

Técnicas Baseadas em Refatoração para resolver *Requirements Smells*

Rafael Nascimento¹, Francisco Neto², Fernanda Alencar³, Ricardo Ramos⁴,
Eduardo Aranha¹ e Márcia Lucena¹

**rafael.jullian@gmail.com; miltonmendes@ufersa.edu.br; fernandaalenc@gmail.com;
ricardo.aramos@univasf.edu; eduardo.aranha@ufrn.br; marciaj@dimap.ufrn.br**

¹ Universidade Federal do Rio Grande do Norte, Natal, Rio Grande do Norte, Brasil

² Universidade Federal Rural do Semi-Árido, Mossoró, Rio Grande do Norte, Brasil

³ Universidade Federal de Pernambuco, Recife, Pernambuco, Brasil

⁴ Universidade Federal do Vale do São Francisco, Petrolina, Pernambuco, Brasil

DOI: 10.17013/risti.55.110-123

Resumo: A Especificação de Requisitos de Linguagem Natural é a tipo mais comum de documentação usada no desenvolvimento de sistemas. No entanto, a Linguagem Natural, devido à má escrita, pode gerar ambiguidade e outros defeitos que acabam comprometendo os critérios de qualidade de requisitos como compreensão, clareza, completude e outros. *Requirements Smells* são problemas de escrita na especificação de requisitos de linguagem natural. No entanto, não existem estudos sobre formas sistemáticas de corrigir *Requirements Smells* e garantir a Qualidade dos Requisitos. O objetivo deste trabalho consiste no desenvolvimento de Técnicas de Refatoração para corrigir *Requirements Smells* e garantir os Critérios de Qualidade. As técnicas de refatoração foram desenvolvidas seguindo o modelo proposto por Martin Fowler para refatoração de código. Espera-se que as aplicações das técnicas possam ser eficazes, trazendo diversos benefícios como a melhoria da qualidade.

Palavras-chave: Requisitos em Linguagem Natural, *Requirements Smells*, Refatoração, Requisitos de Qualidade.

Towards Refactoring-Based Techniques to resolve Requirements Smells

Abstract: Natural Language Requirements Specification is the most common type of documentation used in systems development. However, Natural Language, due to poor writing, can generate ambiguity and other defects that end up compromising Quality Criteria of requirements such as understanding, clarity, completeness and others. Requirements Smells are writing problems in the Natural Language Requirements specification. However, there are no studies on systematic ways to correct Requirements Smells and guarantee the Requirements Quality. The objective of this work consists in the development of Refactoring to

correct symptoms of Requirements Smells and guarantee the Quality Criteria. The Refactoring Techniques were developed following the model proposed by Martin Fowler for code refactoring. It is expected that the use of Refactoring Techniques can be effective, bringing several benefits such as the quality improvement.

Keywords: Requirements Natural Language, *Requirements Smells*, Refactoring, Requirement's Quality.

1. Introdução

A Engenharia de Requisitos tem como principal função capturar, documentar e gerenciar requisitos, e demais informações relevantes, que serão utilizados para a construção de sistemas que atendam às necessidades dos clientes (Pohl, 2016; Soto, 2023; Vera et al., 2024). Logo, é imprescindível que os requisitos sejam capturados e documentados da melhor maneira possível, prezando-se pela qualidade das informações.

O padrão de documentação de requisitos mais utilizado é em Linguagem Natural. Todavia, apesar da praticidade de não ter de aprender uma nova notação, ela é susceptível a erros que causam ambiguidade, falta de entendimento e clareza, por exemplo (Pohl, 2016). Os pesquisadores (Femmer et al., 2014, 2017) identificaram padrões de má escrita em Requisitos em Linguagem Natural que comprometem negativamente Critérios de Qualidade e outras atividades no desenvolvimento de software. Analisando estes padrões de má escrita, (Femmer et al., 2014) desenvolveram um catálogo sobre *Requirements Smells*. Nesse catálogo são descritos vários tipos de Smells que afetam Critérios de Qualidade de Requisitos em Linguagem Natural. O catálogo e os Critérios de Qualidade foram desenvolvidos a partir do padrão IEEE/IEC 29148, que descrever uma Taxonomia de Qualidade para Requisitos em Linguagem Natural (ISO/IEC/IEEE 29148, 2018).

