



ÁREA DE INFORMÁTICA

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS – ADS

MILENE ELIZABETH RIGOLIN FERREIRA LOPES SALVADOR

**DOMAR - UMA ABORDAGEM PARA A PREVENÇÃO DE
PROBLEMAS RELACIONADOS A GESTÃO E DOCUMENTAÇÃO DE
REQUISITOS**

TRABALHO DE CONCLUSÃO DE CURSO – TCC

CARAGUATATUBA

2015

DOMAR - UMA ABORDAGEM PARA A PREVENÇÃO DE PROBLEMAS RELACIONADOS A GESTÃO E DOCUMENTAÇÃO DE REQUISITOS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Instituto Federal de São Paulo.

Orientadora:
Prof.^a Dr.^a Luciana Brasil R. dos Santos

**CARAGUATATUBA
2015**

S182d Salvador, Milene Elizabeth Rigolin Ferreira Lopes
DoMaR : uma abordagem para a prevenção de problemas relacionados a gestão e documentação de requisitos / Milene Elizabeth Rigolin Ferreira Lopes Salvador. – Caraguatatuba, 2015.
47 f.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) -- Instituto Federal de São Paulo, Câmpus Caraguatatuba, 2015.

1. Engenharia de Requisitos. 2. Problemas. 3. Documentação. 4. Gerenciamento. 5. Rastreabilidade. I. Título.

CDD 005.1

TERMO DE APROVAÇÃO

DOMAR - UMA ABORDAGEM PARA A PREVENÇÃO DE PROBLEMAS
RELACIONADOS A GESTÃO E DOCUMENTAÇÃO DE REQUISITOS

por

MILENE ELIZABETH RIGOLIN FERREIRA LOPES SALVADOR

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 09 de Novembro de 2015 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas (ADS). O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados, a qual após deliberação, considerou o trabalho aprovado.



Prof.ª Dr.ª Luciana Brasil R. dos Santos
Orientadora



Prof. Esp. Eduardo Pereira de Sousa
Presidente



Prof. MSc. Renan Cavichi de Freitas
Membro

“O único lugar onde o sucesso vem antes do trabalho é no dicionário ”.
(Albert Einstein)

AGRADECIMENTOS

Gostaria de agradecer a Deus, este ser supremo que nos guia sempre para o melhor.

Ao meu marido Henrique pelo apoio, incentivo e paciência.

À minha orientadora Luciana pela amizade, esclarecimentos, direcionamentos e orientação.

À minha mãe, irmão, tios e primos pelo apoio.

Aos especialistas que participaram do estudo de caso do TCC.

Agradeço também a todos os professores e colegas que contribuíram e fizeram parte desse processo.

RESUMO

A execução adequada do processo de Engenharia de Requisitos (ER) nas fases iniciais do projeto de Engenharia de *Software* é uma das formas mais seguras para se obter qualidade no desenvolvimento e no produto de *software* gerado. Porém, apesar do crescente número de projetos que utilizam o processo de ER, especialistas constataam que o mesmo não está sendo bem estruturado e executado, ocasionando diversos problemas, tanto na coleta quanto na gestão e documentação dos requisitos. Uma das formas de tentar auxiliar na estruturação e execução do processo de ER é prover maneiras de prevenir alguns de seus problemas. Alguns problemas comuns no processo de ER estão relacionados a gestão e documentação dos requisitos, como por exemplo: perda de requisitos, não rastrear os requisitos, não priorizar, descartar informações importantes, entre outros. Partindo desse pressuposto o presente trabalho propõe-se a investigar a área de ER, verificando problemas na gestão e documentação de requisitos, com o intuito de mapear esses problemas e propor um sistema para auxiliar no registro, organização e monitoramento de requisitos visando prevenir alguns problemas relacionados a gestão e documentação de requisitos que ocorrem durante a execução do processo de ER. Com o intuito de validar o sistema proposto, fez-se um estudo de caso com especialistas da área de ER, onde os mesmos puderam utilizar o sistema e avaliar se as funcionalidades implementadas no sistema podem prevenir os problemas relacionados. A avaliação com os especialistas mostrou que o sistema tem potencial para prevenir os problemas elencados na pesquisa.

Palavras-chave: Engenharia de Requisitos. Problemas. Documentação. Gerenciamento. Rastreabilidade.

ABSTRACT

The proper implementation of the requirements engineering process (RE) in the early stages of software engineering project is one of the safest ways to get quality in the development and generated software product. But, despite the growing number of projects using the RE process, experts realize that it is not well structured and executed, causing many problems, both in collecting and in the management and documentation of requirements. One way to try to assist in structuring and implementing the RE process is to provide ways to prevent some of their problems. Some common problems in the RE process are related to management and documentation of requirements, such as: loss of requirements, do not trace the requirements, do not prioritize, discard important information, among others. Based on this assumption the present work aims to investigate the RE area, checking the problems relating to the management and documentation of requirements, in order to map these problems. It also proposes an system to assist the requirements registration, organization and monitoring, in order to prevent some problems related to management and documentation. In order to validate the proposed system, a case study was realized with specialists of RE area. The specialists used the system and assess whether the features implemented in environment may prevent related problems. The case study results showed that the environment features have the potential to prevent the problems reported in the survey.

Keywords: Requirements Engineering. Problems. Documentation. Management. Traceability.

LISTA DE FIGURAS

Figura 1 – Índices do CHAOS <i>Report</i> 2012.	13
Figura 2 – Escopo do Trabalho – etapas tratadas no trabalho.	14
Figura 3 – Modelo em Espiral do Processo de Engenharia de Requisitos.	19
Figura 4 – Documentos para esclarecimento sobre a diferença entre Restrições e Requisitos.	26
Figura 5 – Modelo Entidade e Relacionamento do BD do sistema.	26
Figura 6 – Modelo relacional do BD do sistema.	27
Figura 7 – Interação entre usuário e o sistema proposto.	27
Figura 8 – Tela inicial do sistema proposto.	28
Figura 9 – Tela de <i>login</i> do sistema proposto.	29
Figura 10 – Tela de cadastro de usuário.	29
Figura 11 – Tela de cadastro de origem do requisito.	30
Figura 12 – Tela de cadastro de requisito.	30
Figura 13 – Tela de alteração de origem do requisito.	31
Figura 14 – Tela de requisitos cadastrados a serem mapeados.	31
Figura 15 – Tela de mapeamento entre requisitos.	32
Figura 16 – Matriz de mapeamento entre requisitos.	32
Figura 17 – Tela de consulta de requisitos.	33
Figura 18 – Tela do histórico de modificações gerais.	33
Figura 19 – Resultados da avaliação dos especialistas.	37
Figura 20 – Médias dos resultados do Estudo de Caso.	38

LISTA DE SIGLAS

ER	Engenharia de Requisitos
ESW	Engenharia de <i>Software</i>
TCC	Trabalho de Conclusão de Curso
BD	Banco de Dados
HTML	<i>Hipertext Markup Language</i>
PHP	<i>Hypertext Preprocessor</i>
CSS	<i>Cascading Style Sheets</i>
SGBD	Sistema de Gerenciamento de Banco de Dados

LISTA DE ACRÔNIMOS

DoMaR *Documentation and Management Requirements*

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO	13
1.2	ESCOPO	14
1.3	PROBLEMA DE PESQUISA	15
1.4	ABORDAGEM DE SOLUÇÃO	15
1.5	CONTRIBUIÇÕES	15
1.6	METODOLOGIA DO TRABALHO	15
1.7	TRABALHOS CORRELATOS	16
1.8	ORGANIZAÇÃO DO TRABALHO	17
2	ENGENHARIA DE REQUISITOS	18
2.1	REQUISITOS	18
2.2	PROCESSO DA ENGENHARIA DE REQUISITOS	19
2.2.1	EXTRAÇÃO DE REQUISITOS	19
2.2.2	ANÁLISE E NEGOCIAÇÃO DE REQUISITOS	20
2.2.3	DOCUMENTAÇÃO DE REQUISITOS	21
2.2.4	VALIDAÇÃO DE REQUISITOS	21
2.3	GERENCIAMENTO DE REQUISITOS	21
2.4	PROBLEMAS RELACIONADOS A DOCUMENTAÇÃO E GESTÃO DE REQUISITOS	23
2.4.1	DOCUMENTAÇÃO DE REQUISITOS	23
2.4.2	GERENCIAMENTO DE REQUISITOS	24
3	DESENVOLVIMENTO DO SISTEMA	25
3.1	PROBLEMAS TRATADOS PELO SISTEMA	25
3.2	MODELAGEM DO BANCO DE DADOS	26
3.3	FUNCIONAMENTO DO SISTEMA	27
3.4	IMPLEMENTAÇÃO DO SISTEMA DOMAR	28
3.4.1	PÁGINAS INICIAIS	28
3.4.2	CADASTRO DE DADOS	30
3.4.3	ALTERAÇÃO DE DADOS	31
3.4.4	MAPEAMENTO DE REQUISITOS	31
3.4.5	CONSULTA DE REQUISITOS	33
3.4.6	HISTÓRICO DE MODIFICAÇÕES	33
3.5	MAPEAMENTO ENTRE OS PROBLEMAS E AS FUNCIONALIDADES DO SISTEMA PROPOSTO	34
3.5.1	PROBLEMAS RELACIONADOS COM A DOCUMENTAÇÃO DE REQUISITOS	34
3.5.2	PROBLEMAS RELACIONADOS COM O GERENCIAMENTO DE REQUISITOS	34

4	ESTUDO DE CASO	36
4.1	AVALIAÇÃO	37
4.2	RESULTADOS	37
4.3	SUGESTÕES DOS ESPECIALISTAS	38
5	CONCLUSÃO	40
5.1	CONCLUSÕES OBTIDAS	40
5.2	DIFICULDADES	41
5.3	TRABALHOS FUTUROS	41
	REFERÊNCIAS	42
	APÊNDICE A DOCUMENTOS DO ESTUDO DE CASO	44
A.1	UTILIZAÇÃO DO SISTEMA	44
A.2	AVALIAÇÃO DO SISTEMA	44
	APÊNDICE B QUESTIONÁRIO DE AVALIAÇÃO DO SISTEMA	45
B.1	CARACTERIZAÇÃO DO PARTICIPANTE	45
B.2	AVALIAÇÃO DAS FUNCIONALIDADES	45
B.2.1	PROBLEMAS RELACIONADOS COM A DOCUMENTAÇÃO DE REQUISITOS	45
B.2.2	PROBLEMAS RELACIONADOS COM A GERENCIAMENTO DE REQUISITOS	46

1 INTRODUÇÃO

No presente trabalho, propõe-se a realização de uma investigação na área de Engenharia de Requisitos (ER), verificando problemas nas etapas de gestão e documentação de requisitos, com o intuito de mapear esses problemas e propor um sistema para auxiliar no registro, organização e monitoramento de requisitos visando prevenir alguns problemas relacionados a gestão e documentação de requisitos que ocorrem durante a execução do processo de ER.

