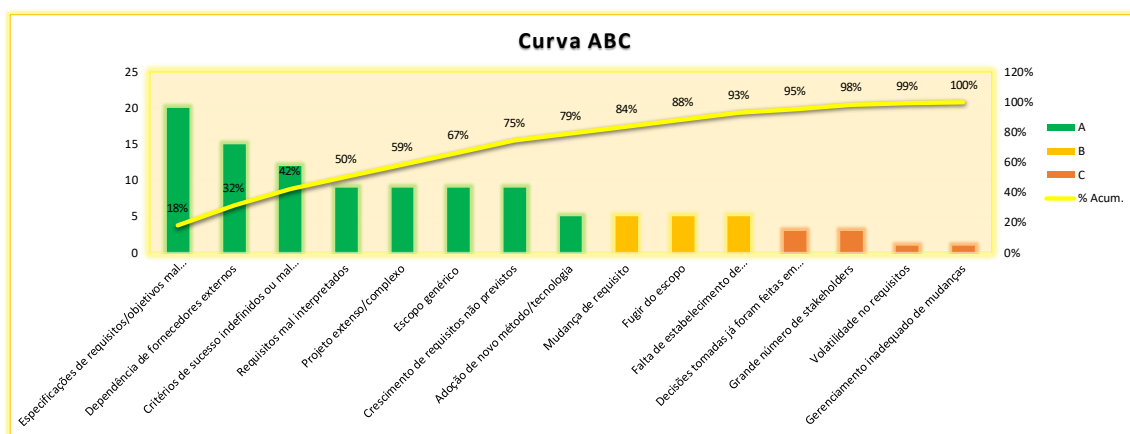


Gráfico 3. Curva ABC da ferramenta colorida

A cor de cada risco serve como efeito visual, sendo a cor verde para os riscos tipo A, laranja para os riscos tipo B, e vermelho para os riscos tipo C. Para aplicar as cores em cada tipo de risco é utilizado uma ferramenta embutida no próprio gráfico.

A aplicação das cores se deu usando os valores para os riscos tipo A, que são os riscos mais relevantes, que devem ter atenção maior pelo valor alto que possuem. Corresponde a 20% dos riscos que totalizam 80% do impacto. B são os riscos que demandam atenção intermediária, mas que também possuem um valor mais alto. Esses riscos representam 15% do total de riscos e correspondem a 30% do impacto. Já C, são os riscos de prioridade reduzida, porque o impacto deles é pequeno. Representam 50% e são relativos a 5% do valor.

Por meio de *feedback* verbal dos desenvolvedores que utilizaram a ferramenta, eles disseram que essa curva ABC com sugestão foi muito útil, porque mostrou de forma visual os riscos que causavam mais impacto e que deveriam ser tratados.

### **7.3 Tela do quadro de avaliação**

Essa é a tela mais importante da ferramenta, a tabela 7 abaixo demonstra como o usuário verá as informações dispostas. É nessa parte que a ferramenta auxilia o usuário como se comportar perante o risco. Essa é parte da ferramenta onde, de fato, existe a avaliação do risco para o bom andamento do projeto.

A tabela 7 mostra os pares, gerados a partir da tabela 8, responsáveis pela classificação dos riscos como evitados, mitigados ou aceitos. Essas são todas as possíveis alternativas para a classificação. O primeiro número, dentro dos parênteses, é o valor da probabilidade inserida pelo usuário e o segundo é o valor do impacto escolhido pelo mesmo. A programação VBA, utilizada para criação da ferramenta, é responsável por anexar cada risco em seu respectivo par, a partir dos valores inseridos pela equipe de desenvolvimento. Dessa forma, é criado o quadro visto na tabela 8.

Tabela 7. Pares de avaliação

Avaliação	Pares (probabilidade; impacto)
Evitar	(5;4)
	(5;5)
	(5;6)
	(4;4)
	(4;5)
	(3;5)
Mitigar	(5;2)
	(4;2)
	(4;3)
	(4;4)
	(3;4)
	(2;4)
	(1;4)
	(2;5)
Aceitar	(5;1)
	(4;1)
	(3;1)
	(2;1)
	(1;1)
	(3;2)
	(2;2)
	(1;2)
	(2;3)
	(1;3)

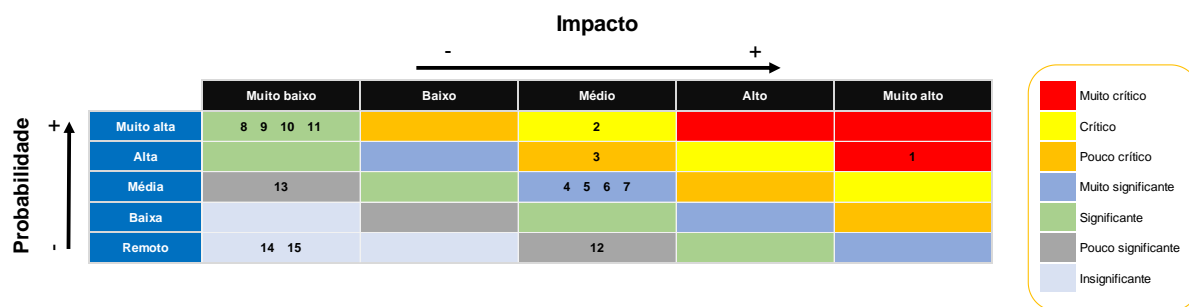
Tabela 8. Quadro avaliação da ferramenta

PROBABILIDADE						
Muito alta	8 9 10 11		2			
Alta			3			1
Média	13		4 5 6 7			
Baixa						
Remota	14 15		12			
	Muito baixo	Baixo	Médio	Alto	Muito alto	
GRAU DE IMPACTO						

A parte vermelha da tabela 8 são os riscos que devem ser evitados, existem 6 resultados possíveis dos riscos a serem evitados. Já na parte laranja, existem 9 resultados possíveis, os riscos que estão nela inseridos geralmente são ameaças nocivas e devem ser mitigadas, esses riscos em sua maioria causam pouco menos impacto do que os riscos da parte vermelha. A parte que está em verde possui 10 resultados possíveis, são os riscos que devem ser aceitos. Esses riscos tendem a não influenciar tanto no desenvolvimento do projeto, assim tratá-los geralmente é desperdício de recurso.