Entretanto, apesar da literatura abordar trabalhos sobre a identificação de *Requirements Smells* em diversos projetos de sistemas e vários tipos de artefatos em linguagem natural, não há indícios de trabalhos que investigam técnicas para corrigir problemas de Smells em Requisitos em Linguagem Natural e garantir Critérios de Qualidade. Portanto, baseado nas abordagens de refatoração propostas por (Fowler, 2020; Russo et al., 1998), este trabalho propõem o desenvolvimento de técnicas de refatoração para resolver problemas de *Requirements Smells* em Requisitos em Linguagem Natural e garantir Critérios de Qualidade para diminuir, total ou parcialmente, impactos negativos em atividades que dependem destes artefatos de requisitos. A ideia é que esses mecanismos sirvam de guias para facilitar o trabalho dos praticantes de requisitos em atividades de análise e correção de requisitos.

Este trabalho está organizado da seguinte forma: na seção de 2, são expostos o Referencial Teórico; na seção 3, são descritos os Trabalhos Relacionados usados como base para o desenvolvimento desta pesquisa; na seção 4, são descritos a Metodologia de Pesquisa e as Técnicas de Refatoração desenvolvidas e aplicadas a correção de *Requirements Smells*; na seção 5, são feitas discussões em relação ao aprendizado adquirido no processo de desenvolvimento das Técnicas de Refatoração; na seção 6, são discutidas as limitações que impactaram o desenvolvimento deste trabalho; e na seção 7, são descritas as Considerações Finais sobre o objetivo e a relevância deste trabalho.

2. Referencial Teórico

2.1. Requirements Smells

Segundo (Femmer et al., 2017), os *Requirements Smells* são sintomas de má escrita de Requisitos em Linguagem Natural. Esta escrita inadequada compromete Critérios de Qualidade nos requisitos, que consequentemente podem impactar atividades no desenvolvimento de sistemas que dependem destas informações, como: projeto arquitetural, desenvolvimento de código, planos de testes. Além disso, Henning Femmer investigou a norma *ISO/IEEE/IEC 29148* para desenvolver uma Taxonomia de Critérios de Qualidade para Requisitos em Linguagem Natural, de forma individual e coletiva, servindo de parâmetro para os impactos negativos produzidos na má escrita (*ISO/IEC/IEEE 29148*, 2018). Nesta pesquisa foi utilizado o catálogo de *Requirements Smells* descrito no trabalho de (Nascimento et al., 2018), que fez um Mapeamento Sistemático da Literatura, fazendo uma extensão no catálogo de Henning Femmer:

1. **Adjetivos e Advérbios Ambíguos:** são adjetivos e advérbios que geram ambiguidade na compreensão dos requisitos. Exemplo: Se a qualidade (...) for muito baixa, um registro de falha deve ser armazenado.
2. **Pronomes Vagos:** são pronomes que fazem referência vaga à 3º pessoa do discurso. Exemplo: O software deve implementar serviços para aplicativos, que devem se comunicar com os aplicativos do controlador implantados em outros controladores.
3. **Linguagem Subjetiva:** são palavras cuja semântica não é objetiva. Exemplos: amigável, fácil de usar, econômico.
4. **Adjetivos Comparativos:** são advérbios e adjetivos, onde os requisitos expressam uma relação do sistema com outros sistemas específicos. Exemplo: melhor que, maior qualidade.
5. **Adjetivos Superlativos:** são advérbios e adjetivos, onde os requisitos expressam uma relação do sistema com todos os outros sistemas. Exemplo: melhor desempenho, menor tempo de resposta.
6. **Sentenças Negativas:** são sentenças usadas em funcionalidades que o sistema não deve fornecer, mas sem justificar o motivo. Exemplo: o sistema não deve aceitar cartões de crédito VISA.
7. **Termos Não Verificáveis:** são palavras difíceis de verificar por oferecer várias possibilidades de ações do sistema. Exemplo: O sistema só pode ser ativado se todos os sensores necessários (...) trabalharem com medição suficiente.
8. **Brechas:** são palavras que possibilitam os stakeholders ignorar as especificações. Exemplos: se possível, conforme apropriado, conforme aplicável.
9. **Referências Incompletas:** São referências que o leitores não conseguem encontrar.
10. **Voz Passiva:** Caracterizada por requisitos onde não está claro o ator que está desempenhando uma determinada ação no sistema.
11. **Funcionalidade Duplicada:** São descrições de ações entre sistemas e atores, repetidas em várias requisitos.