Neste capítulo relata-se a motivação, escopo do trabalho, o problema de pesquisa, a abordagem de solução, as contribuições do trabalho, a metodologia adotada para o desenvolvimento do Trabalho de Conclusão de Curso (TCC), os trabalhos correlatos e como está organizado este documento.

1.1 Motivação

Apesar do crescente número de projetos que utilizam o processo de ER, especialistas constataam que o mesmo não está sendo bem estruturado e executado, ocasionando diversos problemas, tanto na coleta quanto na gestão e documentação dos requisitos (HULL; JACKSON; DICK, 2005), (ALEXANDER; STEVENS, 2002), (FIRESMITH, 2007) e (VERNER et al., 2006). Com isso os projetos de *software* apresentam taxas elevadas de falhas e problemas, as quais podem ser observadas na Figura 1.

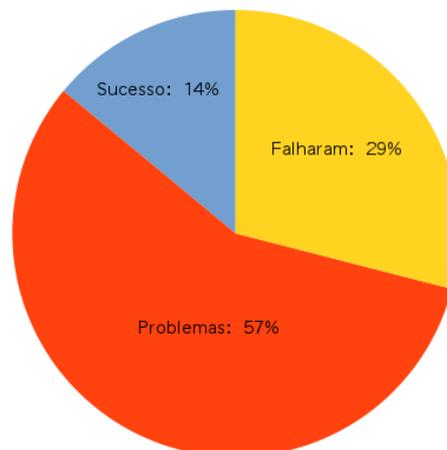


Figura 1 – Índices do CHAOS Report 2012.

Fonte: (GROUP, 2012)

Esse relatório indica que 14% dos projetos são executados com sucesso, ou seja, entregam o produto no prazo, com as funcionalidades solicitadas, dentro do orçamento e com o escopo completo; 57% dos projetos tiveram problemas, como, atrasos, extração do orçamento e/ou entrega do produto com quantidade de funções inferior à solicitada; e 29% projetos falharam, entende-se como falha não entregar o produto, en-

trega sem as funcionalidades requeridas, entrega de parte do produto e o cancelamento do projeto antes do previsto; entende-se como problemas os atrasos de cronograma, a extrapolação do orçamento, entre outros (KOTONYA; SOMMERVILLE, 2000).

Os principais problemas que ocorrem na execução de um projeto de *software* estão relacionados a área de ER (HULL; JACKSON; DICK, 2005), (ALEXANDER; STEVENS, 2002), (FIRESMITH, 2007) e (VERNER et al., 2006). Esses problemas ocorrem na ER pois a maioria das atividades são desenvolvidas por humanos e com isso são mais propensas a erros. Uma das principais consequências de requisitos mal especificados é o retrabalho, que pode consumir de 30 a 50% do custo total de desenvolvimento (BOEHM; PAPACCIO, 1988 apud WIEGERS, 2003) e os erros nos requisitos causam 70 a 85% do custo de retrabalho (LEFFINGWELL, 1997 apud WIEGERS, 2003).

Uma das formas de tentar auxiliar na estruturação e execução do processo de ER é prover maneiras de prevenir alguns de seus problemas. Dessa forma, o presente trabalho apresenta uma abordagem para prevenção de alguns problemas nas etapas de gestão e documentação de requisitos.

1.2 Escopo

O trabalho tem como escopo as etapas de gestão e documentação de requisitos do processo de ER, deste modo serão levantados e mapeados apenas os problemas que ocorrem nessas duas etapas, conforme mostra a Figura 2. A metodologia que será levada em consideração para o desenvolvimento do projeto de *software* é a tradicional, como por exemplo projetos que utilizam o modelo cascata e o modelo espiral.

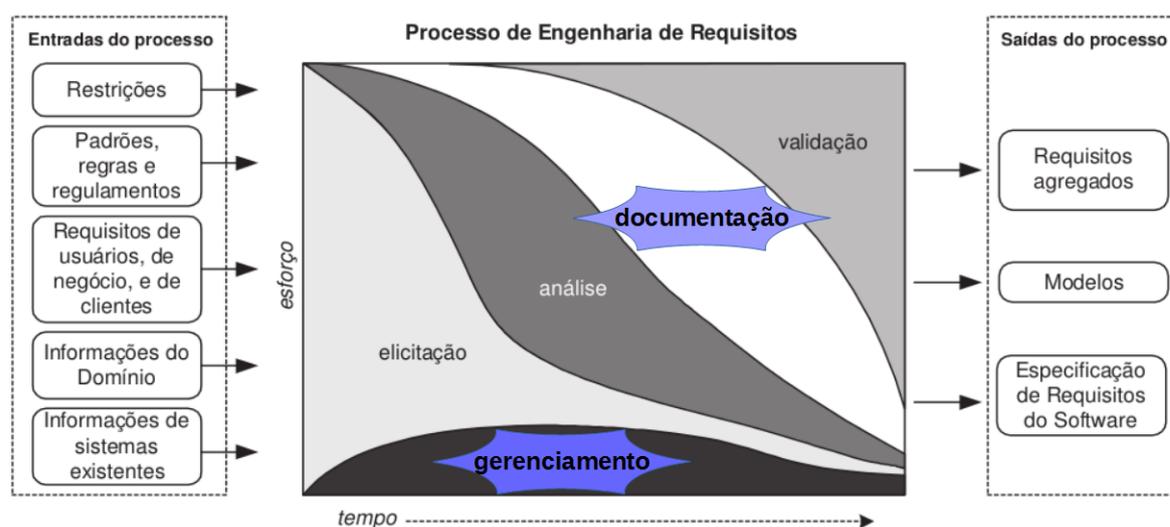


Figura 2 – Escopo do Trabalho – etapas tratadas no trabalho.

Fonte: Adaptada (NASA, 1990 apud GENVIGIR, 2009) e (KOTONYA; SOMMERVILLE, 2000 apud GENVIGIR, 2009)

1.3 Problema de Pesquisa

O problema a ser tratado no presente trabalho é como melhorar a execução das atividades relacionadas às etapas de gestão e documentação de requisitos, prevenindo o aparecimento de alguns de seus problemas, tais como: perda de requisitos, não rastrear os requisitos, não priorizar os requisitos e descartar informações importantes.

1.4 Abordagem de Solução

A abordagem de solução proposta pelo presente trabalho é realizar uma investigação da área de ER, verificando os problemas das etapas de gestão e documentação de requisitos, com o intuito de mapear esses problemas e propor um sistema para auxiliar no registro, organização e monitoramento de requisitos visando prevenir alguns problemas relacionados a gestão e documentação de requisitos que ocorrem durante a execução do processo de ER.

1.5 Contribuições

As contribuições do presente trabalho são:

- Produção de material sobre a área de ER;
- Levantamento dos problemas relacionados a gestão e documentação de requisitos; e
- Sistema para auxiliar no registro, organização e monitoramento de requisitos.

1.6 Metodologia do Trabalho

A metodologia adotada no desenvolvimento do trabalho consiste na:

1. Revisão bibliográfica no contexto da ER;
2. Verificação de trabalhos correlatos;
3. Levantamento dos problemas das etapas de gestão e documentação de requisitos;
4. Construção de um sistema que dê suporte ao processo da ER; e
5. Aplicação do sistema em um estudo de caso.

1.7 Trabalhos Correlatos

Existem diversos sistemas comerciais que oferecem apoio no processo de ER, como por exemplo: *DOORS*, *Caliber*, *RequisitePro*, entre outros. Existem também iniciativas acadêmicas de protótipos de sistemas com o intuito de auxiliar na gestão, documentação e rastreamento de requisitos, dentre eles pode-se ressaltar os trabalhos de (SANT'ANNA, 2000), (GRANDE; MARTINS, 2006), (LOPES, 2011), (PONTES, 2006) e (SALES et al., 2010).

Alguns trabalhos acadêmicos, tentam auxiliar o processo de ER provendo soluções pontuais, ou seja alguns tratam só de rastreamento, outros só de documentação, outros se preocupam com a gestão de requisitos e assim por diante. Mas existem também trabalhos que visam a melhora do processo de desenvolvimento de *software* como um todo, se preocupando em resolver os problemas mas de maneira global. Desse modo, os trabalhos descritos aqui são correlatos com a abordagem proposta, pois possuem o mesmo objetivo de melhorar o processo de ER.

No trabalho de Sant'Anna (2000) discute-se como deixar o processo de gestão e desenvolvimento de *software* mais eficiente utilizando o trabalho cooperativo e distribuído, gestão integrada do conhecimento e dos eventos ocorridos nos projetos, participação ativa de agentes autônomos computacionais. Para dar suporte ao processo de desenvolvimento e gestão criou, com base nos conceitos discutidos, um sistema integrado para o apoio ao desenvolvimento e gestão de projetos de *software* para sistemas de controle de satélites.

Os autores Grande e Martins (2006), criaram a ferramenta SIGERAR com o intuito de auxiliar no gerenciamento de requisitos, fazendo a coleta, o armazenamento e a manutenção dos requisitos acordados, durante o ciclo de vida do *software*. Desse modo, realiza o rastreamento dos relacionamentos entre requisitos e requisitos-documentos no processo de desenvolvimento do sistema. Os autores visaram o controle de rastreamento e versionamento dos requisitos.

No trabalho de Lopes (2011), fez-se um sistema especialista para verificar se os problemas que ocorriam no processo de ER poderiam ser causados por erros humanos. No trabalho o sistema especialista sugere soluções para os erros com o intuito de solucionar os problemas do processo avaliado.