Já o quadro sugestão representado na tabela 9, demonstra os níveis de criticidade dos riscos. Ele usa a mesma métrica da tabela 8 acima. Sua diferença está na facilidade visual do usuário perceber os riscos por sua maior variação de cores. Ele também traz mais níveis de análises para o desenvolvedor.

Tabela 9. Níveis de criticidade



Após realizada a primeira análise e avaliação do contexto de desenvolvimento, os desenvolvedores utilizaram-se da ferramenta para realizar uma segunda avaliação. Isso foi feito com o intuito de testar o uso da ferramenta e de auxiliar os programadores no tratamento dos riscos.

### 7.3.1 Análise dos riscos utilizando a ferramenta

Nessa etapa o desenvolvedor inseriu na tela inicial os valores para os riscos. O usuário teve que valorar dois aspectos, sendo eles probabilidade e impacto, e dizer se o risco já tinha ocorrido desde a última análise ocorrida.

Tabela 10. Segunda análise

Análise 2		O	P	I	PxI
1	Especificações de requisitos/objetivos mal definidos ou não identificados	Não	4	3	12
2	Dependência de fornecedores externos	Sim	4	3	12
3	Critérios de sucesso indefinidos ou mal definidos	Não	3	3	9
4	Requisitos mal interpretados	Não	3	3	9
5	Projeto extenso/complexo	Sim	3	3	9
6	Escopo genérico	Sim	3	3	9
7	Adoção de novo método/tecnologia	Não	5	1	5
8	Fugir do escopo	Sim	5	1	5
9	Crescimento de requisitos não previstos	Sim	4	1	4
10	Mudança de requisito	Sim	4	1	4
11	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	1	3	3
12	Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	3	1	3
13	Grande número de stakeholders	Sim	1	1	1
14	Volatilidade no requisitos	Sim	1	1	1
15	Gerenciamento inadequado de mudanças	Não	1	1	1

A tabela 10 acima representa a segunda análise do estudo, pode-se observar que existe uma certa homogeneidade nos valores das probabilidades e impacto. Também demonstra que se trata de uma metodologia ágil no desenvolvimento de projeto, pois alguns requisitos como: fuga de escopo, mudança de requisitos e crescimento de requisitos não previstos não possuem valor de PxI extremos. Os riscos citados anteriormente não possuem alto impacto, pois essas ameaças já eram previstas desde o começo do desenvolvimento do *software*, devido a utilização do método *scrum* que elícita os requisitos ao longo do projeto, podendo ocorrer mudanças e crescimento não esperado dos requisitos e fuga do escopo.

Ademais, pode-se observar a alta ocorrência dos riscos - sendo que exatos sessenta por cento deles aconteceram. Porém, a maioria dos riscos que acometeram o projeto são as ameaças listadas na parte inferior da tabela 10, que são as que impactam de forma menos grave o andamento do projeto.

### 7.3.2 Comparativo da primeira análise com a segunda

Na tabela 11, onde temos o comparativo entre a primeira e a segunda análise, é possível ver as trocas de posição da primeira análise para a segunda e as trocas de valores das probabilidades e impactos de cada risco. A primeira parte foi a análise 1, que não foi realizada com a ferramenta. Porém, para facilitar a compreensão, a visualização e a comparação foram inseridos os dados da primeira análise na ferramenta. Como a ferramenta seguiu os moldes escolhidos da análise anterior, não ocorreu discrepância nos valores. Na análise 2 da tabela 11 foi utilizada a ferramenta.

Vale ressaltar que os métodos utilizados na análise 1 são os mesmos utilizados na análise 2, dessa forma não ocorreu nenhuma diferença de avaliação metodológica. A diferença consiste no lapso temporal de aproximadamente 1 mês entre as análises, afetando assim os valores de probabilidade e impacto de cada risco. A segunda análise foi realizada após ter ocorrido uma avaliação de risco anterior. O fundamento para a consideração da ocorrência dos riscos é que eles devem ocorrer durante o período analisado.

Tabela 11. Comparativo das análises

Análise 1		O	P	I	Análise 2		O	P	I
1	Especificações de requisitos/objetivos mal definidos ou não identificados	Não	5	3	1	Especificações de requisitos/objetivos mal definidos ou não identificados	Não	4	3
2	Dependência de fornecedores externos	Sim	4	3	2	Dependência de fornecedores externos	Sim	4	3
3	Crêterios de sucesso indefinidos ou mal definidos	Não	3	3	3	Crêterios de sucesso indefinidos ou mal definidos	Não	3	3
4	Requisitos mal interpretados	Sim	3	3	4	Requisitos mal interpretados	Não	3	3
5	Projeto extenso/complexo	Sim	3	3	5	Projeto extenso/complexo	Sim	3	3
6	Escopo genérico	Sim	3	3	6	Escopo genérico	Sim	3	3
7	Crescimento de requisitos não previstos	Sim	5	1	7	Adoção de novo método/tecnologia	Não	5	1
8	Adoção de novo método/tecnologia	Não	5	1	8	Fugir do escopo	Sim	5	1
9	Mudança de requisito	Sim	5	1	9	Crescimento de requisitos não previstos	Sim	4	1
10	Fugir do escopo	Sim	5	1	10	Mudança de requisito	Sim	4	1
11	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	1	3	11	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	1	3
12	Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	3	1	12	Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	3	1
13	Grande número de stakeholders	Sim	1	1	13	Grande número de stakeholders	Sim	1	1
14	Volatilidade no requisitos	Sim	1	1	14	Volatilidade no requisitos	Sim	1	1
15	Gerenciamento inadequado de mudanças	Não	1	1	15	Gerenciamento inadequado de mudanças	Não	1	1

### 7.3.3 Comparativo da classificação

O quadro 13 é o comparativo entre as posições das análises realizadas. É perceptível que não ocorreu a troca de posições entre os riscos nas primeiras e nas últimas colocações. Acredita-se que essa alteração do ranking ocorreu pois houve o tratamento dos principais riscos previamente. Do sétimo risco ao décimo, houve troca de posições, notou-se que esses riscos estão relacionados a mudança de requisitos e crescimento de requisitos não previstos. Isso pode ter advindo do tratamento dado aos riscos pela equipe, a facilitação da troca de informação entre as partes envolvidas