2.2. Refatoração de Requisitos

Para (Russo et al., 1998), refatoração de requisitos consiste na reestruturação da escrita de Requisitos em Linguagem Natural para corrigir, de forma sistematizada, problemas que afetam a sua qualidade. Trazendo alguns benefícios, como: encontrar falhas; tornar os requisitos legíveis; melhorar o projeto de software e facilitar a programação (Opdyke, 1992; Ramos et al., 2007; Russo et al., 1998). Assim, (Opdyke, 1992) descreve que a refatoração é composta de duas etapas. Na etapa de pré-condições, são estabelecidas as condições necessárias para que o artefato esteja correto. Na etapa de pós-condições, é verificado se o artefato atingiu a corretude necessária após as transformações aplicadas. Para (Mens & Tourwé, 2004), a refatoração, adaptado para artefatos de requisitos, deve seguir seis etapas distintas:

- **Identificar o/os artefatos para refatoração:** neste casos, são os artefatos de requisitos de software.
- **Determinar as refatorações aplicadas:** aplicar as técnicas mais adequadas para os respectivos problemas de inconsistência identificados;
- **Garantir a qualidade:** estabelecer quais as pré-condições e pós-condições em relação aos Critérios de Qualidade para requisitos (Ramos et al., 2007);
- **Aplicar a refatoração:** que pode ser aplicada de forma manual ou automatizada;
- **Verificar a garantia da qualidade:** observar se os Critérios de Qualidade foram garantidos;

Manter a consistência com outros artefatos: os demais artefatos devem ser atualizados para se manter a consistência com os artefatos de requisitos.

Nesta pesquisa foram considerados somente 5 etapas, pois não fez parte do escopo do nosso trabalho manter a consistência dos artefatos de requisitos com outros artefatos de desenvolvimento de software.

3. Trabalhos Relacionados

Observa-se uma lacuna na literatura no que se refere a trabalhos focados na refatoração de Requisitos em Linguagem Natural afetados por *Requirements Smells*. No entanto, os trabalhos relacionados serviram de norte para compreender os benefícios da refatoração de requisitos e de inspiração para o desenvolvimento das técnicas propostas nesta pesquisa.

No trabalho de (Russo et al., 1998) foram desenvolvidas Técnicas de Refatoração para especificações de Requisitos em Linguagem Natural. Os autores recomendam decompor requisitos em estruturas denominadas pontos de visão, que encapsulam requisitos parciais de componentes de software e suas interações. Tem como benefício melhorar o entendimento, facilitar a detecção de inconsistências e gerenciar a evolução dos requisitos.

No trabalho de (Ramos et al., 2007) foram desenvolvidas Técnicas de Refatoração de Requisitos em Linguagem Natural, baseado em Cenários de Casos de Uso. Eles consideram

que requisitos inadequados, chamados de *Bad Smells*, propagam problemas para as fases subsequentes no desenvolvimento de software, comprometendo a manutenção. As Técnicas de Refatoração foram desenvolvidas baseadas no modelo proposto por (Fowler, 2020). Além disso, mais de uma técnica pode ser usada para uma *Bad Smells* diferente: Extrair Requisitos, para o problema de Requisitos Grandes e Estrutura Condicional Complexa; Renomear Requisitos, para o Problema de Nomenclatura; Mover Atividades, para o problema de Requisitos Grandes, Estrutura Condicional Complexa e *Lazy Requirements*.

No trabalho de (Minuzzi, 2007) foi desenvolvido uma ferramenta para refatoração de *Story Cards*. Os artefatos são mapeados e transformados em diagrama de classes, depois analisados para validar se necessitam de refatoração ou não. Os mecanismos de refatoração são compostos de um conjunto de ontologias, propriedades OWL, conjunto de regras, pré e pós-condições.

No trabalho de (Rago et al., 2014) foi desenvolvido uma ferramenta, baseada em Processamento de Linguagem Natural, para refatorar Descrições de Casos de Uso. Foram desenvolvidas 8 técnicas de refatoração que são categorizadas em níveis baixo, médio e alto em relação a priorização para correção dos problemas de violação estrutural das Descrições de Casos de Uso e problemas de funcionalidades duplicadas.

No trabalho de (Sarmiento-Calisaya et al., 2020) foi desenvolvido uma ferramenta, baseado em Processamento de Linguagem Natural e Redes de Petri, para analisar Descrições de Casos de Uso baseados no *template* Cockburn (Cockburn, 2001). A ferramenta foi utilizada para detectar problemas de ambiguidade, completude e inconsistência de *template*. Logo, a ferramenta sugere refatorações de acordo com os problemas de escrita relacionados com as violações de *template*.

No trabalho de (Irshad et al., 2022) foi desenvolvido uma abordagem para refatorar especificações de requisitos com *template Behavior-driven Development* (BDD). Neste trabalho, são desenvolvidas 4 técnicas de refatoração usadas para corrigir problemas relacionados com similaridades de aspectos em especificações BDD: etapas, parâmetros, funcionalidades. Concluem que uma maior diversidade de especificações em BDD resultam em uma maior cobertura de funcionalidades de produtos.