No trabalho de Pontes (2006), criou-se uma ferramenta para rastrear requisitos, onde existe uma matriz de rastreabilidade que mostra a dependência entre requisitos, casos de uso, casos de teste e entre caso de uso e diagrama de classe. Nesse trabalho, o autor não cita problemas do processo de ER a serem resolvidos, mas propõe um local

que dê apoio ao rastreamento dos requisitos durante o processo de ER.

O artigo de Sales et al. (2010) apresenta uma ferramenta para apoiar a gerência de requisitos que é integrada ao sistema de desenvolvimento de *software* centrado em processos. Possui as seguintes funcionalidades: gerenciamento, manutenção, rastreabilidade e controle de mudança dos requisitos. Essa ferramenta emite relatórios, como por exemplo, a lista de requisitos e o relatório de impacto de mudança para o requisito.

O sistema proposto possui o mesmo objetivo que os trabalhos correlatos relatados nessa seção, que é melhorar o processo de ER, porém o mesmo traz funcionalidades específicas para tentar prevenir problemas relacionados a gestão e documentação de requisitos. O sistema se diferencia dos demais trabalhos, pois propõe um sistema integrado de gestão e documentação de requisitos, o que normalmente se encontram sistemas separados e específicos para essas duas atividades.

1.8 Organização do Trabalho

Para melhor entendimento do trabalho, este documento está organizado nos seguintes capítulos: o Capítulo 2 descreve a área de Engenharia de Requisitos e os problemas que serão abordados no TCC; no Capítulo 3 relata-se o desenvolvimento do sistema proposto; o Capítulo 4 apresenta o estudo de caso e seu resultado; e o Capítulo 5 apresenta as conclusões do trabalho realizado.

2 ENGENHARIA DE REQUISITOS

A Engenharia de Requisitos é uma subárea da Engenharia de *Software* que fornece métodos, técnicas e ferramentas que auxiliam o processo de coleta, documentação e gestão de requisitos (SOMMERVILLE, 2007).

Este capítulo apresenta os principais conceitos sobre a Engenharia de Requisitos e para melhor organização está dividido em: na Seção 2.1 mostra-se a definição de requisitos e seus tipos; na Seção 2.2 apresenta-se o processo de engenharia de requisitos; na Seção 2.3 descreve-se a disciplina de gerenciamento de requisitos; na Seção 2.4 descreve-se os problemas tratados no presente trabalho, relacionados a gestão e documentação de requisitos.

2.1 Requisitos

Um requisito pode ser definido como descrições dos serviços fornecidos pelo sistema e suas restrições operacionais (SOMMERVILLE, 2007).

A IEEE *Standard Glossary of Software Engineering Terminology* (ELECTRICAL; ENGINEERS, 1990) define requisito como **(1)** uma condição ou capacidade necessária, a ser provida pelo sistema em desenvolvimento, para que um usuário possa resolver algum problema ou alcançar um objetivo; **(2)** Uma condição ou capacidade a ser encontrada no sistema ou em um componente do sistema para satisfazer um contrato, normas ou outro documento instituído formalmente; e **(3)** Uma representação documentada de uma condição ou capacidade como nos itens 1 ou 2.

Segundo Wieggers (2003), requisito é uma propriedade que o produto deve conter para prover aos *stakeholders*. Já Young (2004), define requisito como um atributo necessário em um sistema, uma declaração que identifica uma capacidade, característica ou fator de qualidade, com o intuito do sistema ter valor e utilidade para os *stakeholders*.

Segundo Sommerville (2007), existem dois tipos de requisitos:

- **Requisitos de usuário:** são declarações das necessidades e restrições do usuário em linguagem natural podendo utilizar diagramas para facilitar o entendimento.
- **Requisitos de sistema:** definem funções, serviços e restrições operacionais do sistema detalhadamente, pode-se utilizar linguagem técnica e formal. Dividem-se em:
 - **Requisitos funcionais:** descrevem as funcionalidades do sistema e como o sistema deve reagir às entradas específicas, e se comportar em determi-

nadas situações.

- **Requisitos não funcionais:** descrevem as restrições dos serviços ou funções oferecidos pelo sistema.
- **Requisitos de domínio:** refletem as características e as restrições do domínio de aplicação do sistema a ser desenvolvido.

2.2 Processo da Engenharia de Requisitos

O processo da ER consiste em descobrir, analisar, documentar e verificar essas funções e restrições, com o intuito de gerar um documento contendo a especificação dos requisitos do sistema.

É composto pelas etapas de Extração de requisitos, Análise e Negociação de requisitos, Documentação de requisitos e Validação de requisitos, como mostra a Figura 3. e para organizar essas etapas, tem-se a área de Gerenciamento de Requisitos. Essas etapas não acontecem em uma sequência estrita; elas se repetem durante todo o processo de desenvolvimento, quando necessário (SOMMERVILLE, 2007).

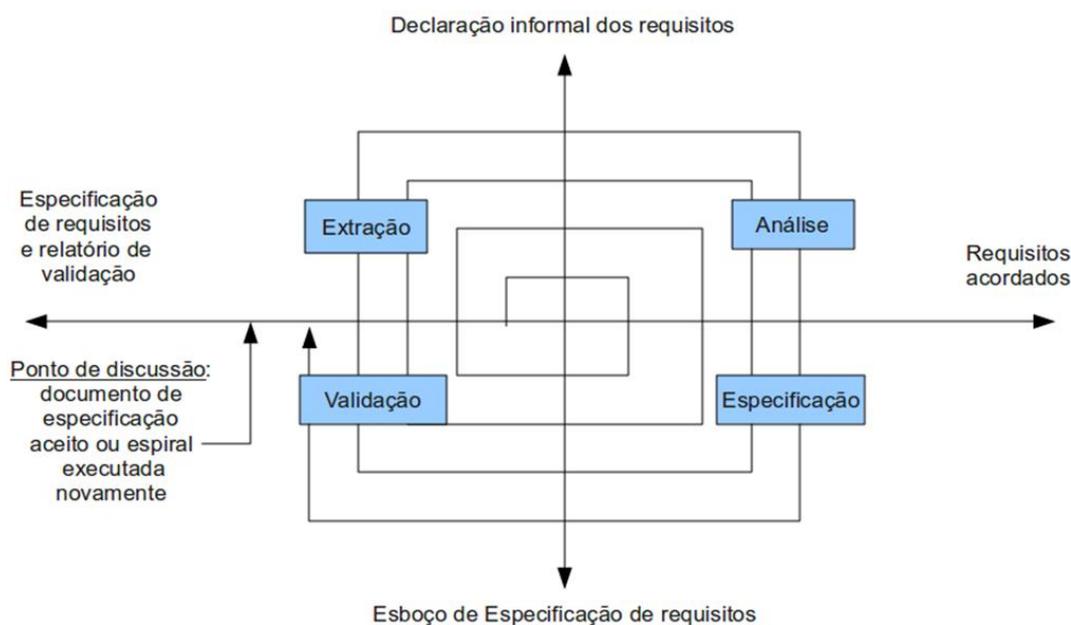


Figura 3 – Modelo em Espiral do Processo de Engenharia de Requisitos.

Fonte: (SOMMERVILLE, 2007).

2.2.1 Extração de Requisitos

A etapa de extração de requisitos tem por objetivo a descoberta dos requisitos do sistema por meio da interação com os *stakeholders*. Existem diversas técnicas para captura de conhecimentos que auxiliam na coleta dos requisitos, dentre elas destacam-se (SOMMERVILLE, 2007):

- **Entrevista** – Existem dois tipos de entrevistas: as entrevistas formais, onde os *stakeholders* se restringem a responder a um grupo de perguntas e as entrevistas informais, onde não existe um roteiro predefinido, deixando os *stakeholders* livres para responder da maneira que quiserem;
- **Pontos de vista** – consistem na análise de diversos pontos de vista dos *stakeholders* e é comumente usada como meio de classificação dos *stakeholders* e de outras fontes de requisitos. Essa técnica está dividida em: (1) pontos de vista de interação que representam as pessoas ou outros sistemas que interagem diretamente com o sistema a ser desenvolvido; (2) pontos de vista indiretos que representam os *stakeholders* que não utilizam o sistema diretamente; e (3) pontos de vista de domínio que representam as características e restrições de domínio que influenciam os requisitos do sistema;
- **Cenários** – são utilizados para melhorar o entendimento dos requisitos do sistema e devem incluir a descrição das expectativas dos usuários em relação ao sistema, o fluxo normal de eventos, os eventos que podem dar errado e a forma de tratá-los, o estado do sistema no fim do cenário e as informações sobre outras atividades que podem acontecer simultaneamente;
- **Casos de uso** – consistem na identificação das interações individuais dos *stakeholders* com o sistema;
- **Histórias do usuário** – consistem em uma ou mais descrições do problema que o usuário fornece para o especialista, com o intuito de coletar as necessidades do usuário e as transformar em requisitos; e
- **Estudo etnográfico** – consiste na observação do dia-a-dia do ambiente de trabalho do usuário, com o intuito de verificar como são executadas as tarefas reais. Essa técnica auxilia na descoberta de requisitos implícitos do sistema.

2.2.2 Análise e Negociação de Requisitos

Na etapa de análise e negociação de requisitos, os requisitos são analisados e verificados, com o objetivo de descobrir problemas, consertá-los e assim conseguir a aceitação dos *stakeholders*. Possui as seguintes atividades:

- **Classificação e Organização dos requisitos** – consiste em agrupar os requisitos relacionados e organizá-los em conjuntos diferentes;
- **Priorização de requisitos e Resolução de conflitos** – realiza-se a priorização dos requisitos, efetuando a resolução de conflitos entre *stakeholders* por meio de negociação; e

- **Verificação dos requisitos** – consiste em verificar os requisitos para checar: o valor para sua implementação, se estão dentro do escopo e negociar quais requisitos são essenciais e desejáveis.

2.2.3 Documentação de Requisitos

A etapa de documentação dos requisitos consiste em escrever as informações coletadas, na etapa de extração de requisitos, em linguagem formal e dentro de um formato padrão, que implica em obrigatoriedade, com o intuito de elaborar um documento de requisitos.