Quadro 13. Comparação de produção entre análises

Análise 1	Análise 2
Especificações de requisitos/objetivos mal definidos ou não identificados	Especificações de requisitos/objetivos mal definidos ou não identificados
Dependência de fornecedores externos	Dependência de fornecedores externos
Critérios de sucesso indefinidos ou mal definidos	Critérios de sucesso indefinidos ou mal definidos
Requisitos mal interpretados	Requisitos mal interpretados
Projeto extenso/complexo	Projeto extenso/complexo
Escopo genérico	Escopo genérico
Crescimento de requisitos não previstos	Adoção de novo método/tecnologia
Adoção de novo método/tecnologia	Fugir do escopo
Mudança de requisito	Crescimento de requisitos não previstos
Fugir do escopo	Mudança de requisito
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto
Decisões tomadas já foram feitas em arquiteturas anteriores	Decisões tomadas já foram feitas em arquiteturas anteriores
Grande número de stakeholders	Grande número de stakeholders
Volatilidade nos requisitos	Volatilidade nos requisitos
Gerenciamento inadequado de mudanças	Gerenciamento inadequado de mudanças

Pode-se concluir que a análise dos riscos gera uma troca de posições da classificação de cada risco: riscos que geravam mais danos na análise anterior reduzem seu potencial ofensivo na análise subsequente e vice e versa.

### 7.3.4 Comparativo da ocorrência

A tabela 12 é comparativa entre as ocorrências dos riscos nas duas análises. É possível ver na tabela que a maioria dos riscos ocorreram nas duas etapas da análise do *software*. Apenas o risco de “requisito mal interpretado” não ocorreu depois de ser tratado, isso foi devido ao tratamento prévio realizado pelos desenvolvedores. Esse se deu com a melhora da comunicação entre os *stakeholders*. O



risco de “dependência dos fornecedores externos” reincidiu. A equipe optou pela troca de fornecedor e não por realizar o processo de *layout*, mantendo-se assim, a dependência.

De certa forma, o tratamento dos riscos na análise anterior de pouco interferiu na nova análise, porque se deu de forma tardia. Contudo ainda houve grande impacto na redução da ocorrência dos riscos: a não incidência de um risco classificado na quarta posição altera bastante o curso do desenvolvimento do *software*.

Tabela 12. Comparativo de ocorrências

Riscos		O1	O2
1	Especificações de requisitos/objetivos mal definidos ou não identificados	Não	Não
2	Dependência de fornecedores externos	Sim	Sim
3	Critérios de sucesso indefinidos ou mal definidos	Não	Não
4	Requisitos mal interpretados	Sim	Não
5	Projeto extenso/complexo	Sim	Sim
6	Escopo genérico	Sim	Sim
7	Crescimento de requisitos não previstos	Sim	Sim
8	Adoção de novo método/tecnologia	Não	Não
9	Mudança de requisito	Sim	Sim
10	Fugir do escopo	Sim	Sim
11	Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	Não	Não
12	Decisões tomadas já foram feitas em arquiteturas anteriores	Sim	Sim
13	Grande número de stakeholders	Sim	Sim
14	Volatilidade no requisitos	Sim	Sim
15	Gerenciamento inadequado de mudanças	Não	Não

Como conclusão, vemos que existe diferença mínima nas ocorrências dos riscos. Mas, mesmo a mínima redução na incidência de riscos é de grande valia para o andamento do projeto. O resultado na redução de riscos pós-tratamento foi abaixo do esperado devido as condições de implementação tardias no desenvolvimento do *software*, no entanto, o valor agregado não foi insignificante. Temos, então, que em condições favoráveis o emprego da ferramenta pode impactar de forma significativa na redução de riscos.

### 7.3.5 Comparativo das probabilidades

A tabela 13 representa os comparativos de probabilidade entre as duas análises realizadas neste estudo. A tabela nos mostra que ocorreu alteração de valor na “especificação de requisitos mal definidos ou não identificados”, “mudança de requisito” e “fuga do escopo”. Como pode ser visto, o tratamento realizado na primeira análise interferiu nas novas probabilidades, isso demonstra uma alteração benigna decorrida a partir do uso da avaliação de risco.

Tabela 13. Comparativo de probabilidade

	P1	P2
1 Especificações de requisitos/objetivos mal definidos ou não identificados	5	4
2 Dependência de fornecedores externos	4	4
3 Critérios de sucesso indefinidos ou mal definidos	3	3
4 Requisitos mal interpretados	3	3
5 Projeto extenso/complexo	3	3
6 Escopo genérico	3	3
7 Crescimento de requisitos não previstos	5	5
8 Adoção de novo método/tecnologia	5	5
9 Mudança de requisito	5	4
10 Fugir do escopo	5	4
11 Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	1	3
12 Decisões tomadas já foram feitas em arquiteturas anteriores	3	1
13 Grande número de stakeholders	1	1
14 Volatilidade no requisitos	1	1
15 Gerenciamento inadequado de mudanças	1	1

Conclui-se da tabela 13 que houve diminuição na probabilidade de incidência dos riscos, sugerindo funcionamento adequado dos métodos da ferramenta. Além disso, a soma total das segundas probabilidades também foi menor.

### 7.3.6 Comparativo do impacto

O comparativo do impacto exposto na tabela 14, evidencia que não ocorreu nenhuma alteração dos impactos de uma análise para a outra. Mesmo estando em diferentes etapas de análise, os desenvolvedores resolveram não alterar o impacto que os riscos causavam no desenvolvimento do projeto. Isso pode ter ocorrido porque ambas as análises ocorreram já no fim do processo de desenvolvimento de *software*.