No trabalho de (Farrell et al., 2022) foi desenvolvido uma ferramenta para refatoração de requisitos baseados no *template* FRETISH. O *template* FRETISH é usado para especificação de Requisitos em Linguagem Natural usando Lógica Temporal. Os adaptaram 4 técnicas de refatoração de código para refatorar requisitos. Estas técnicas de refatoração foram usadas para corrigir as violações no mau uso do *template* e que comprometem a qualidade da especificação.

Conforme pode ser observado, pesquisas vem sendo desenvolvidas com o objetivo de propos soluções, baseadas em refatoração, para correção de problemas de escrito em Requisitos em Linguagem Natural. Estas pesquisas propõem soluções para requisitos em Descrições de Casos de Uso, BDD, Cockburn e FRETISH. Nosso trabalho tem como foco a refatoração de *Requirements Smells* presentes em Requisitos em Linguagem Natural sem uso de *templates*.

4. Metodologia de Pesquisa

Nesta pesquisa, toma-se como pressuposto que: *Requirements Smells* são problemas de escrita que afetam a qualidade dos requisitos e de atividades subsequentes; Técnicas de Refatoração têm sua eficácia comprovada pela literatura (Fowler, 2020) para o tratamento de anomalias de código; e a Refatoração de Requisitos, traz benefícios para a qualidade da escrita de requisitos e para a sua evolução. Assim, busca-se neste estudo explorar o desenvolvimento de Técnicas de Refatoração para resolver *Requirements Smells*. Assim, foi realizado um Estudo Exploratório envolvendo a construção das Técnicas de Refatoração e a sua aplicação em conjunto de Histórias de Usuário. Analisando quais são os Critérios de Qualidade para Requisitos em Linguagem Natural, que são afetados por cada *Requirements Smells*.

Portanto, foi analisado o trabalho de (Nascimento et al., 2021), que fez um mapeamento entre cada *Requirements Smells* e os Critérios de Qualidade afetados. Neste estudo, foi realizada uma ampliação da Taxonomia de Critérios de Qualidade, desenvolvida por (Femmer et al., 2017), para cobrir Critérios de Qualidade para Histórias de Usuários:

- **Mínima:** é caracterizada pela presença dos componentes básicos da estrutura de um requisito;
- **Navegável:** caracterizada por links e/ou referências para acessar informações complementares aos requisitos;
- **Orientada ao Problema:** se refere ao requisito possuir informações sobre o problema ou necessidade do cliente;
- **Não Ambígua:** quando o requisito possui somente uma interpretação;
- **Livre de Conflito:** quando não existe duplicidade ou conflito de funcionalidades entre os requisitos;
- **Estimável:** está relacionada com o tamanho do requisito e o seu nível de complexidade;
- **Única:** quando um requisito não está duplicado na documentação;
- **Uniforme:** quando os requisitos seguem uma mesma estrutura de escrita;
- **Completo:** os requisitos possuem todas as informações necessárias dentro de si ou outras referências;
- **Negociável:** os requisitos descrevem total ou parcialmente o valor de negócio;
- **Valioso:** os requisitos entregam valor ao cliente;
- **Testável:** os requisitos possuem informações necessárias para estabelecer parâmetros e métricas para testes;

Após analisar o trabalho de (Nascimento et al., 2021) sobre como as *Requirements Smells* comprometem a qualidade, foi analisado o modelo proposto por (Fowler, 2020) para construção de técnicas de refatoração. Este modelo segue a seguinte estrutura:

1. **Nome:** nome da técnica de refatoração;
2. **Contexto:** descrição da Smell que deve ser refatorada;
3. **Motivação:** descrição dos impactos negativos da Smell para as questões de qualidade;

4. **Procedimento:** descrição sistemática das etapas de como deve ser feita a refatoração da Smell;

Assim, foi desenvolvido uma Técnica de Refatoração para cada *Requirements Smells*, onde cada técnica possui o nome de cada *Requirements Smells* correspondente. É importante frisar que, o uso das Técnicas de Refatoração dependem das decisões tomadas pelos praticantes de requisitos junto com os *stakeholders* em reuniões de negociação. Porque, são os usuários que devem disponibilizar as informações necessárias para que os praticantes, com o uso das Técnicas de Refatoração, façam a correção adequada dos requisitos (Pohl, 2016; Sommerville et al., 2011; Soto, 2023; Vera et al., 2024).

