

UNIVERSIDADE FEDERAL DO PAMPA

Guilherme Legramante Martins

**Estudos Empíricos sobre Abordagens para
Identificação de Interesses Transversais na
Engenharia de Requisitos**

Alegrete
2018

Guilherme Legramante Martins

**Estudos Empíricos sobre Abordagens para
Identificação de Interesses Transversais na
Engenharia de Requisitos**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Software da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Engenharia de Software.

Orientador: Prof. Me. João Pablo Silva da
Silva

Alegrete
2018


Guilherme Legramante Martins

**Estudos Empíricos sobre Abordagens para
Identificação de Interesses Transversais na
Engenharia de Requisitos**

Trabalho de Conclusão de Curso apresentado
ao Curso de Graduação em Engenharia de
Software da Universidade Federal do Pampa
como requisito parcial para a obtenção do tí-
tulo de Bacharel em Engenharia de Software.

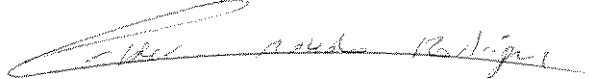
Trabalho de Conclusão de Curso defendido e aprovado em 03. de dezembro de 2013

Banca examinadora:



Prof. Me. João Pablo Silva da Silva

Orientador
Unipampa



Prof. Dr. Elder de Macedo Rodrigues

Unipampa



Prof. Dr. Maicon Bernardino da Silveira

Unipampa

Este trabalho é dedicado aos meus filhos, Vallentina e Mateus.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me permitir chegar até aqui, vencendo mais essa etapa da vida. Agradeço à minha família, aos meus pais Bia e César, por ensinarem desde cedo o valor do estudo, à minha esposa Aline, a principal incentivadora nesta jornada e aos meus filhos Vallentina e Mateus por entenderem os momentos de ausência.

Também agradeço aos professores que tiveram um papel importante nesta caminhada, em especial ao meu orientador, prof^o João Pablo, por dar o suporte necessário, mostrando o rumo a ser seguido e ao prof^o Paulo Júnior (UFSCar), pela presteza e disponibilidade.

Agradeço aos colegas, que se tornaram amigos, durante o percurso desses 4 anos, particularmente ao Adriel Herter pela hospitalidade e aulas de FIFA, ao Adriel Rodrigues pelas parcerias musicais e Dionatã Mafaldo pelas sempre produtivas discussões, muito obrigado pela amizade de vocês!

“Tudo o que fizerem, façam de todo o coração,
como para o Senhor, e não para os homens,
sabendo que receberão do Senhor a recompensa da herança.
É a Cristo, o Senhor, que vocês estão servindo.
(Bíblia Sagrada, Colossenses 3:23-24)

RESUMO

Requisitos descrevem as funções a ser oferecidas pelo software e as possíveis restrições a seu funcionamento. Um grupo de requisitos com uma mesma finalidade é chamado de interesse. Esses interesses podem estar espalhados e entrelaçados no sistema, nesse caso são chamados Interesses Transversais (ITs). Identificar e separar esses interesses é fundamental para garantir uma melhor manutenibilidade e evolução do software. A Engenharia de Requisitos Orientada a Aspectos (EROA) estuda a identificação, representação e composição desses interesses, conhecidos também como aspectos, durante a engenharia de requisitos. Existem diversos trabalhos que propõem abordagens para facilitar a identificação e separação de interesses transversais durante a engenharia de requisitos. Todavia, estudos empíricos que comparem e analisem em detalhes essas abordagens ainda são escassos. Este trabalho tem como objetivo principal realizar estudos sobre abordagens para identificação de interesses de software em documentos de requisitos. Para isso, foram realizados dois estudos empíricos, uma Revisão Sistemática da Literatura (RSL), para investigação do estado da arte sobre o tema, e um Experimento Controlado com Sujeitos para comparar a efetividade de duas abordagens para identificação de Interesses Transversais em documentos de requisitos. Por intermédio de uma RSL foram selecionadas as abordagens para identificação de Interesses Transversais mais relevantes e por meio de um experimento controlado duas dentre as abordagens selecionadas foram comparadas para análise da efetividade das mesmas. Com a realização dos estudos, verificou-se em detalhes as abordagens mais relevantes para EROA, bem como duas delas foram analisadas através de um estudo experimental que comparou a efetividade das mesmas. A RSL realizada permitiu selecionar os principais trabalhos da área de estudo e os resultados obtidos por meio do experimento realizado indicam que, para o contexto e cenário previstos, existem diferenças significativas entre as abordagens para identificação de interesses sob estudo. Com a realização deste trabalho foi possível obter um catálogo das principais e mais relevantes abordagens pra identificação de Interesses Transversais na ER, compilando o conhecimento sobre essas abordagens. Evidências empíricas foram obtidas que possibilitam diferenciar as abordagens *ObasCId* e *Theme/Doc* com relação à efetividade das mesmas.