Tabela 14. Comparativo de impacto

	I1	I2
Especificações de requisitos/objetivos mal definidos ou não identificados	3	3
Dependência de fornecedores externos	3	3
Critérios de sucesso indefinidos ou mal definidos	3	3
Requisitos mal interpretados	3	3
Projeto extenso/complexo	3	3
Escopo genérico	3	3
Crescimento de requisitos não previstos	1	1
Adoção de novo método/tecnologia	1	1
Mudança de requisito	1	1
Fugir do escopo	1	1
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto	3	3
Decisões tomadas já foram feitas em arquiteturas anteriores	1	1
Grande número de stakeholders	1	1
Volatilidade no requisitos	1	1
Gerenciamento inadequado de mudanças	1	1

A conclusão que podemos chegar é que o impacto, nesse projeto, independe da avaliação anterior dos riscos, ou seja, as principais áreas de atuação da equipe seriam na modificação da probabilidade pelo tratamento dos riscos, a fim de diminuir a ocorrência dos mesmos. Mesmo mantendo o impacto, conseguindo diminuir a probabilidade, podemos ter benefícios ao transformar riscos que devem ser evitados, ou mitigados, para aceitos.

### 7.3.7 Avaliação dos riscos por meio da curva ABC

O gráfico de Pareto, representado pelo gráfico 4, apresentou visualmente 6 riscos a serem avaliados, que começam nas “especificações de requisitos mal definidos” e vão até “escopo genérico”. Esses riscos a serem avaliados tem o impacto de aproximadamente 70% do projeto, o que não é fiel a regra 80/20 perpetuada por Pareto. Com essa avaliação, há redução do número de requisitos a serem tratados em aproximadamente 60%, acarretando invariavelmente na redução do tempo despendido na avaliação e tratamento de riscos.

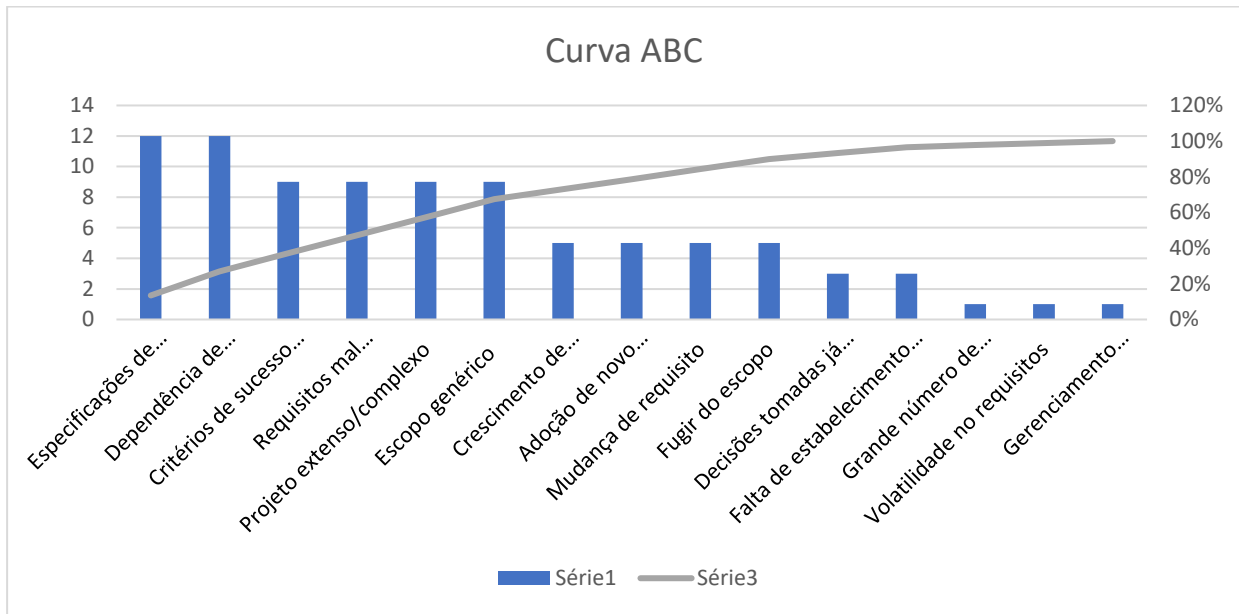


Gráfico 4. Curva ABC segunda análise

A curva ABC sugestão, mostrada pelo gráfico 5, evidencia visualmente a queda dos valores dos riscos entre o “escopo genérico” e o “crescimentos não previstos de requisitos”.

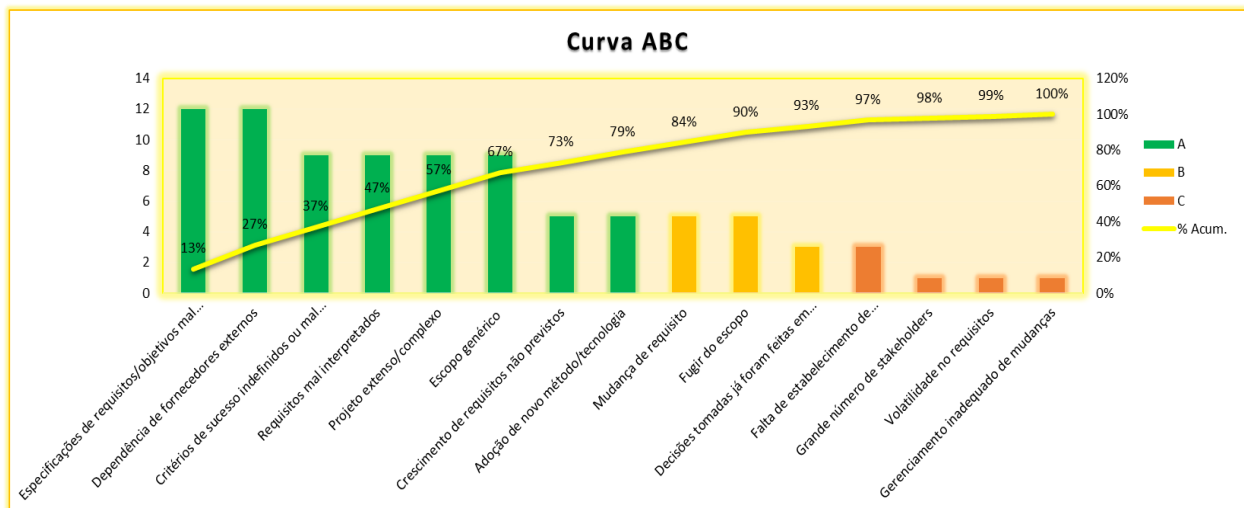


Gráfico 5. Curva ABC sugestão

Em conclusão, a queda citada anteriormente demonstra que o uso visual da ferramenta não é sempre preciso, uma vez que a curva ABC sugestão possui padrão de cores que advém da regra de Pareto

– que tem constância no valor das porcentagens. Mas, com interpretação adequada do operador da ferramenta, o método torna-se fiel aliado na avaliação das ameaças.