Os exemplos mencionados aqui são de artefatos de Histórias de Usuário do projeto *SmartHome*; um projeto de Casa Inteligente onde os requisitos foram capturados de um conjunto de *stakeholders*, para serem utilizados em pesquisas para automatização de requisitos (Murukannaiah et al., 2017). Estes artefatos foram traduzidos para o Português, idioma nativo dos autores, e afetados por *Requirements Smells* (infelizmente, a literatura carece de artefatos afetados por *Requirements Smells* disponíveis publicamente para pesquisas). A seguir serão descritas as Técnicas de Refatoração para *Requirements Smells*.

Substituição de Adjetivos e Advérbios Ambíguos

Contexto: a Smell Adjetivos e Advérbios Ambíguos é caracterizada pelo uso inadequado do conjunto de adjetivos e/ou advérbios, gerando ambiguidade na interpretação das informações.

Solução: substituir os adjetivos e advérbios ambíguos por numerais ou detalhamento das informações.

Motivação: a Smell compromete as Estimativas e Testes, pois não é possível mensurar as palavras ambíguas. Compromete a Completude, devido as várias interpretações de informações. Compromete a Negociação e o Valor, devido a ausência de compreensão da necessidade do cliente.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar o adjetivo e/ou advérbio ambíguo.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Se o adjetivo e/ou advérbio ambíguo está relacionado com quantidade ou qualidade, então substituir por quantificadores numéricos. Se geram ambiguidade pela ausência de informações, então acrescentar as informações necessárias.
4. Verificar se a ambiguidade foi removida.

Requisito Infectado: Como morador, quero que a minha *SmartHome* possa apagar as luzes quando estiver **muito tarde**.

Requisito Refatorado: Como morador, quero que a minha *SmartHome* possa apagar as luzes quando for **10:00PM**.

Substituição de Pronomes Vagos

Contexto: os Pronomes Vagos são pronomes com relações pouco claras, ou seja, pronomes indefinidos independente de sua classificação.

Solução: neste caso a melhor alternativa é alterar a posição das orações na frase.

Motivação: a *Smell* gera Ambiguidade pela várias interpretações sobre qual sujeito está tendo relação com o pronome. Compromete a Negociação e o Valor, devido a ausência de compreensão da necessidade do cliente.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar o Pronome Indefinido.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Se a sentença não possui sujeito, então colocar o sujeito na oração e substituir o pronome indefinido por um pronome definido correspondente ao sujeito da sentença ou reestruturar a frase. Se a sentença possui um sujeito, então substituir o pronome indefinido por um pronome definido correspondente ou reestruturar a frase. Se a sentença possui mais de uma palavra candidata a sujeito do pronome indefinido, então selecionar a palavra que será o sujeito, reestruturar a sentença ou substituir o pronome indefinido por um pronome definido e reestruturar a sentença.
4. Verificar se a ambiguidade foi removida.

Requisito Infectado: Como morador, quero que a minha SmartHome sincronize com o meu aplicativo de batimentos cardíacos, **onde** possa ativar uma música para se adequar ao meu humor quando eu chego do trabalho.

Requisito Refatorado: Como morador, quero que a minha SmartHome **possa ativar uma música** sincronizando com o meu aplicativo de batimentos cardíacos, para se adequar ao meu humor quando eu chego do trabalho.

Substituição de Linguagem Subjetiva

Contexto: a Linguagem Subjetiva ocorre quando são usados termos que passam a ideia de subjetividade das informações, ou seja, palavras que descrevem emoções, sentimentos ou expressam opinião pessoal.

Solução: trocar as palavras que geram subjetividade por palavras que passam a ideia de objetividade.

Motivação: a *Smell* compromete as Estimativas e Testes, pois não é possível mensurar as palavras ambíguas, a Completude e a Ambiguidade, pois leva a várias interpretações ou ausência de informações, a Negociação e o Valor, devido a ausência de compreensão da necessidade do cliente.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar o termo que gera Linguagem Subjetiva.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.

3. Substituir este(s) termo(s) por termo(s) com Linguagem Objetiva (uso de quantificadores numéricos, por exemplo). Ou adicionar informações objetivas relacionadas com a Linguagem Subjetiva.
4. Verificar se a Linguagem Subjetiva foi removida ou corrigida.

Requisito Infectado: Como morador, eu quero que a minha *SmartHome* avise quando o chuveiro aquecer na temperatura **adequada**.

Requisitos Refatorado: Como morador, eu quero que a minha *SmartHome* avise quando o chuveiro aquecer na temperatura de **30°C**.

Substituição de Adjetivos e Advérbios Comparativos

Contexto: os Adjetivos e Advérbios Comparativos ocorrem quando são utilizados adjetivos e advérbios para expressar relação de comparação do sistema com outro sistema.