Palavras-chave: Interesses de Software. Interesses Transversais. Engenharia de Requisitos Orientada a Aspectos. Estudos Empíricos.

ABSTRACT

Requirements describe the functions to be offered by the software and possible restrictions on its operation. A group of requirements with the same purpose is called interest. These interests may be scattered and tangled in the system, in which case they are called Crosscutting Concerns (CCs). Identifying and separating these interests is critical to ensuring better software maintainability and evolution. Aspect-Oriented Requirements Engineering (AORE) studies the identification, representation and composition of these interests, also known as aspects, during the requirements engineering. There are several papers that propose approaches to facilitate the identification and separation of Crosscutting Concerns during requirements engineering. However, empirical studies comparing and analyzing these approaches in detail are still scarce. This paper has as main objective to carry out studies on approaches to identify interests of software in documents of requirements. Thereunto, two empirical studies, a Systematic Review of Literature (SLR), were conducted to investigate the state of the art on the subject, and a Experiment with Subjects to compare the effectiveness of two approaches to identify Crosscutting Concerns in requirements documents. The most relevant approaches were identified through a systematic literature review, and by a experiment two of the selected approaches were compared for their effectiveness analysis. With the accomplishment of the studies, the most relevant approaches to AORE were verified in detail, as well as two of them were analyzed by an experimental study that compared their effectiveness. The review made it possible to select the main works of the study area and the results obtained through the experiment indicate that, for the context and scenario predicted, there are significant differences between the approaches to identify concerns under study. With the accomplishment of this paper it was possible to obtain a catalog of the main and most relevant approaches for the identification of Crosscutting Concerns in Requirements Engineering, compiling the knowledge about these approaches. Empirical evidence has been obtained that makes it possible to differentiate the *ObasCIId* and *Theme/Doc* approaches with respect to their effectiveness.

Key-words: Concerns. Crosscutting Concerns. Aspect-Oriented Requirements Engineering. Empirical Studies.

LISTA DE FIGURAS

Figura 1 – Estrutura Analítica do Projeto - EAP.	25
Figura 2 – Exemplo de interesses transversais.	28
Figura 3 – Etapas da abordagem AOFA.	41
Figura 4 – Exemplo de Goal Model.	43
Figura 5 – Visão Geral da abordagem DISCERN.	44
Figura 6 – Parte A da ontologia O4C.	45
Figura 7 – Parte B da ontologia O4C.	46
Figura 8 – Fases da ObasCId.	47
Figura 9 – Exemplo de um Catálogo de Interesses de Software.	47
Figura 10 – Atividades da fase de Identificação e Classificação de Interesses.	48
Figura 11 – Exemplo de Action View.	51
Figura 12 – Exemplo de Composição de Interesses.	52
Figura 13 – Resultados do experimento.	65
Figura 14 – Resultados do questionário.	67
Figura 15 – Treinamento Interesses de Software.	81
Figura 16 – Treinamento abordagem ObasCId.	81
Figura 17 – Treinamento abordagem Theme/Doc.	82
Figura 18 – Termo de Consentimento.	83
Figura 19 – Catálogo de Interesses de Software.	87
Figura 20 – Requisitos ObasCId Tool.	89
Figura 21 – Requisitos Sistema de Gerenciamento de Cursos.	90
Figura 22 – Gabarito dos requisitos utilizados no experimento.	91

LISTA DE TABELAS

Tabela 1 – Exemplo de aplicação de abordagem para EROA.	30
Tabela 2 – Resultado da busca nas bases	37
Tabela 3 – Aplicação dos critérios de qualidade	38
Tabela 4 – Motivações dos trabalhos selecionados	38
Tabela 5 – Síntese do resultado	39
Tabela 6 – Soluções propostas	40
Tabela 7 – Aplicação dos critérios de qualidade nos trabalhos selecionados	40
Tabela 8 – Modelo de documento de requisitos adotado pela abordagem ObasCId.	48
Tabela 9 – Exemplo de documento de requisitos com as palavras-chave identificadas.	49
Tabela 10 – Exemplo de uma matriz de entrelaçamentos entre interesses.	50
Tabela 11 – Seleção das abordagens mais relevantes	52
Tabela 12 – Sessões do experimento	62
Tabela 13 – Resultados da identificação de interesses	64
Tabela 14 – Resultados teste <i>Shapiro-Wilk</i>	65