### 7.3.8 Avaliação pelo quadro de avaliação

A tabela 15 mostrou que não existe mais riscos a serem evitados, ou seja, riscos na parte superior direita do quadro. A parte central, que abrange os riscos 1 a 6, conforme o método proposto, devem ser mitigados. Já os riscos de 7 a 15, que se encontram nas partes periféricas à esquerda, devem ser aceitos pela equipe de projeto. Percebe-se que os riscos que devem ser mitigados, do 1 ao 6, estão muito próximos da região de aceitação do risco, o que permite inferir que o resultado obtido pela avaliação de risco trouxe benefícios ao desenvolvedor.

PROBABILIDADE					
Muito alta	7 8 9 10				
Alta			1 2		
Média	11		3 4 5 6		
Baixa					
Remota	13 14 15		12		
	Muito baixo	Baixo	Médio	Alto	Muito alto
	GRAU DE IMPACTO				

Tabela 15. Quadro segunda análise

A tabela 16 mostra as sugestões da ferramenta para a significância do risco no desenvolvimento do projeto. Os riscos estão divididos por cores especificadas na legenda como “muito crítico”, “crítico”, “pouco crítico” etc. Dessa forma, além dos riscos serem classificados por probabilidade, impacto e significância, há também auxílio visual da ferramenta para estratificar o risco e evitar confusões de leitura.

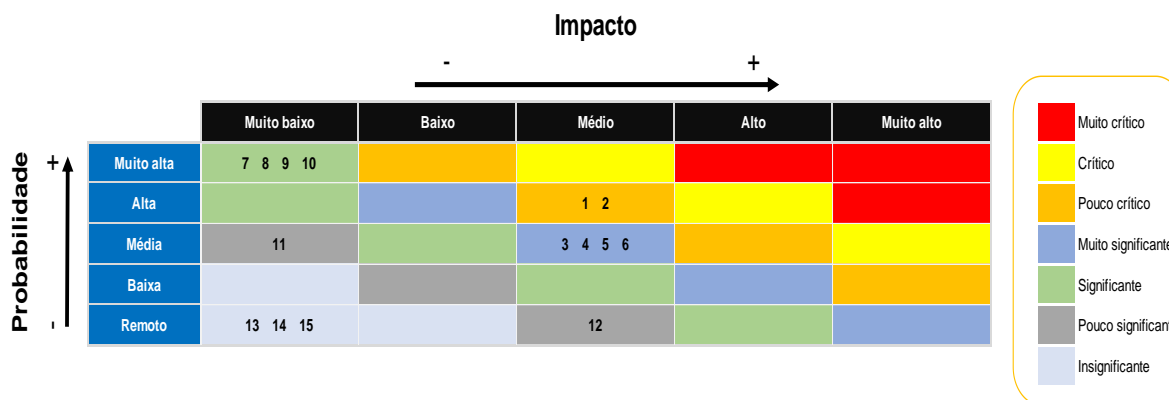


Tabela 16. Nível de criticidade segunda análise

Podemos concluir, durante a construção desse projeto de graduação, que a ferramenta de avaliação de riscos é de grande auxílio para pequenos desenvolvedores que utilizam de metodologia ágil. Os motivos que levam a essa conclusão iniciam-se no aumento do entendimento da equipe de projeto sobre os riscos envolvidos na elicitación de requisitos, para assim tratá-los e reduzir seus efeitos negativos. Nesse projeto, vemos, nos resultados expressos, que houve diminuição dos riscos que impactavam e atrasavam o desenvolvimento do *software* e redução do retrabalho tido pelos desenvolvedores.

## 8 CONSIDERAÇÕES FINAIS

Foram inseridos os conceitos básicos sobre *software*, elicitación de requisitos, riscos, engenharia de requisitos e metodologia ágil. Com o intuito de elucidar o leitor sobre aspectos relevantes para a compreensão deste projeto de conclusão de curso, o estudo trouxe diversas técnicas de análise e avaliação de risco segundo autores renomados e segundo normas da organização internacional de normalização - muito importantes para o bom desenvolvimento da ferramenta desenhada no projeto.

Esse estudo trouxe duas técnicas principais de como analisar e avaliar riscos envolvidos na elicitación de requisitos no processo de desenvolvimento de *software*. Como resultado criamos uma ferramenta para auxiliar os desenvolvedores de *software* a analisarem e avaliarem riscos de maneira mais rápida e fácil, beneficiando principalmente empresas pequenas. Esse tema tem grande importância na redução de custos e de retrabalho realizado pelas equipes de desenvolvimento desses tipos de projetos, haja vista que a maioria das pessoas envolvidas no desenvolvimento de *software* tendem a negligenciar a análise, avaliação e tratamento de riscos.

A etapa do contexto de desenvolvimento exprime uma visão aprofundada sobre a empresa, seu cliente e o *software* desenvolvido. Esse período é demasiadamente significativo para compreender o contexto no qual a ferramenta será empregada. Dessa forma, ocorre o entendimento das dimensões dos riscos, das probabilidades e principalmente do impacto que esses acarretam no desenvolvimento do *software* e de como a ferramenta auxilia na avaliação desses riscos.

O resultado na redução de riscos pós-tratamento foi abaixo do esperado por ser implementado tardiamente, a apenas três quartos do final do desenvolvimento do produto. Ainda assim, houve valor agregado ao projeto. Usando a ferramenta em condições favoráveis e desde o início do desenvolvimento do *software*, seu uso pode impactar de forma significativa na diminuição de riscos: redução de custos no retrabalho e na agilização das entregas dos sprints. Os desenvolvedores da equipe de projeto aceitaram a ferramenta proposta e perceberam que a utilização trouxe benefícios para o andamento do projeto em questão.