Solução: substituir os adjetivos/advérbios de comparação, remover a sentença de comparação com outro sistema e reestruturar a sentença utilizando linguagem objetiva.

Motivação: a *Smell* compromete as Estimativas e Testes, pois não é possível mensurar as palavras ambíguas, a Completude e Ambiguidade, pois leva a várias interpretações ou ausência de informações, a Negociação, o Valor e Orientação ao Problema, devido a ausência de compreensão da necessidade do cliente.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar o adjetivo/advérbio Comparativo.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Remover o adjetivo/advérbio e a sentença de Comparativo.
4. Reestruturar a sentença, usando uma Linguagem Objetiva, adicionando informações sobre o problema.
5. Verificar se o Comparativo foi removido.

Requisito Infectado: Como morador, eu quero que a minha *SmartHome* tenha aspersores automáticos **melhores que** os aspersores automáticos da *SmartHome X*.

Requisito Refatorado: Como morador, eu quero que a minha *SmartHome* tenha aspersores automáticos de **tecnologia Y e versão 1.Y**.

Substituição de Sentenças Negativas

Contexto: as Sentenças Negativas são palavras usadas em funcionalidades que o sistema não deve fornecer, e onde falta de explicação sobre o comportamento do sistema em tais casos.

Solução: reestruturar a frase com as palavras sobre a funcionalidade que o sistema deve fornecer, ou acrescentar informações justificando o comportamento que o sistema não deve fornecer.

Motivação: a *Smell* compromete a Negociação e o Valor, devido a ausência de compreensão da necessidade do cliente. Pode gerar Conflito com outros requisitos, que

possuam a mesma sentença de forma positiva. Compromete a completude, devido a falta de informações.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar a Sentença Negativa.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Se a sentença não possui informações justificando a negação do comportamento do sistema, então acrescentar informações justificando o comportamento que o sistema não deve fornecer. Caso contrário, remover o requisito.
4. Verificar se a Sentença Negativa foi removida ou corrigida.

Requisito Infectado: Como morador, eu quero que a *SmartHome* **não** desative o controle de temperatura.

Requisito Refatorado: Como morador, eu quero que a *SmartHome* não desative o controle de temperatura, **porque** o sistema precisa monitorar de forma contínua a presença de pessoas ou animais na casa.

Substituição de Termos Não Verificáveis

Contexto: os Termos Não Verificáveis ocorre quando são utilizadas palavras que são difíceis de verificar, pois oferecem várias possibilidades de execução do sistema, ou que possuem várias interpretações.

Solução: substituir os Termos Não Verificáveis por termos objetivos ou colocar a descrição sobre o termo utilizado.

Motivação: a *Smell* compromete as Estimativas e Testes pois não é possível mensurar as palavras ambíguas. Compromete a Completude e Ambiguidade, devido as várias interpretações ou ausência de informações. Compromete a Negociação e o Valor, devido a ausência de compreensão da necessidade do cliente. Compromete a navegabilidade caso o Termo Não Verificável seja um termo técnico, que precisa de uma referência complementar para melhorar o entendimento.

Mecanismos: as seguintes atividades devem ser desempenhadas:

1. Localizar o Termo Não Verificável.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Se o Termo Não Verificável é um termo técnico, então acrescentar informações complementares ou adicionar citação ou *link* para referência complementar. Se o Termo Não Verificável não é um termo técnico, então substituir este termo por um termo verificável (por exemplo, quantificadores numéricos) ou acrescentar informações complementares.
4. Verificar se o Termo Não Verificável foi removido ou corrigido.

Requisito Infectado: Como morador, eu quero que a *SmartHome* realize o controle da temperatura **a qualquer momento**.

Requisito Refatorado: Como morador, eu quero que a *SmartHome* realize o controle da temperatura a **cada 1 minuto**.

Substituição de Voz Passiva

Contexto: a Voz Passiva corresponde a ausência do ator que está desempenhando uma determinada ação no sistema.

Solução: substituir Voz Passiva por Voz Ativa.

Motivação: a *Smell* gera Ambiguidade e Incompletude devido o sujeito da ação está oculto e gerar várias interpretações. Compromete a necessidade e o valor devido as várias interpretações sobre a necessidade do usuário.

Mecanismos: As seguintes atividades devem ser desempenhadas:

1. Localizar o sujeito paciente e a agente da passiva. Caso o agente da passiva não exista, neste caso ele sempre será o sistema.
2. Consultar o *Stakeholder* ou fornecedor do requisito para obtenção da informação correta.
3. Transformar o sujeito paciente em objeto direto e a agente da passiva em sujeito agente.
4. Substituir a locução verbal (caracterizada por verbos no particípio) por verbo no pretérito perfeito do indicativo.
5. Verificar se a Voz Passiva foi removida.