2.2.4 Validação de Requisitos

A etapa de validação de requisitos consiste em analisar os requisitos, juntamente com os *stakeholders*, com o objetivo de verificar a consistência, o realismo, a abrangência, a validade, a completeza e a testabilidade dos requisitos.

2.3 Gerenciamento de Requisitos

O gerenciamento de requisitos é um processo para compreender e controlar as mudanças dos requisitos do sistema, visando avaliar o impacto das mudanças de requisitos (SOMMERVILLE, 2007).

Para organizar o processo de ER tem-se a disciplina de gerenciamento de requisitos, que consiste em dar suporte ao processo de ER com o intuito de controlar mudanças, rastrear e realizar a gerência de configuração e de qualidade de requisitos (SAYAO; BREITMAN, 2009). As atividades envolvidas são:

- **Controle de mudanças** – consiste em tratar todas as propostas de mudança para que as mesmas ocorram de maneira controlada. Possui três estágios: (1) Análise do problema e especificação da mudança; (2) Análise de mudança e estimativa de custo; e (3) Implementação das mudanças (SOMMERVILLE, 2007);
- **Rastreabilidade de requisitos** – O rastreamento de requisitos fornece a visão dos relacionamentos entre requisitos, projeto e implementação de um dado sistema, visando controlar suas mudanças e dependências (ARTEIRO, 2014);
- **Gerência de Configuração** – possui um conjunto de atividades para administrar as alterações nos requisitos durante o processo de ER. Controla as versões de documentos e modificações realizadas nestes documentos, normalmente utiliza-se ferramentas CASE (*Computer-Aided Software Engineering*) para realizar a gerência; e

- **Gerência de Qualidade de Requisitos** – consiste em garantir que os requisitos estão contemplando as características desejáveis a uma declaração de requisito visando a identificação de problemas e a diferenciação entre os bons e maus requisitos (WIEGERS, 2003). Existem diversas características, dentre elas pode-se ressaltar as descritas por (KAR; BAILEY, 1996):
 - O requisito deve ser necessário: O requisito deve descrever uma capacidade essencial, característica física ou fator de qualidade para o produto ou processo.
 - O requisito deve ser conciso: A declaração do requisito deve conter “o que fazer”, declarado de forma simples e clara.
 - O requisito deve ser livre de implementação: O requisito deve declarar o que os *stakeholders* solicitaram e não como o requisito será alcançado.
 - O requisito deve ser atingível: O requisito deve ser alcançável por um ou mais conceitos de sistema desenvolvido dentro do orçamento disponível.
 - O requisito deve ser auto-contido: O requisito não deve precisar de informações adicionais.
 - O requisito deve ser consistente: O requisito não deve contradizer outros requisitos.
 - O requisito não deve conter ambiguidade: Cada requisito deve propiciar uma e somente uma interpretação; a linguagem utilizada não deve deixar dúvidas ao leitor.
 - O requisito deve ser verificável: O requisito deve ser verificável por um dos quatro métodos: inspeção, análise, demonstração ou teste.

Wieggers (2003), complementa as características descritas acima acrescentando:

- O requisito deve ser correto: O requisito deve descrever exatamente a funcionalidade a ser construída.
- O requisito deve ser priorizado: Deve-se atribuir uma prioridade de execução para cada requisito, para indicar essencialidade do mesmo para uma dada versão do produto.
- O requisito deve ser modificável: A declaração do requisito deve ser flexível; deste modo, deve possibilitar revisões quando necessário e fornecer o histórico de modificações.
- O requisito deve ser rastreável: Os requisitos devem manter uma ligação com suas origens, tal como documentos e necessidades, para que possibilite a verificação da origem do requisito, caso necessário.

2.4 Problemas relacionados a documentação e gestão de requisitos

Nessa seção descreve-se os problemas que foram encontrados no levantamento realizado. Esses problemas foram relatados pelos autores Alexander e Stevens (2002), Firesmith (2007), Wiegers (2003) Verner et al. (2006) e Hull, Jackson e Dick (2005).

2.4.1 Documentação de requisitos

Os problemas relacionados com a documentação de requisitos são:

- **Confusão entre restrições e requisitos** - ocorre quando existe um mal entendimento das definições e diferenças entre restrições e requisitos, não se conseguindo diferenciá-los.
- **Falta de metadados** - consiste na falta de informação descritiva sobre um requisito, para informar o status e a prioridade de cada requisito. A não contemplação deste item acarreta em desatualização dos requisitos e influencia na organização do processo de desenvolvimento, causando confusão, trabalho duplicado (duas pessoas contemplando o mesmo requisito), tempo desperdiçado, entre outros.
- **Falta de identificação** - consiste em não identificar os requisitos através do uso dos códigos, números, entre outros.
- **Fontes dos requisitos não documentadas** - consiste em não armazenar as informações sobre a origem do requisito, como por exemplo, quem solicitou, de qual documento foi retirado e assim por diante.
- **Perda de informação dos *stakeholders*** - consiste em não anotar, esquecer, descartar informações ou documentos que os *stakeholders* forneceram para base dos requisitos.
- **Perda de Requisitos** - consiste em perder os requisitos, seja ocasionado por apagar o requisito consolidado e não ter maneiras de recuperá-lo ou por perda de informação.
- **Informações duplicadas** - consiste na declaração ou escrita da mesma informação duas vezes, no mesmo documento.
- **Desatualização** - consiste em não atualizar os requisitos quando ocorrem alterações.

2.4.2 Gerenciamento de requisitos

Os problemas relacionados com o gerenciamento de requisitos são:

- **Requisitos não rastreados** - consiste no não armazenamento de informações importantes para os requisitos, como as fontes dos requisitos e a não alocação dos requisitos em seus elementos de design, arquitetura e conjunto de testes.
- **Falta de priorização** - Não realizar priorização dos requisitos.
- **Falta de classificação dos requisitos** - consiste em não classificar os requisitos em requisito funcional, não funcional ou de domínio.
- **Descartar informações erroneamente** - consiste em apagar informações importantes relacionadas ao requisito.
- **Requisitos mal organizados** - consiste na falta de organização dos requisitos em categorias, módulos, por prioridades, entre outros. Está relacionado também com a falta de um ambiente apropriado para o gerenciamento dos mesmos.
- **Requisitos mal gerenciados** - consiste na falta de documentação, histórico de modificações, relatórios do andamento do estado do requisito. Essa falta de gerenciamento ocasiona retrabalho, atrasos de prazos, entre outros transtornos.
- **Perda da fonte dos requisitos** - consiste em não documentar quem solicitou o requisito ou de onde foi retirado o mesmo.

3 DESENVOLVIMENTO DO SISTEMA

Após o levantamento dos problemas nas etapas de gestão e documentação de requisitos, fez-se: a definição dos problemas a serem tratados; a modelagem e a implementação do sistema; e o mapeamento entre os problemas e as funcionalidades utilizadas no sistema para prevenir esses problemas. Com isso, deu-se início o desenvolvimento do sistema proposto, que está dividido nas etapas descritas nas subseções 3.1, 3.2, 3.3, 3.4, 3.5. O sistema foi implementado utilizando as linguagens HTML, PHP e para formatação o CSS (FREEMAN; FREEMAN, 2008), (BEIGHLEY; MORRISON, 2010) e (OLIVIERO, 2010).

3.1 Problemas tratados pelo sistema

Após a análise dos problemas relacionados à gestão e documentação de requisitos definiu-se quais problemas seriam tratados no sistema proposto, são eles: **(1)** Falta de metadados, **(2)** Falta de identificação, **(3)** Fontes dos requisitos não documentadas, **(4)** Perda de Requisitos, **(5)** Requisitos não rastreados, **(6)** Falta de priorização, **(7)** Falta de classificação dos requisitos, **(8)** Requisitos mal organizados, **(9)** Requisitos mal gerenciados e **(10)** Perda da fonte dos requisitos.

Os outros problemas não serão tratados pois alguns não se consegue tratar com recursos computacionais, como confusão entre restrições e requisitos, perda de informação dos *stakeholders* e desatualização e o problema informações duplicadas, apesar de ser possível, fugiria do escopo por ser muito específico por isso optou-se em deixá-los para trabalhos futuros.

O problema descartar informações erroneamente está sendo tratado parcialmente com o histórico de modificações, no entanto faz-se necessário tratar vários outros itens para prevenir por completo esse problema, assim essa outra parte foi deixada para trabalhos futuros também.

Com o intuito de tentar prevenir o problema confusão entre restrições e requisitos, colocou-se um *link* para um texto informativo sobre o assunto no sistema, como pode ser observado na Figura 4.

Nesta página tem-se também *links* para outros textos interessantes que podem auxiliar na documentação adequada do requisito. O usuário poderá se informar sobre o formato adequado de um requisito e as características desejáveis a uma boa declaração de requisito.



Orientações Gerais

Nesta página você encontra orientações sobre diversos assuntos relacionados a escrita adequada de requisitos.

Artigos:

- Qual a diferença entre Requisitos e Restrições?
- Características desejáveis a uma declaração de requisitos
- Formato adequado para especificação formal de requisitos

Figura 4 – Documentos para esclarecimento sobre a diferença entre Restrições e Requisitos.

3.2 Modelagem do Banco de Dados

Nesta etapa fez-se a modelagem do banco de dados (BD) do sistema, deste modo criou-se o MER – Modelo Entidade e Relacionamento apresentado na Figura 5 e o MRel – Modelo Relacional do banco de dados apresentado na Figura 6. O Sistema de Gerenciamento de Banco de Dados (SGBD) utilizado neste trabalho foi o MySQL.