Dessa maneira, o objetivo principal desse estudo, qual seja, desenvolver uma ferramenta de avaliação dos riscos envolvidos na elicitación de requisitos de um contexto de projeto de desenvolvimento de *software*, foi alcançado. Geramos uma ferramenta editada no Microsoft Excel, que foi utilizada pelos desenvolvedores do contexto estudado, e também documentamos todos os processos de desenvolvimento de *software* em geral e do referido *software* no corpo deste trabalho.

Como limitações de pesquisa, destacam-se a utilização tardia da ferramenta na análise de avaliação dos riscos e o estudo do contexto de uma única equipe de desenvolvimento de *software*. Outras limitações a serem observadas foram o tamanho da empresa analisada e os tamanhos dos projetos

realizados por essa empresa, que eram de pequeno porte, ou seja, a gravidade dos riscos e os riscos eram de menor impacto para o andamento do projeto.

Sugere-se, então, para futuras linhas de pesquisa, implementar a ferramenta em equipes de desenvolvimento de *software* de diversos tamanhos. Outra sugestão é a aplicação da ferramenta a empresas maiores com projetos mais robustos. Também se sugere a criação de modelos de tratamento de *software* para auxiliar e economizar tempo dos desenvolvedores. Ademais, adicionar mais etapas de avaliação dos riscos no desenvolvimento de *software*, a fim de realizar-se uma avaliação mais holística do processo de desenvolvimento de *software*.

Depreende-se que o emprego da ferramenta de avaliação de riscos é de grande ajuda para pequenas empresas desenvolvedoras de *software* que utilizam de método ágil e que geralmente não possuem técnicas de avaliação de riscos pré-definidas. Isso se dá por diversos motivos, sendo o maior desses o acréscimo da compreensão do gestor de projeto sobre os riscos relacionados a elicitação de requisitos. Isso pode ser validado nos resultados expressos por esse projeto de graduação, que reduziu os riscos que impactavam o processo de desenvolvimento de *software*, os quais retardavam o desenvolvimento do projeto, e também diminuiu o retrabalho tido pelos desenvolvedores.

## 9 REFERÊNCIAS

- ABNT- Associação Brasileira de Normas Técnicas. NBR ABNT/ISO 31000:2009 Gestão de riscos — Princípios e diretrizes. Rio de Janeiro, ABNT, 2009.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ABNT/ISO 9001/2000: Sistemas de Gestão da Qualidade. Rio de Janeiro, 2001.
- BANNMAN, P L. Risk and risk management in software projects: A reassessment. *The Journal of Systems and Software*, v. 81, n. 12, p. 2118-2133, 2008.
- BOEHM, B. W. Software Risk Management: Principles and Practices. *IEEE Software*. V. 8. N. 1. p. 32-41, Jan. 1991.
- BOURQUE, P. et al. SWEBOK: Guide to the Software Engineering Body of Knowledge. 3 ed. Nova Jersey, 2014
- CHARRETE, R. (2005), “Why software fails”, *IEEE Spectrum*, Vol.42 No.9
- FORD MOTOR COMPANY. - Potential failure mode and effects analysis (FMEA) Reference Manual. 1988.
- GILB, T. Principles of Software Engineering Management. Wokingham, England, Add ABNT/ISON Wesley, 1988.
- Handbook of Occupational Safety and Health, Second Edition, edited by Lou Diberardinis, “Chapter 6 Risk Assessment Techniques,” Thomas M. Dougherty, pp. 127-178, John Wiley and Sons, 1999.
- JOHNSON, D. M. The Systems Engineer and the Software Crisis. 21 ed. Editora: ACM SIGSOFT. 1996.
- KOTONYA, G.; SOMMERVILLE, I. Requirements Engineering: Processes and Techniques. Editora: Willer. 1998.
- LEOPOLDINO, Cláudio Bezerra. Avaliação de Risco em Desenvolvimento de Software. Dissertação de Mestrado, Universidade do Rio Grande do Sul, Rio Grande do Sul, 2004
- LIPSHITZ, R.; STRAUSS, O. Coping With Uncertainty: A Naturalistic Decision Making Analysis. *Organizational Behavioral and Human Decision Processes*, 1997
- MACEDO, M; SALGADO, E. Gerenciamento de Risco Aplicado ao Desenvolvimento De Software. In: *Sistemas & Gestão* 10,2015. Alfenas.
- PMI - Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK® Guide), 2013.
- Project Management Institute - PMI®. A Guide to Project Management Body of Knowledge - PMBOK® Guide. 5th edition. Project Management Institute, Inc. Pennsylvania, 2013.
- ROCHA, P. C.; BELCHIOR, A. D. Mapeamento do Gerenciamento de Riscos no PMBOK, CMMI-SW e RUP. In: *Simpósio Internacional de Melhoria de Processos de Software*, 4, São Paulo, 2004.
- SCHUYLER, J. (2001), Risk and decision analysis in projects, 2 ed., Newtown Square, USA.
- SOMMERVILLE, Ian. Engenharia de software. Engenharia de requisitos. 9ª edição. São Paulo: Pearson Prentice Hall, 2011.
- SURI, P.K.; NARULA, Kanchan. Simulating the Probability of Risk During Project Completion. *International Journal of Advanced Research in Computer Science and Software Engineering*, v. 3, 2013.
- TOM GILB, Principles of Software Engineering Management, Wokingham, England: Addison Wesley, 1988.