Requisito Infectado: Como morador, eu quero **ser notificado** quando alguém que não seja um membro da família ou outra pessoa de confiança estiver perto de minha casa.

Requisito Refatorado: Como morador, eu quero que a *SmartHome* me notifique por mensagem no celular quando alguém que não seja um membro da família ou outra pessoa de confiança estiver perto de minha casa.

5. Discussão sobre a Pesquisa

O que podemos analisar em relação ao desenvolvimento das Técnicas de Refatoração, é de que exige um conhecimento prévio sobre a Gramática da Língua Portuguesa. Estes conhecimentos envolvem: definições sobre as classes gramaticais, interpretação de texto e vícios de linguagem. Além disso, foi necessária a compreensão de estruturas gramaticais cuja semântica afetam Critérios de Qualidade; como por exemplo, uso de comparativos e superlativos.

Também, para o desenvolvimento dos mecanismos de correção, de forma sistematizada, foi necessária a compreensão de como corrigir os problemas gramaticais nos seus níveis léxico, sintático e semântico, por meio da compreensão das regras gramaticais. Para que a correção das sentenças possam garantir os Critérios de Qualidade para Requisitos em Linguagem Natural e possam manter a coesão gramatical.

Assim, observamos que o nível de correção realizado pelos praticantes de requisitos dependerá do nível de compreensão da gramática e de informações relacionadas

com os requisitos disponibilizados pelos stakeholders para realização das atividades de especificação. Portanto, os mecanismos de correção oferecem orientações de quais elementos gramaticais devem ser reestruturados ou substituídos. Ficando na responsabilidade do praticante de requisitos quais novas informações deverão ser acrescentadas.

6. Limitações do Trabalho

A principal limitação deste trabalho diz respeito à ausência, na literatura, de documentações de requisitos afetadas por *Requirements Smells*, principalmente de sistemas reais da indústria. Por isso, foi necessário selecionar alguns requisitos do sistema *SmartHome* e gerar novas versões contendo *Requirements Smells* para utilizarmos como exemplo nesta pesquisa. A outra limitação se deve a ausência de documentação de requisitos, em Português, disponíveis na literatura. Assim sendo necessária a tradução dos requisitos do Inglês para o Português, gerando mudança na estrutura gramatical, de modo que a criação dos mecanismos de refatoração vão seguir a estrutura gramatical do idioma de tradução. A outra limitação está relacionada com a validação das técnicas que refatoração, mas que será solucionado em trabalhos futuros.

7. Considerações Finais

Neste contexto, a ausência de mecanismos para correção de *Requirements Smells* disponíveis na literatura motivou o desenvolvimento desta pesquisa. Além disso, estudos correlatos justificando os benefícios do uso da refatoração de requisitos serviram de norte como métodos de correção, mostrando que técnicas de refatorações são soluções eficazes para melhoria da qualidade dos requisitos. Este trabalho propõe mecanismos de técnicas de refatoração para *Requirements Smells*, com o objetivo de resolver os problemas e garantir a qualidade dos requisitos. As técnicas seguem o modelo proposto por Martin Fowler, de modo que o usuário das técnicas possa compreender qual o tipo de *Smell* cada técnica atacar e quais os impactos na qualidade da especificação cada *Smell* compromete. Assim, o usuário pode verificar os exemplos para compreender melhor o uso das técnicas e verificar como fazer uso dos mecanismos. A ideia é que estas técnicas possam beneficiar praticantes de requisitos no seu cotidiano na análise, correção e negociação dos requisitos. Como trabalhos futuros, está a aplicação de estudos de casos envolvendo praticantes de requisitos, para avaliarmos a viabilidade das técnicas de refatoração em projetos reais.

Referências

- Cockburn, A. (2001). Writing effective use cases. Pearson Education India. <https://dl.acm.org/doi/abs/10.5555/517669>
- Farrell, M., Luckcuck, M., Sheridan, O., & Monahan, R. (2022). Towards Refactoring FRETish Requirements. In Proceedings of NASA Formal Methods: 14th International Symposium, NFM 2022, Pasadena, CA, USA, May 24–27, 2022, (pp. 272–279). https://doi.org/10.1007/978-3-031-06773-0_14