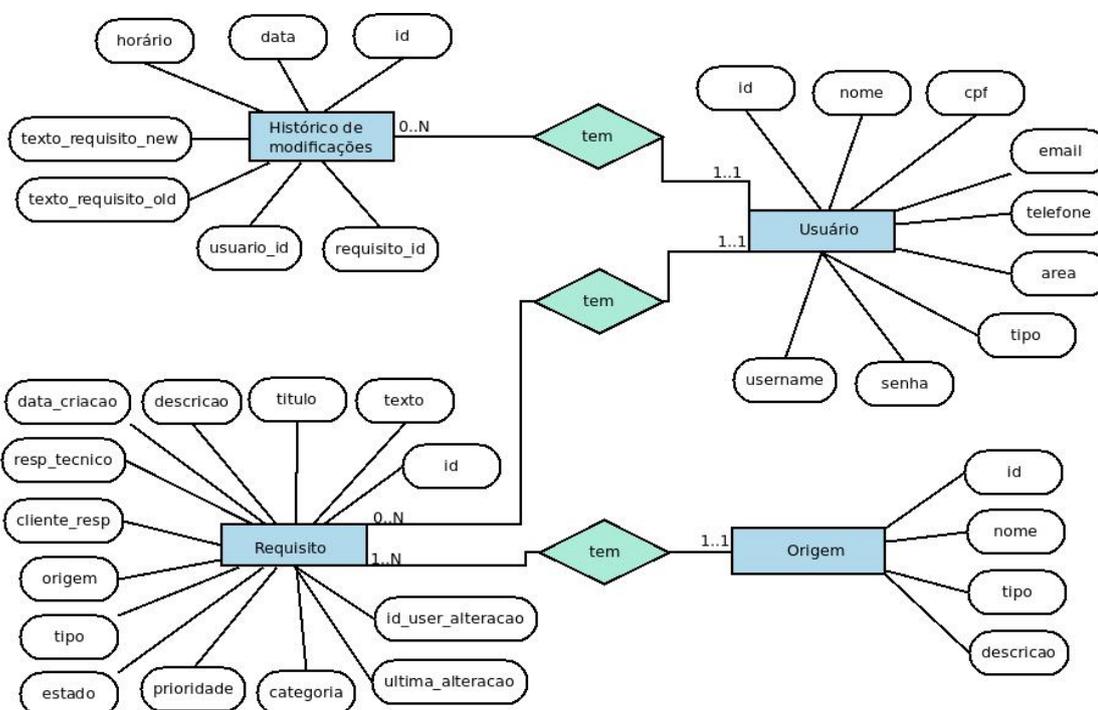


Figura 5 – Modelo Entidade e Relacionamento do BD do sistema.

No modelo entidade e relacionamento foram mapeadas as entidades que serão criadas no BD com seus respectivos atributos, mostrando o relacionamento entre as entidades e a cardinalidade desse relacionamento.

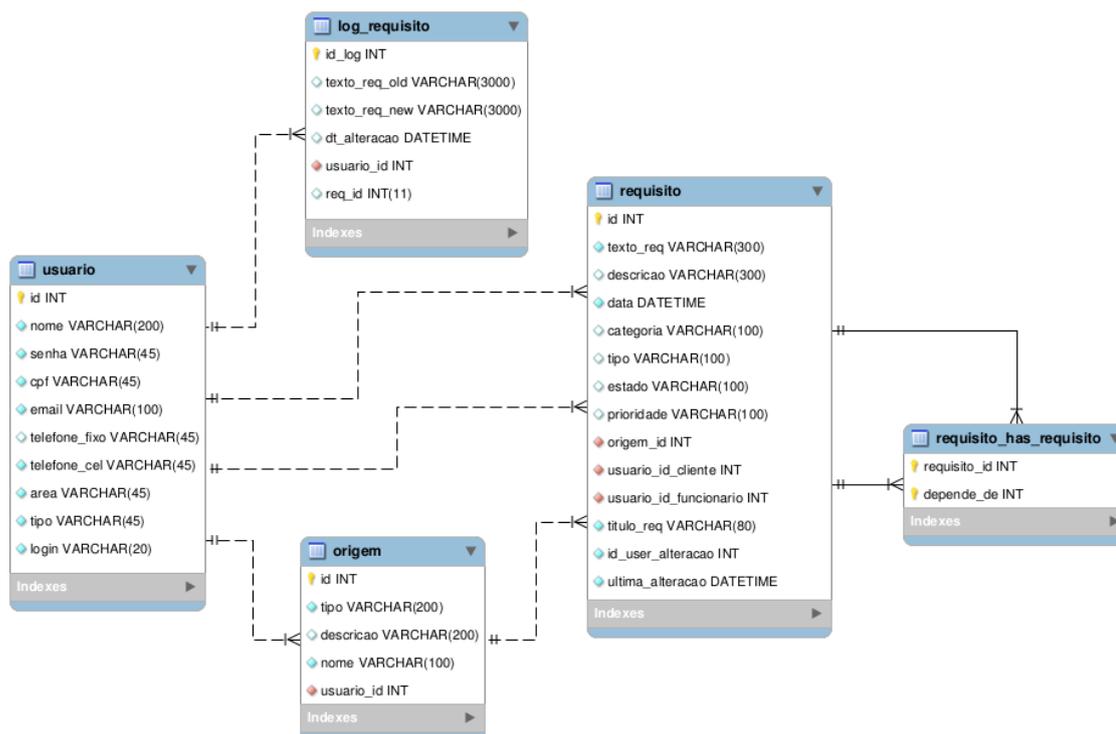


Figura 6 – Modelo relacional do BD do sistema.

No MRel, tem-se as tabelas / relações do BD juntamente com os atributos, respectivas tipagens, relacionamentos entre as tabelas / relações e a cardinalidade desses relacionamentos.

3.3 Funcionamento do sistema

O sistema proposto, denominado DoMaR, tem como objetivo auxiliar no registro, organização e monitoramento de requisitos com o intuito de tentar prevenir os problemas elencados na seção 3.1.

A interação entre usuário e o sistema pode ser visualizada na Figura 7.



Figura 7 – Interação entre usuário e o sistema proposto.

Fonte: Figura produzida pelo próprio autor.

Para o usuário utilizar o sistema ele deve primeiramente realizar seu cadastro e após efetuar o *login*. Quando logado, o usuário pode escolher dentre as seguintes atividades: (1) cadastrar origem e/ou requisito; (2) alterar informações do usuário, origem e/ou requisito; (3) rastrear o requisito; (4) consultar os requisitos cadastrados; e (5) visualizar o histórico de modificações dos requisitos.

3.4 Implementação do sistema DoMaR

Nesta etapa fez-se a implementação do sistema proposto. Para o desenvolvimento utilizou-se as linguagens HTML, PHP e CSS (FREEMAN; FREEMAN, 2008), (BEIGHLEY; MORRISON, 2010) e (OLIVIERO, 2010).

As telas do sistema proposto e a descrição de cada recurso podem ser observadas nas subseções descritas a seguir. A subseção 3.4.1, mostra as telas iniciais do sistema; Na subseção 3.4.2, apresenta as telas de cadastro de origem e de requisito; A subseção 3.4.3, descreve as alterações possíveis; Na subseção 3.4.4, mostra as telas de mapeamento de requisitos; Na subseção 3.4.5, relata as consultas possíveis; e a subseção 3.4.6, que apresenta o histórico de modificações.

3.4.1 Páginas iniciais

As telas iniciais do sistema proposto podem ser observadas nas Figuras 8 e 9. Na Figura 10, tem-se a tela de cadastro de usuários;

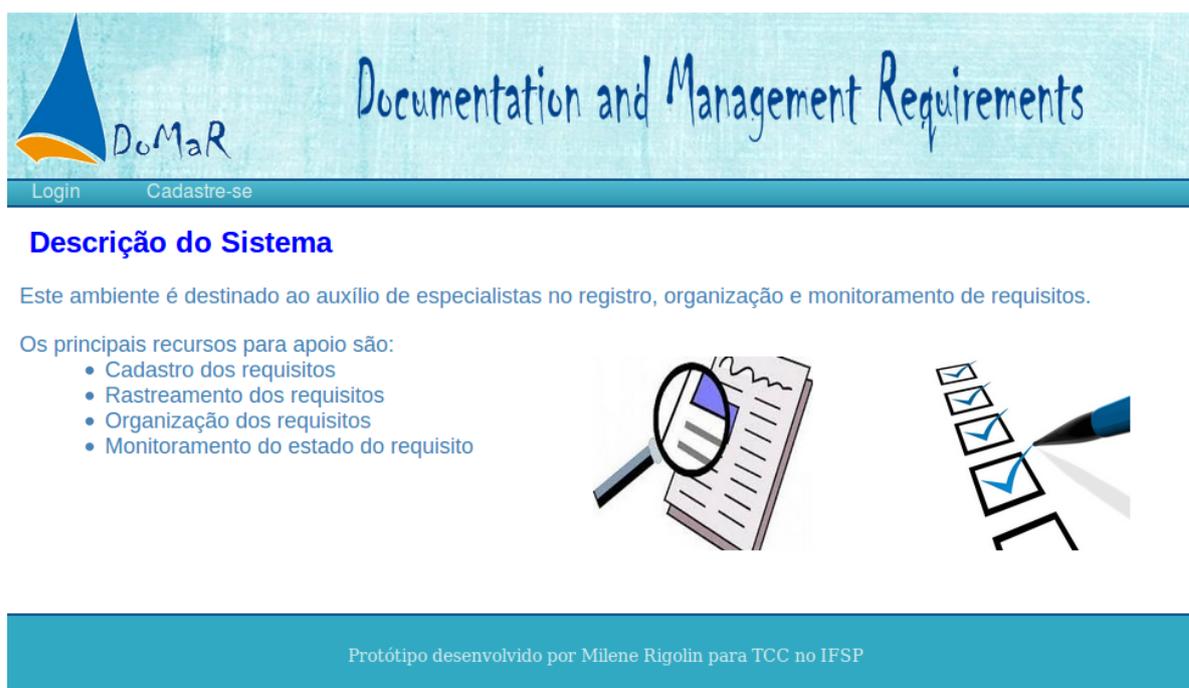


Figura 8 – Tela inicial do sistema proposto.

Na Figura 8, tem-se a descrição do sistema e as principais funcionalidades de apoio

que o sistema fornece. No menu tem-se um *link* para a página de *login* e para página de cadastro do usuário.

Na Figura 9, tem-se a tela de *login* do sistema. Essa tela possui campos usuário e senha.

Caso o usuário não tenha cadastro pode-se clicar no *link* “Cadastre-se” e efetuar o cadastro. Esta página pode ser visualizada na Figura 10.