- ENGHOLM, Hélio. Engenharia de software na prática. 1ª edição. São Paulo: Novatech, 2013.
- Universidade Metodista de Piracicaba. Especificação e implementação de uma ferramenta para elicitação de requisitos de software baseada na teoria da atividade. Piracicaba, SP, 2005. p. 21.
- WIEGERS, K.; BEATTY, J. software requirement .3 ed. Editora: Pearson Education, 2013.
- MIDDLETON, P.; JOYCE, D. Lean software management: BBC worldwide case study. IEEE Transactions on Engineering Management, v. 59, n. 1, p. 20–32, 2012.
- KERZNER, H. Strategic Planning for Project Management using Project Management Maturity Model. New York, NY: John Wiley & Sons.”; 2001.
- SCHUH, P. Integrating agile development in the real word. Programming Series. Charle Rives Media, 2005.
- PRADO, Felipe. Análise de Riscos em Projetos de Software Disponível em <[https://www.researchgate.net/publication/231703381\\_Analise\\_de\\_Riscos\\_em\\_Projetos\\_de\\_Softwar](https://www.researchgate.net/publication/231703381_Analise_de_Riscos_em_Projetos_de_Softwar)> , acesso em: 2 de janeiro de 2019.
- SCHWABER & SUTHERLAND, 2009 SCHWABER K.; SUTHERLAND J. Scrum Guide: Developed and sustained. Scrum.org. 2009.
- PRESSMAN, Roger. Engenharia de software. Engenharia de requisitos. 8ª edição. São Paulo: Pearson Prentice Hall, 2016.
- BROWN, Anthony. Análise de risco. Disponível em: <[http://scholar.googleusercontent.com/scholar?q=cache:xVPzzX7BTIYJ:scholar.google.com/+analise+de+risco&hl=pt-BR&as\\_sdt=0,5](http://scholar.googleusercontent.com/scholar?q=cache:xVPzzX7BTIYJ:scholar.google.com/+analise+de+risco&hl=pt-BR&as_sdt=0,5)>, acesso em: 13 de dezembro de 2018.
- BONANOMI, Roberto. Aplicação da teoria grey e fmea – análise dos modos de Falha e efeitos na priorização de riscos de projetos de desenvolvimento de software produto Disponível em: <<https://periodicos.utfpr.edu.br/revistagi/article/view/678/578> >acesso em: 11 de dezembro de 2018.
- DOUGHERTY, Thomas. Disponível em: <<http://web.mit.edu/10.27/www/1027CourseManual/1027CourseManual-AppVI.html>>, acesso em: 10 de novembro de 2018
- WESTFALL, Linda. SOFTWARE RISK MANAGEMENT Disponível em: <[http://www.westfallteam.com/Papers/risk\\_management\\_paper.pdf](http://www.westfallteam.com/Papers/risk_management_paper.pdf)>, acesso em: 25 de dezembro de 2018.
- KRIGSMAN, Michael. 68 PERCENT OF IT PROJECTS FAIL. Disponível em: <<https://www.zdnet.com/article/study-68-percent-of-it-projects-fail/>>, acesso em: 07 de outubro de 2018.
- ABNT- Associação Brasileira de Normas Técnicas. NBR ABNT/ISO 31010:2012 Técnicas para o processo de avaliação de riscos. Rio de Janeiro, ABNT, 2012.
- SOARES, Michel. Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software Disponível em <<http://www.periodicosibepes.org.br/index.php/reinfo/article/view/146/38>>, acesso em: 13 de janeiro de 2019.
- LEITÃO, Michele. TRABALHO DE CONCLUSÃO DE CURSO ENGENHARIA DE COMPUTAÇÃO. Disponível em <[https://tcc.ecomp.poli.br/20101/TCC\\_final\\_Michele.pdf](https://tcc.ecomp.poli.br/20101/TCC_final_Michele.pdf)>, acesso em: 16 de janeiro de 2019.
- VARGAS, Ricardo. Técnicas básicas de identificação de riscos. Disponível em <<https://ricardo-vargas.com/pt/downloads/download-file/6439>>, acesso em: 16 de outubro de 2018.

## 10 ANEXO

Risco
A distribuição geográfica da equipe pode afetar no custo, prazo e/ou escopo do projeto
Criação de requisitos pela equipe de desenvolvimento
Falha de comprometimento do cliente
Atrito entre os desenvolvedores
Carga tributária elevada
Decisões serem limitadas/afetadas pelas características não funcionais dentro do mesmo subsistema cogitado/escolhido
O tamanho da equipe pode afetar no prazo, escopo e/ou custo
Introdução de nova tecnologia
Cronograma fora da realidade
Falta de motivação da equipe
Sem definição de função dos membros da equipe
Inadequately trained development team members
Inexperienced team members
Frequent conflicts among development team members
Frequent turnover within the project team
Negative attitudes by development team
Team members lack specialized skills required by the project
High level of technical complexity
Project affects a large number of user departments or units
Large number of links to other systems required
Immature technology

Project involves use of technology that has no been used in prior projects
Lack of an effective project management methodology
Lack of people skills in project leadership
Inadequate estimation of required resources
Poor project planning
Project milestones not clearly defined
Inadequate estimation of project budget
Ineffective project manager
Inexperienced project manager
Ineffective communication
Ferramentas impróprias para o desenvolvimento
Projetos de múltiplos fornecedores
Mudança na propriedade ou no gerentes sênior do projeto
Principais gestores tomam decisões sem consultar os demais participantes do projeto.
Membros da equipe desmotivados.
Falta de poderes para o gerenciamento de projetos.
Escopo ou objetivos pouco claros ou mal interpretados
Não obtenção de financiamentos
Ter vantagens para um tipo negativo de usuário
Ter um viés de interpretação negativa (gerar um modelo de uso negativo)
Os padrões de arquitetura já implementaos podem afetar/limitar as decisões a serem tomadas
Características estruturais existentes que podem afetar nas decisões
A existente escolha dos protocolos de comunicação pode afetar na decisões
O nível de experiência da equipe pode afetar no custo, prazo e/ou no escopo do projeto
Mudanças nas leis de impostos
Mudanças nos softwares devido as exigências do mercado

Requisito do projeto mal entendido
Mudança de membro da equipe
Crescimento de requisitos não previstos
Resources shifted from the project due to changes in organizational priorities
Corporate politics with negative effect on project
Unstable organizational environment
Dependency on outside suppliers
Volatilidade do pessoal da equipe
Assunto novo ou não familiar
Definição inadequada de funções e de responsabilidades.
Projeto extenso/complexo.
Gerente de projetos inexperiente/sem habilidades necessárias.
Alta rotatividade da equipe.
Falhas na comunicação.
Membros da equipe treinados inadequadamente.
Conflito e falta cooperação entre os membros da equipe
Projeto de múltiplos fornecedores
Falha em obter comprometimento do cliente com o projeto.
Usuários resistentes às mudanças.
Mudança de propriedade/gestão/prioridade organizacional.
Introdução de novas tecnologias
Assunto novo ou não familiar tanto para os usuários quanto para os desenvolvedores
Falha em gerenciar as expectativas dos usuários finais
Número de unidades organizacionais do cliente envolvidas
Conflitos entre departamentos do usuário
Uso da história de Usuário contra o mesmo