- Femmer, H., Fernández, D. M., Juergens, E., Klose, M., Zimmer, I., & Zimmer, J. (2014). Rapid requirements checks with requirements smells: two case studies. In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, (pp. 10–19). <http://dx.doi.org/10.1145/2593812.2593817>
- Femmer, H., Fernández, D. M., Wagner, S., & Eder, S. (2017). Rapid quality assurance with requirements smells. *Journal of Systems and Software*, 123, 190–213. <https://doi.org/10.1016/j.jss.2016.02.047>
- Fowler, M. (2020). *Refatoração: Aperfeiçoando o design de códigos existentes*. Novatec Editora.
- Irshad, M., Börstler, J., & Petersen, K. (2022). Supporting refactoring of BDD specifications—An empirical study. *Information and Software Technology*, 141, 106717. <https://doi.org/10.1016/j.infsof.2021.106717>
- ISO/IEC/IEEE 29148. (2018). ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering.. <https://doi.org/10.1109/IEEESTD.2018.8559686>
- Mens, T., & Tourwé, T. (2004). A survey of software refactoring. *IEEE Transactions on software engineering*, 30(2), 126–139. <https://doi.org/10.1109/TSE.2004.1265817>
- Minuzzi, T. da S. (2007). *Ustory-Refactory: ferramenta de refatoração de requisitos aplicada em cartões user stories (CRC Cards)*. [Tese de Mestrado, Escola Politécnica, Universidade Vale do Rio dos Sinos]. Repositório UNISINOS. <https://repositorio.jesuita.org.br/handle/UNISINOS/2242>
- Murukannaiah, P. K., Ajmeri, N., & Singh, M. P. (2017). Toward automating crowd RE. 2017 IEEE 25th International Requirements Engineering Conference (RE), 512–515. <http://dx.doi.org/10.1109/re.2017.74>
- Nascimento, R., Aranha, E., Kulesza, U., & Lucena, M. (2018). Requirements Smells como indicadores de má qualidade na especificação de requisitos: Um Mapeamento Sistemático da Literatura. *WER*. <http://dx.doi.org/10.17771/pucRio.wer.inf2018-40>
- Nascimento, R., Guimarães, E., & Lucena, M. (2021). Requirements Smells como Indicador de Qualidade para Histórias de Usuários: Estudo Exploratório. Em M. L. P. de Menezes Cruz, G. D. S. Hadad, & J. C. Marques (Orgs.), *Anais do WER21 - Workshop em Engenharia de Requisitos*, Brasília, BSB, Brasil, August 23-27, 2021. Editora PUC-Rio. <http://dx.doi.org/10.29327/1298728.24-19>
- Opdyke, W. F. (1992). *Refactoring Object-Oriented Frameworks*. University of Illinois. <https://dl.acm.org/doi/abs/10.5555/871113>
- Pohl, K. (2016). *Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant*. Rocky Nook, Inc. <https://dl.acm.org/doi/10.5555/1869735>
- Rago, A., Frade, P., Ruiva, M., & Marcos, C. (2014). An Approach for Automating Use Case Refactoring. *Electronic Journal of SADIO*, 13(1), 54–68. <https://ojs.sadio.org.ar/index.php/EJS/article/view/41>

- Ramos, R., Piveta, E. K., Castro, J., Araújo, J., Moreira, A., Guerreiro, P., Pimenta, M. S., & Price, R. T. (2007). Improving the quality of requirements with refactoring. In *Anais do VI Simpósio Brasileiro de Qualidade de Software*, (pp. 141–155). <http://dx.doi.org/10.5753/sbqs.2007.15573>
- Russo, A., Nuseibeh, B., & Kramer, J. (1998). Restructuring requirements specifications for managing inconsistency and change: A case study. In *Proceedings of IEEE International Symposium on Requirements Engineering: RE'98*, (pp. 51–60). <http://dx.doi.org/10.1109/icre.1998.667808>
- Sarmiento-Calisaya, E., Cárdenas, E. H., Cornejo-Aparicio, V., & Alzamora, G. S. (2020). Towards the Improvement of Natural Language Requirements Descriptions: The C&L Tool. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, (pp. 1405–1413). <https://doi.org/10.1145/3341105.3374028>
- Sommerville, I., Melnikoff, S. S. S., Arakaki, R., & de Andrade Barbosa, E. (2011). *Engenharia de software* (Vol. 9). Addison Wesley.
- Soto, W. (2023). Contrato Inteligente para la Gestión de Requerimientos en la Construcción de Software. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (49), 147–160. <https://doi.org/10.17013/risti.49.147-160>
- Vera, R. A. A., Charuf, R. A. S., Mendoza, J. C. D., & Pech, J. P. U. (2024). Factores Sociales y Culturales que Influyen en las Técnicas de Priorización de Requisitos de Software: Un Estudio Secundario. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (53), 53–68. <https://doi.org/10.17013/risti.53.53-68>