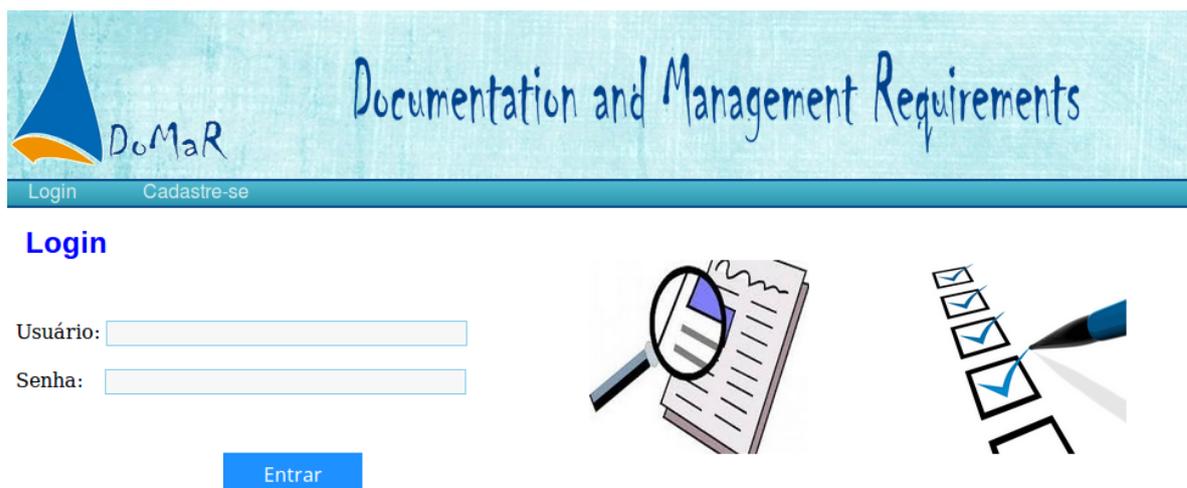


Figura 9 – Tela de *login* do sistema proposto.

Na Figura 10, tem-se os campos para o cadastro do usuário. Ao se cadastrar o usuário fornece informações essenciais para realizarmos a rastreabilidade do requisito que ele irá documentar ou solicitar. As informações são: tipo, usuário, senha, nome, CPF, área de atuação, e-mail, telefones fixo e celular.

Cadastro de Usuário

Tipo: Funcionário Cliente

User Name: Senha:

Nome: CPF:

Área Atuação: E-Mail:

Telefone fixo: Celular:

Figura 10 – Tela de cadastro de usuário.

Quando o usuário já está logado no sistema ele possui acesso ao menu com todos os funcionalidades do sistema.

3.4.2 Cadastro de dados

Com o recurso de cadastro de dados, o Engenheiro de Requisitos pode efetuar o cadastro de documentos considerados como origem do requisito e o cadastro de requisitos.

Na Figura 11, pode-se observar a tela de cadastro de origem do requisito, que pode ser um documento, norma, padrão, lei, entre outros.

INÍCIO CADASTRAR DADOS ALTERAR DADOS RASTREAR REQUISITOS CONSULTAR REQUISITOS HISTÓRICO MODIFICAÇÕES

Olá Milene Sair

Cadastro de Origem

Nome: Tipo:

Descrição:

Finalizar Cadastro Limpar

Figura 11 – Tela de cadastro de origem do requisito.

Na Figura 12, pode-se observar a tela de cadastro dos requisitos.

Cadastro de Requisitos

Título do Requisito:

Requisito:

Descrição:

Detalhes Técnicos

Responsável: Milene Origem: ISO 9001

Cliente responsável: Moacir Categoria do requisito: Interface

Tipo do Requisito:
 Funcional
 Não Funcional
 Domínio

Estado:
 Incompleto
 Completo
 Revisão

Prioridade:
 Essencial
 Importante
 Desejável

Finalizar Cadastro Limpar

Figura 12 – Tela de cadastro de requisito.

Nesta tela, vide Figura 12, tem-se informações essenciais sobre o requisito, tais como: o responsável técnico do requisito, o cliente responsável, o documento de origem do requisito, o tipo do requisito, o estado do texto do requisito e a prioridade do requisito que está sendo documentado.

3.4.3 Alteração de Dados

Uma outra funcionalidade do sistema é a alteração dos dados já cadastrados, deste modo pode-se alterar e deletar: o cadastro de usuário, requisitos e origem. Na Figura 13 pode-se observar um exemplo dessas telas.

Documentos Cadastrados

ID	Nome	Tipo	Descrição	Editar
1	ISO 9001	documento	norma1	 
2	Relatorio Tecnico - Projeto ANA	relatorio	relatorio projeto ana	 
8	MPSBR1	documento	Escreva a descrição do documento aqui	 
9	Relatório 002	Relatório	Relatório técnico	 

Figura 13 – Tela de alteração de origem do requisito.

3.4.4 Mapeamento de Requisitos

Na Figura 14 tem-se a lista de requisitos cadastrados prontos para serem mapeados. O mapeamento realizado no sistema é apenas entre requisitos dependentes.



Selecione o requisito mapeado a ser alterado

ID	Título	Requisito	Mapear
1	Comunicacao segura	O sistema deve prover criptografia de dados na comunicação entre máquinas	
2	Controle de acesso	O sistema deve prover controle de acesso por usuário, por arquivo.	
3	Senha segura	O sistema deve exigir senhas com determinadas regras.	
4	Integridade dos dados	O sistema deve prover garantia de integridade dos dados via hash.	
6	cadastro de usuario	O sistema deverá propiciar o cadastro de usuário.	
7	Visualizar cadastro	O sistema deverá propiciar a visualização do cadastro do usuário212121	
8	Cadastro de requisitos	Propiciar o cadastro de requisitos	

Figura 14 – Tela de requisitos cadastrados a serem mapeados.

Na tela de mapeamento, Figura 15, tem-se o requisito selecionado e os requisitos que podem ser adicionados como dependentes.

Mapear Requisito

ID:

Título do Requisito:

Requisito:

Altere a(s) dependência(s) do requisito

X	ID	Título	Requisito
<input checked="" type="checkbox"/>	1	Comunicacao segura	O sistema deve prover criptografia de dados na comunicação entre máquinas
<input type="checkbox"/>	2	Controle de acesso	O sistema deve prover controle de acesso por usuário, por arquivo.
<input type="checkbox"/>	4	Integridade dos dados	O sistema deve prover garantia de integridade dos dados via hash.
<input type="checkbox"/>	6	cadastro de usuario	O sistema deverá propiciar o cadastro de usuário.
<input type="checkbox"/>	7	Visualizar cadastro	O sistema deverá propiciar a visualização do cadastro do usuário212121
<input type="checkbox"/>	8	Cadastro de requisitos	Propiciar o cadastro de requisitos
<input type="checkbox"/>	9	Visualizar o mapeamento de requisitos	Propiciar a visualização do mapeamento entre requisitos

Figura 15 – Tela de mapeamento entre requisitos.

Na Figura 16, tem-se a matriz de mapeamento. Ela mostra o requisito e suas dependências. Quando dois requisitos são dependentes entre si o símbolo se modifica.



Visualização do Mapeamento entre requisitos

R/D	1	2	3	4	6	7	8	9
1		-	-	-	-	-	-	-
2	-		-	-	-	-	-	-
3	↗	-		-	-	-	-	-
4	-	↗	-		-	-	-	-
6	-	-	-	-		↻	-	-
7	-	-	-	-	↻		-	-
8	-	-	-	-	↗	-		-
9	-	-	-	-	↗	-	↗	

Propiciar a visualização do mapeamento entre requisitos

Figura 16 – Matriz de mapeamento entre requisitos.

3.4.5 Consulta de Requisitos

Com o intuito de auxiliar na organização dos requisitos fez-se páginas para consultas dos requisitos por: cliente, funcionário responsável, origem, categoria, tipo, estado, prioridade e pode-se também combinar esses fatores. Essas consultas podem ser observadas na Figura 17.



Figura 17 – Tela de consulta de requisitos.

3.4.6 Histórico de Modificações

O sistema possui também um histórico de modificações onde as modificações no texto do requisito são armazenadas juntamente com o nome do responsável pela alteração, data e hora da modificação. Quando um requisito é deletado também guarda-se, inserindo no campo de novo requisito a informação de que o requisito foi deletado.



Requisitos Alterados

ID	Requisito Antigo	Requisito Novo	Usuário	Data
7	O sistema deverá propiciar a visualização do cadastro do usuário	O sistema deverá propiciar a visualização do cadastro do usuário212121	Eduardo	2015-09-18 14:32:24

Figura 18 – Tela do histórico de modificações gerais.

A tela do histórico de modificações pode ser observada na Figura 18. Pode-se consultar o histórico de três maneiras: (1) no geral, onde pode-se ver todas as alterações

realizadas por usuários do sistema; (2) por cliente, onde se seleciona o cliente desejado; e (3) por responsável técnico, onde se seleciona o responsável desejado.

3.5 Mapeamento entre os Problemas e as funcionalidades do sistema proposto

Nesta seção apresenta-se o mapeamento realizado entre os problemas que são tratados na pesquisa e as funcionalidades que foram criadas para tentar prevenir os mesmos.

3.5.1 Problemas relacionados com a documentação de requisitos

As funcionalidades criadas para tentar prevenir os problemas relacionados com a documentação de requisitos estão organizados por problemas, são elas:

- **Falta de metadados:** esse problema foi tratado inserindo os campos status, prioridade e responsável pelo requisito ao realizar o cadastro do requisito.
- **Falta de identificação:** esse problema foi tratado inserindo uma numeração única, que é gerada automaticamente pelo banco de dados ao cadastrar os requisitos.
- **Fontes dos requisitos não documentadas:** com o intuito de tratar este problema criou-se os campos para inserção do nome da fonte em que o requisito foi retirado e o campo para marcar o cliente responsável pelo requisito.
- **Perda de Requisitos:** este problema está sendo tratado através de um histórico que armazena as modificações dos requisitos.

3.5.2 Problemas relacionados com o gerenciamento de requisitos

As funcionalidades criadas para tentar prevenir os problemas relacionados com a gestão de requisitos estão organizados por problemas, são elas:

- **Requisitos não rastreados:** foi criada uma página para realizar o rastreamento entre os requisitos e também se criou os campos de seleção da origem e categoria do requisito.
- **Falta de priorização:** visando a prevenção desse problema criou-se um campo para selecionar a prioridade do requisito cadastrado.
- **Falta de classificação dos requisitos:** foi criado um campo para possibilitar a marcação do tipo do requisito, com o intuito de prevenir esse problema.