LISTA DE SIGLAS

ACM Association for Computing Machinery

AOFA Aspect-Oriented Feature Analysis

EAP Estrutura Analítica do Projeto

ER Engenharia de Requisitos

EROA Engenharia de Requisitos Orientada a Aspectos

IEEE Institute of Electrical and Electronics Engineers

OA Orientação a Aspectos

PLN Processamento de Linguagem Natural

RSL Revisão Sistemática da Literatura

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	23
1.2	Objetivos	24
1.3	Metodologia	25
1.4	Contribuições do Trabalho	26
1.5	Organização	26
2	FUNDAMENTAÇÃO	27
2.1	Interesses de Software	27
2.2	Engenharia de Requisitos Orientada a Aspectos	28
2.3	Engenharia de Software Experimental	29
2.3.1	Revisão Sistemática da Literatura	31
2.3.2	Experimento Controlado	31
2.4	Lições do Capítulo	32
3	REVISÃO SISTEMÁTICA DA LITERATURA	33
3.1	Trabalhos Relacionados	33
3.2	Protocolo da RSL	34
3.2.1	Questões de Pesquisa	34
3.2.2	Processo de Busca	34
3.2.2.1	Association for Computing Machinery	35
3.2.2.2	IEEE Explore Digital Library	35
3.2.2.3	Elsevier Science Direct	35
3.2.2.4	Springer International Publishing	35
3.2.3	Processo de Seleção	35
3.2.4	Coleta e Análise de Dados	36
3.3	Execução da RSL	37
3.4	Análise do Resultado	37
3.4.1	Q1. Quais as motivações dos trabalhos relacionados a Interesses Transversais na Engenharia de Requisitos?	38
3.4.2	Q2. Quais os problemas relacionados a Interesses Transversais na Engenharia de Requisitos?	39
3.4.3	Q3. Quais soluções são mais relevantes para a identificação de Interesses Transversais na Engenharia de Requisitos?	39
3.5	Lições do Capítulo	39
4	CATÁLOGO DE ABORDAGENS PARA IDENTIFICAÇÃO DE INTERESSES	41
4.1	Detalhamento das Abordagens	41

4.1.1	Aspect-Oriented Feature Analysis	41
4.1.2	AspectQuery	42
4.1.3	DISCERN	43
4.1.4	EA-Miner	44
4.1.5	ObasCIId	45
4.1.6	Theme/Doc	50
4.2	Análise das Abordagens	51
4.3	Lições do Capítulo	52
5	EXPERIMENTO CONTROLADO	55
5.1	Trabalhos Relacionados	55
5.2	Escopo	55
5.3	Planejamento	56
5.3.1	Seleção do Contexto	56
5.3.2	Formulação das Hipóteses	56
5.3.3	Seleção dos Sujeitos	57
5.3.4	Desenho do Experimento	58
5.3.5	Instrumentação	58
5.3.6	Ameaças à Validade	59
5.3.6.1	Validade de Conclusão	59
5.3.6.2	Validade Interna	60
5.3.6.3	Validade de Construção	61
5.3.6.4	Validade Externa	61
5.4	Execução do Experimento	62
5.5	Resultados do Experimento	63
5.5.1	Teste de Hipótese	64
5.6	Utilidade Percebida e Facilidade de Uso	66
5.7	Lições do Capítulo	66
6	CONSIDERAÇÕES FINAIS	69
6.1	Trabalhos Futuros	69
	REFERÊNCIAS	71
	APÊNDICES	77
	APÊNDICE A – QUESTIONÁRIO PARA DEFINIÇÃO DOS	
	GRUPOS	79

APÊNDICE B – DOCUMENTAÇÃO DOS TREINAMEN- TOS REALIZADOS	81
APÊNDICE C – TERMO DE CONSENTIMENTO	83
ANEXOS	85
ANEXO A – CATÁLOGO DE INTERESSES DE SOFTWARE UTILIZADO NA ABORDAGEM OBASCID .	87
ANEXO B – DOCUMENTAÇÃO DE REQUISITOS - EX- PERIMENTO	89
Índice	93

1 INTRODUÇÃO

Um requisito define uma característica, restrição ou funcionalidade que atenda às regras de negócio definidas para o software, sendo classificado como requisito funcional ou requisito não funcional (SOMMERVILLE, 2011). A primeira etapa do ciclo de vida de desenvolvimento de um software é a Engenharia de Requisitos (ER). Segundo Pressman (2010), essa etapa foi criada para definir todas as atividades envolvidas em descobrir, documentar e manter um conjunto de requisitos para um projeto de sistema de software. A ER envolve as atividades do ciclo de vida com fins de identificação de requisitos de usuário, análise e documentação dos requisitos como uma especificação e validação dos requisitos documentados de acordo com as necessidades do usuário, bem como os processos envolvidos nesse contexto.