Decisões serem limitadas/afetadas pelo hardware existente
A reusabilidade de módulos pode limitar/afetar decisões
Decisões tomadas já foram feitas em arquiteturas anteriores
A tecnologia escolhida para a execução do projeto pode afetar no custo, prazo e/ou escopo
Fugir do escopo
Escopo genérico
Mudança de requisito
Ignorar o risco presente no projeto
undefined project success criteria
Continually changing project scope/objectives
System requirements not adequately identified
Incorrect system requirements
undefined project goals
Users lack understanding of system capabilities and limitations
Difficulty in defining the inputs and outputs of the system
Falta de cooperação dos usuários
Falta de poderes para o gerenciamento de projetos
Pessoal envolvido insuficiente ou inapropriado
Falha em gerenciar as expectativas dos usuários finais
Falta de habilidades para o gerenciamento de projetos
Custos mal estimados
Falha em obter comprometimento do cliente
Falta de motivação da equipe
Definição imprópria de papéis e responsabilidades
Falta de metodologia efetiva de gerenciamento de projetos
Adoção de novo método/tecnologia

Falta de conhecimentos/habilidades interpessoais na liderança
Estimativa inadequada dos recursos necessários.
Cronograma irreal.
Conflito e falta cooperação entre os usuários
Falha em gerenciar expectativas do usuário final.
Falta de uma metodologia efetiva para o gerenciamento de projetos
Volatilidade do pessoal envolvido
Gerenciamento inadequado de mudanças
Falta de conhecimentos e/ou habilidades necessárias da equipe do projeto
Requisitos mal interpretados e/ou mal definidos no início do desenvolvimento
Falha em obter comprometimento do usuário por parte do gerente do projeto
Falta de habilidades interpessoais dos gestores na liderança da equipe do projeto
Falta de comprometimento da alta gerência com o projeto
Invasão/Vulnerabilidade de Privacidade
Quebra de segurança (no sentido de dano material)
Quebra de Segurança (No sentido de danos pABNT/ISOLógicos)
Decisões serem limitadas/afetadas pela características não funcionais em subsistemas diferentes que estiverem sendo cogitados/escolhidos
A existência de características que podem ser facilmente modificadas
Pessoal insuficiente
Quebra de Equipamento
Perda de dados do documento de requisito
Users resistant to change
Conflict between users
Escopo/ objetivos pouco claros ou equivocados
Mudança de Escopo/ objetivos

Prazos e tempo para tarrfas mal estimados
Planejamento inexistente ou inadequado
Volatilidade no requisitos
Falta de comprometimento da alta gerência
Gerenciamento impróprio de mudanças
Falta de estabelecimento de processos/procedimentos/metodologia/planejamento do projeto.
Especificações de requisitos/objetivos mal definidos.
Tecnologia nova/imatura/não familiar para usuários e membros da equipe do projeto.
Falta de conhecimento/habilidades necessárias no projeto

Tabela A.1 – Aplicabilidade das ferramentas utilizadas para o processo de avaliação de riscos

Ferramentas e técnicas	Processo de avaliação de riscos					Ver Anexo
	Identificação de riscos	Análise de riscos			Avaliação de riscos	
		Consequência	Probabilidade	Nível de risco		
<i>Brainstorming</i>	FA <sup>1</sup>	NA <sup>2</sup>	NA	NA	NA	B 01
Entrevistas estruturadas ou semi-estruturadas	FA	NA	NA	NA	NA	B 02
Delphi	FA	NA	NA	NA	NA	B 03
Listas de verificação	FA	NA	NA	NA	NA	B 04
Análise preliminar de perigos (APP)	FA	NA	NA	NA	NA	B 05
Estudo de perigos e operabilidade (HAZOP)	FA	FA	A <sup>3</sup>	A	A	B 06
Análise de perigos e pontos críticos de controle (APPCC)	FA	FA	NA	NA	FA	B 07
Avaliação de risco ambiental	FA	FA	FA	FA	FA	B 08
Técnica estruturada "E se" (SWIFT)	FA	FA	FA	FA	FA	B 09
Análise de cenários	FA	FA	A	A	A	B 10
Análise de impactos no negócio	A <sup>2</sup>	FA	A	A	A	B 11
Análise de causa-raiz	NA	FA	FA	FA	FA	B 12
Análise de modos de falha e efeito	FA	FA	FA	FA	FA	B 13
Análise de árvore de falhas	A	NA	FA	A	A	B 14
Análise de árvore de eventos	A	FA	A	A	NA	B 15
Análise de causa e consequência	A	FA	FA	A	A	B 16
Análise de causa e efeito	FA	FA	NA	NA	NA	B 17
Análise de camadas de proteção (LOPA) c	A	FA	A	A	NA	B 18
Árvore de decisões	NA	FA	FA	A	A	B 19



Tabela A.1 (continuação)

Ferramentas e técnicas	Processo de avaliação de riscos					Ver Anexo
	Identificação de riscos	Análise de riscos			Avaliação de riscos	
		Consequência	Probabilidade	Nível de risco		
Análise da confiabilidade humana	FA	FA	FA	FA	A	B 20
Análise <i>Bow tie</i>	NA	A	FA	FA	A	B 21
Manutenção centrada em confiabilidade	FA	FA	FA	FA	FA	B 22
<i>Sneak analysis (SA)</i> e <i>sneak circuit analysis (SCA)</i>	A	NA	NA	NA	NA	B 23
Análise de Markov	A	FA	NA	NA	NA	B 24
Simulação de Monte Carlo	NA	NA	NA	NA	FA	B 25
Estatística Bayesiana e Redes de Bayes	NA	FA	NA	NA	FA	B 26
Curvas FN	A	FA	FA	A	FA	B 27
Índices de risco	A	FA	FA	A	FA	B 28
Matriz de probabilidade/consequência	FA	FA	FA	FA	A	B 29
Análise de custo/benefício	A	FA	A	A	A	B 30
Análise de decisão por multicritérios ( <i>MCDM</i> )	A	FA	A	FA	A	B 31

<sup>1</sup> FA - Fortemente aplicável.  
<sup>2</sup> NA - Não aplicável.  
<sup>3</sup> A - Aplicável.