- **Requisitos mal organizados:** foi criada categorias para melhor organização dos requisitos. Foram criados também filtros para visualização dos requisitos por categoria.
- **Requisitos mal gerenciados:** com o intuito de prevenir este problema, criou-se filtro para visualizar o requisito por estado e também criou-se uma página com o histórico de modificações do requisito.
- **Perda da fonte dos requisitos:** para prevenir esse problema criou-se os campos para documentação das fontes dos requisitos, tanto o documento quanto o cliente responsável.

4 ESTUDO DE CASO

O estudo de caso consistiu na aplicação do sistema em um cenário que simula atividades rotineiras de gestão e documentação de requisitos.

O objetivo principal com o estudo de caso é verificar se as funcionalidades implementadas no sistema proposto podem prevenir os problemas levantados.

Para a organização do estudo de caso, fez-se as seguintes atividades: (1) Estruturação do estudo de caso; (2) Preparação do questionário; (3) Definição do perfil do especialista; (4) Seleção dos especialistas; e (5) Execução da avaliação.

Na atividade de estruturação do estudo de caso, definiu-se o roteiro e etapas da avaliação. Esse roteiro pode ser visualizado no Apêndice A.

Na atividade de preparação do questionário, elaborou-se um questionário de avaliação dos problemas a serem prevenidos pelas funcionalidades do sistema proposto. Esse questionário pode ser visualizado no Apêndice B.

Para utilizar e avaliar o sistema foram selecionados onze especialistas. Os especialistas tinham o seguinte perfil:

- Com relação ao conhecimento na área da ER:
 - 45,5% possuem conhecimento médio;
 - 27,3% possuem conhecimento avançado;
 - 27,3% possuem conhecimento básico;
- Com relação ao porte de projetos trabalhados, em média:
 - 72,7% porte médio;
 - 18,2% porte grande;
 - 9,1% porte pequeno;
- No total, os especialistas trabalharam em 168 projetos de desenvolvimento de *software*, dando em média 15 projetos por especialista.

O número de especialistas foi limitado em onze pois eram os especialistas disponíveis no momento.

A execução da avaliação foi organizada em oito tarefas: (1) Cadastro e *login*; (2) Cadastro de quatro requisitos; (3) Alteração do texto de um dos requisitos cadastrados;

(4) Realização do mapeamento entre requisitos; (5) Visualização da matriz de mapeamento; (6) Exploração dos filtros de pesquisa dos requisitos cadastrados; (7) Verificação do histórico de alteração de requisitos; e (8) Avaliação das funcionalidades do sistema.

4.1 Avaliação

Após o uso do sistema, os especialistas analisaram se as funcionalidades do sistema podem prevenir os problemas relacionados a gestão e documentação dos requisitos. Para a análise e avaliação das funcionalidades em relação a prevenção dos problemas utilizou-se a Escala *Likert* (LIKERT, 1932), com 4 fatores. São eles: 4- muito alto; 3- alto; 2- baixo; 1- muito baixo. Desse modo, no questionário tinha-se a lista de problemas relacionados e a escala de avaliação.

4.2 Resultados

Os resultados obtidos com a avaliação podem ser observados nas Figuras 19 e 20.

Especialistas / Problemas	Problemas relacionados à Documentação de Requisitos				Problemas relacionados à Gestão de Requisitos						
	Falta de metadados	Falta de identificação	Fontes dos requisitos não documentadas	Perda de Requisitos	Requisitos não rastreados	Falta de priorização	Falta de classificação dos requisitos	Requisitos mal organizados	Requisitos mal gerenciados	Perda da fonte dos requisitos	
Especialista 1	3	4	4	4	3	4	4	4	4	4	
Especialista 2	4	4	4	3	3	2	4	3	4	4	
Especialista 3	3	4	4	3	4	3	4	3	3	4	
Especialista 4	3	4	4	3	4	3	4	4	3	4	
Especialista 5	4	3	4	4	4	4	4	4	4	2	
Especialista 6	4	4	4	4	4	4	4	4	4	4	
Especialista 7	2	4	2	2	3	3	4	4	3	1	
Especialista 8	4	3	4	4	3	4	4	2	4	4	
Especialista 9	3	4	4	4	3	4	4	4	4	4	
Especialista 10	3	4	4	4	3	4	4	4	3	4	
Especialista 11	4	4	4	4	4	4	4	4	4	4	
Média	3,4	3,8	3,8	3,5	3,5	3,5	4,0	3,6	3,6	3,5	

Figura 19 – Resultados da avaliação dos especialistas.

Com base nos resultados obtidos, pode-se concluir que, na média, as funcionalidades implementadas pelo sistema proposto conseguiram prevenir os problemas levantados. Pode-se perceber também que os problemas que estão com a média entre 3,0 e 3,5 precisam ser melhorados.

Todos os problemas foram analisados utilizando a Escala *Likert* (LIKERT, 1932), onde os problemas que obtiveram as médias maiores na avaliação foram: Falta de classificação dos requisitos, Fontes dos requisitos não documentadas e Falta de identificação e o problema que obteve a média mais baixa na avaliação foi: Falta de metadados, porém apesar desse problema ter tido a menor média, o mesmo ainda está dentro do fator 3-alto.

Os especialistas, que participaram do Estudo de Caso, sugeriram colocar uma escala maior para priorização do requisito, como por exemplo adotar uma escala de 0 a 100, para melhorar o problema Falta de metadados.

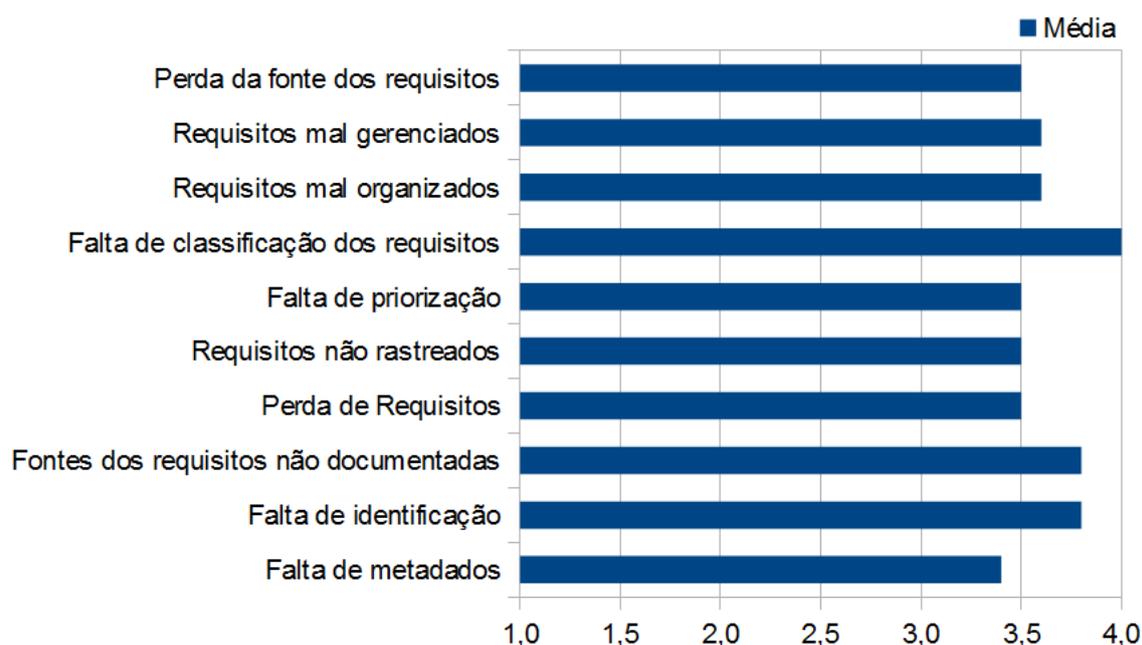


Figura 20 – Médias dos resultados do Estudo de Caso.

A inserção da escala para melhoramento do problema Falta de metadados foi deixada para trabalhos futuros. Ressalta-se ainda que seria interessante também o estudo mais aprofundado em técnicas de priorização de requisitos, com o intuito de verificar qual seria a melhor abordagem para se tratar esse problema.

4.3 Sugestões dos Especialistas

Os especialistas sugeriram modificações e melhorias para o sistema, essas sugestões foram classificadas em categorias, de acordo com as funcionalidades implementadas, pois nem todas as sugestões se enquadram nos problemas levantados.

- Histórico de modificações:
 - sugeriu-se a visualização das alterações nos detalhes técnicos do requisito;
e
 - o registro do histórico de mapeamentos.
- Priorização de requisitos:
 - sugeriu-se oferecer uma escala maior para priorização, como por exemplo adotar uma escala de 0 a 100.
- Mapeamento de requisitos:

- sugeriram tratar a consistência em relação a definição de relacionamentos entre requisitos, principalmente requisitos com dependência, sub-requisitos, requisitos complementares, entre outros; e
 - gerar facilidades para identificar dependência transitiva entre as funcionalidades.
- Consulta de requisitos:
 - sugeriram ter um link para alterar o requisito ao ser consultado;
 - ter árvore de dependência entre fontes dos requisitos;
 - consultar requisitos deletados; e
 - ao consultar o requisito ter um link para visualizar dados do mesmo e vir com o histórico de modificações.
 - Sugestões adicionais:
 - ter um acompanhamento de status de progresso dos requisitos;
 - colocar os tipos de requisitos conforme normas da IEEE;
 - quando o sistema for utilizado em projetos de segurança, inserir atributos da ISO 15408;
 - inserir o documento ou link do documento de origem do requisito;
 - ter um link para cadastrar a origem na tela de cadastro de requisitos; e
 - apagar no menu o link de mapeamento e manter só o de alterar mapeamento, já que o mesmo carrega os requisitos dependentes.

5 CONCLUSÃO

Neste capítulo apresentam-se as conclusões obtidas com a finalização do trabalho, as limitações e as sugestões de trabalhos futuros.

5.1 Conclusões obtidas

Este trabalho teve como foco a investigação da área de ER, verificando os problemas relacionados a gestão e documentação de requisitos, com o intuito de mapear esses problemas e propor um sistema para auxiliar no registro, organização e monitoramento de requisitos visando prevenir tais problemas.

Notou-se que alguns problemas mapeados não se resolviam através de recursos computacionais ou fugiriam do escopo, deste modo teve-se que realizar uma triagem dos problemas para selecionar os que seriam viáveis tratar no sistema proposto.

Para o desenvolvimento do trabalho realizou-se os seguintes passos: (1) Revisão bibliográfica no contexto da ER; (2) Verificação de trabalhos correlatos; (3) Levantamento dos problemas das etapas de gestão e documentação de requisitos; (4) Construção de um sistema que dê suporte ao processo da ER, para isso utilizou-se as linguagens HTML, PHP e CSS; e (5) Aplicação do sistema DoMaR em um estudo de caso.

Em relação aos trabalhos correlatos notou-se que existem diversas abordagens e sistemas para auxiliar as atividades de ER, porém a maioria não possui o sistema integrado entre a documentação dos requisitos, mapeamento e gestão. Existem apenas uma das funcionalidades por exemplo: sistemas para realizar o mapeamento; sistemas para documentação; sistemas para gestão e o sistema proposto faz essa integração.

O estudo de caso foi realizado com onze especialistas da área da computação que tinham experiências em projetos com a ER. Os especialistas utilizaram o sistema proposto e depois avaliaram se as funcionalidades disponíveis no sistema poderiam prevenir os problemas elencados.

Com base nas avaliações dos especialistas pode-se concluir que, na média, o sistema pode prevenir os problemas elencados. Porém notou-se que os problemas que ficaram com a média entre 3,0 e 3,5 precisam ser melhorados.

As sugestões dos especialistas foram deixadas como sugestões para trabalhos futuros.

5.2 Dificuldades

A maior dificuldade para a realização do trabalho foi encontrar um número significativo de especialistas para avaliar as funcionalidades do sistema, uma vez que o estudo de caso precisava ser feito presencialmente.

5.3 Trabalhos futuros

Como sugestão de trabalhos futuros tem-se:

- Implementar as sugestões dadas pelos especialistas no estudo de caso;
- Após a implementação das sugestões, reavaliar o sistema, com o intuito verificar o impacto das sugestões na prevenção dos problemas levantados;
- Realizar testes de usabilidade para melhorar a interface do sistema;
- Verificar a possibilidade de adaptar o sistema para utilização em metodologias ágeis; e
- Utilizar o sistema em um projeto real para verificar suas limitações e propor melhorias.

REFERÊNCIAS

- ALEXANDER, I.; STEVENS, R. *Writing better requirements*. London: Pearson Education, 2002.
- ARTEIRO, I. C. J. B. Melhoria de processo de engenharia de requisitos em empresas de mercado através de transferência de tecnologia. 2014.
- BEIGHLEY, L.; MORRISON, M. *Use A Cabeça! PHP e MySQL*. Rio de Janeiro: Alta Books, 2010.
- BOEHM, W. B.; PAPACCIO, P. N. *Understanding and controlling software costs*. *IEEE Transactions on Software Engineering*, v. 11, 1988.
- ELECTRICAL, I. O.; ENGINEERS, E. *Standard glossary of software engineering terminology*. Los Alamitos, CA, 1990.
- FIRESMITH, D. *Common requirements problems, their negative consequences, and industry best practices to help solve them*. *Journal of Object Technology*, v. 06, n. 1, 2007.
- FREEMAN, E.; FREEMAN, E. *Use a Cabeça! HTML com CSS e XHTML*. Rio de Janeiro: Alta Books, 2008.
- GENVIGIR, E. C. *Um modelo para rastreabilidade de requisitos de software baseado em generalização de elos e atributos*. Tese (Doutorado) — INPE, 2009.
- GRANDE, J. I. D.; MARTINS, L. E. G. Sigerar: Uma ferramenta para gerenciamento de requisitos. *Workshop em Engenharia de Requisitos*, 2006.
- GROUP, S. *Relatório: CHAOS report 2012*. Boston, 2012.
- HULL, E.; JACKSON, K.; DICK, J. *Requirements engineering*. 2. ed. London: Springer, 2005.
- KAR, P.; BAILEY, M. *Characteristics of good requirements*. In: *INTERNATIONAL SYMPOSIUM OF THE NATIONAL COUNCIL ON SYSTEMS ENGINEERING*, Boston, v. 6, 1996.
- KOTONYA, G.; SOMMERVILLE, I. *Requirements engineering: processes and techniques*. Chichester: John Wiley and Sons, 2000.
- LEFFINGWELL, D. *Calculating the return on investment from more effective requirements management*. *Cutter IT Journal*, v. 10, 1997.
- LIKERT, R. *A technique for the measurement of attitudes*. *Archives of Psychology*, v. 140, 1932.
- LOPES, M. E. R. F. *Aplicação de Teorias do Erro Humano no Processo de Engenharia de Requisitos*. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, 2011.
- NASA. *Manager's Handbook for Software Development: software engineering laboratory series – SEL-84-101*. USA: NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, 1990.

- OLIVIERO, C. *PHP 5 com MySQL - Comércio Eletrônico*. São Paulo: Editora Érica, 2010.
- PONTES, I. de S. Ferramenta de apoio à rastreabilidade de requisitos de *Software*. 2006.
- SALES, M. F. et al. Uma ferramenta de apoio à gerência de requisitos integrada a um ambiente de desenvolvimento de *Software* centrado em processos. *CBSOft*, 2010.
- SANT'ANNA, N. *Um Ambiente Integrado para o Apoio ao Desenvolvimento e Gestão de Projetos de Software para Sistemas de Controle de Satélites*. Tese (Doutorado) — INPE, 2000.
- SAYAO, M.; BREITMAN, K. K. Gerência de requisitos. 2009.
- SOMMERVILLE, I. *Engenharia de software*. 8. ed. São Paulo: Pearson Education, 2007.
- VERNER, J. et al. *Predicting good requirements for in-house development projects*. In: *INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING*, New York, 2006.
- WIEGERS, K. E. *Software requirements*. Washington, DC: Microsoft Press, 2003.
- YOUNG, R. R. *The requirements engineering handbook*. London: Artech House, 2004.

APÊNDICE A – DOCUMENTOS DO ESTUDO DE CASO

A.1 Utilização do sistema

Etapa 1: realize o login e cadastre os quatro requisitos no sistema.

1. O sistema deverá propiciar o cadastro do usuário.
2. O sistema deverá ser capaz de propiciar a visualização do cadastro do usuário após o mesmo estar logado.
3. O sistema deverá ser capaz de propiciar o cadastro de requisitos, após o usuário estar logado.
4. O sistema deverá ser capaz de propiciar a visualização do mapeamento entre requisitos.

Nas informações técnicas marque as informações que achar pertinente.

Etapa 2: altere algum texto do requisito ou delete algum requisito.

Etapa 3: faça o mapeamento entre requisitos, colocando os requisitos de 2 a 4 como dependentes do requisito 1 e o requisito 4 dependente do requisito 3.

Etapa 4: visualize o mapeamento entre os requisitos.

Etapa 5: explore os filtros do sistema.

Etapa 6: visualize o histórico de alterações do requisito.

A.2 Avaliação do sistema

Avaliar as funcionalidades do sistema através do questionário. O questionário pode ser visualizado no Apêndice B.

APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DO SISTEMA

B.1 Caracterização do Participante

1. Qual o seu nível de conhecimento na área de Engenharia de Requisitos?

Básico Médio Avançado

2. Em média, quantos projetos trabalhou?

3. Em média, qual o porte dos projetos trabalhados?

Básico Médio Avançado

Informações Pessoais Nome: Email:

B.2 Avaliação das Funcionalidades

Verificar se as funcionalidades podem prevenir problemas da ER. Julgue cada problema em uma escala de 1 a 4.

Em sua opinião podemos prevenir os problemas utilizando as funcionalidades implementadas?

B.2.1 Problemas relacionados com a documentação de requisitos

Falta de metadados

consiste na falta de informação descritiva sobre um requisito, para informar o status e a prioridade de cada requisito. A não contemplação deste item acarreta em desatualização dos requisitos e influencia na organização do processo de desenvolvimento, causando confusão, trabalho duplicado (duas pessoas contemplando o mesmo requisito), tempo desperdiçado, entre outros.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Falta de identificação

consiste em não identificar os requisitos através do uso dos códigos, números, entre outros. Fontes dos requisitos não documentadas consiste em não armazenar as informações sobre a origem do requisito, como por exemplo, quem solicitou, de qual documento foi retirado e assim por diante.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Fontes dos requisitos não documentadas

consiste em não armazenar as informações sobre a origem do requisito, como por exemplo, quem solicitou, de qual documento foi retirado e assim por diante.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Perda de Requisitos

consiste em perder os requisitos, seja ocasionado por apagar o requisito consolidado e não ter maneiras de recuperá-lo ou por perda de informação.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

B.2.2 Problemas relacionados com a gerenciamento de requisitos**Requisitos não rastreados**

consiste no não armazenamento de informações importantes para os requisitos, como as fontes dos requisitos e a não alocação dos requisitos em seus elementos de design, arquitetura e conjunto de testes.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Falta de priorização

Não realizar priorização dos requisitos.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Falta de classificação dos requisitos

consiste em não classificar os requisitos em requisito funcional, não funcional ou de domínio.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Requisitos mal organizados

consiste na falta de organização dos requisitos em categorias, módulos, por prioridades, entre outros. Está relacionado também com a falta de um ambiente apropriado para o gerenciamento dos mesmos.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Requisitos mal gerenciados

consiste na falta de documentação, histórico de modificações, relatórios do andamento do estado do requisito. Essa falta de gerenciamento ocasiona retrabalho, atrasos de prazos, entre outros transtornos.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Perda da fonte dos requisitos

consiste em não documentar quem solicitou o requisito ou de onde foi retirado o mesmo.

1- muito pouca; 2- pouca; 3- alta; 4- muito alta

Existe algum comentário adicional sobre o sistema? Melhoria, apontamento de defeito.