Um conjunto de requisitos de software relacionados com um mesmo propósito é chamado de interesse ou *concern*. Parnas (1972) e Dijkstra (1976) em meados da década de 70 contribuíram para as definições sobre os conceitos de separação de interesses, coesão e acoplamento no contexto da Engenharia de Software por meio de noções de modularização. De acordo com o *Standard Glossary of Software Engineering Terminology*, a modularização em sistemas de software se refere ao grau de modificações necessárias em componentes de um determinado sistema que são provocadas em decorrência de mudanças em um componente (IEEE, 1990). Deste modo, sistemas bem modularizados podem trazer alguns benefícios como reusabilidade, compreensão, adaptabilidade, manutenibilidade entre outros (MEYER, 1997). Por esse motivo, engenheiros de software consideram a modularização como um princípio chave (SANTANNA, 2008). Um interesse espalhado e entrelaçado, por entrecortar transversalmente as demais estruturas de um sistema, é também chamado de Interesse Transversal (IRWIN, 1997). **Desempenho** é um exemplo de Interesse Transversal, pois afeta diversos outros interesses de um sistema.

Por meio de características da Orientação a Aspectos (OA) é possível garantir a separação dos interesses espalhados no software. Por meio da OA é possível o encapsulamento de Interesses Transversais, adicionando um mecanismo extra de abstração, chamado Aspecto (SUTTON Jr., 2002). A Programação Orientada a Aspectos permite uma melhor modularização dos sistemas, aumentando assim sua manutenibilidade. Nesse contexto, encontra-se a Engenharia de Requisitos Orientada a Aspectos (EROA) que visa a identificação dos interesses de software na ER.

1.1 Motivação

Para diminuição dos efeitos causados pelo espalhamento e entrelaçamento dos Interesses Transversais existentes no sistema, o Engenheiro de Software deve identificá-los o mais cedo possível no ciclo de vida do software (RESENDE, 2007). Porém, essa não é uma atividade trivial, pois muitos interesses estão implícitos no documento de requisitos. Com base nisso, são necessárias tecnologias que ajudem o engenheiro a identificar os

Interesses Transversais. Várias pesquisas têm sido conduzidas no contexto da EROA com o propósito de obter técnicas e abordagens para essa identificação de Interesses de Software durante a ER.

Segundo Baniassad (2004), os engenheiros de software encontram dificuldade em identificar os Interesses Transversais de um software. Duas causas podem estar associadas a esses problemas. A primeira remete à falta de compreensão a respeito do domínio de interesses de software. De acordo com Razzaq (2012), há poucos estudos que expliquem detalhadamente o conceito de interesse de software. Já a segunda causa está relacionada à escassez de recursos apropriados para apoiar os engenheiros de software na atividade de identificação e classificação dos interesses de software.

Algumas abordagens foram propostas para a atividade de identificação de interesses de software durante a ER. A abordagem *AOFA*, proposta por Razzaq (2012), é baseada na OA. A abordagem *AspectQuery*, proposta por Chengwan (2013), usa um modelo de objetivos para a identificação dos interesses. A abordagem *Discern*, proposta por Rosenhainer (2005a), apoia-se em um modelo de casos de uso. A abordagem *EA-Miner*, proposta por Sampaio e Rayson (2005), é uma abordagem automatizada para a extração de interesses em documentos de requisitos. A abordagem *ObasCId*, proposta por Parreira Jr. (2018), é fundamentada em um modelo ontológico que auxilia o engenheiro de software na identificação dos interesses e a abordagem *Theme/Doc*, proposta por Baniassad (2004), usa um modelo chamado de *Action View* para facilitar a visualização dos Interesses Transversais identificados no software.

Apesar de já existirem técnicas e abordagens para EROA, ainda não há muitos trabalhos que realizem estudos empíricos em cima dessas abordagens. Essa necessidade fica evidenciada na Seção “*Trabalhos Futuros*” do trabalho de Parreira Jr. (2018) onde é sugerido que sejam feitos estudos empíricos da abordagem proposta com outras abordagens já existentes.

1.2 Objetivos

O objetivo geral deste trabalho é realizar estudos empíricos sobre abordagens para identificação de interesses de software. Esses estudos buscam detalhar as principais atividades e técnicas envolvidas na EROA, bem como selecionar as abordagens mais relevantes e obter indícios sobre possíveis diferenças entre a efetividade das mesmas.

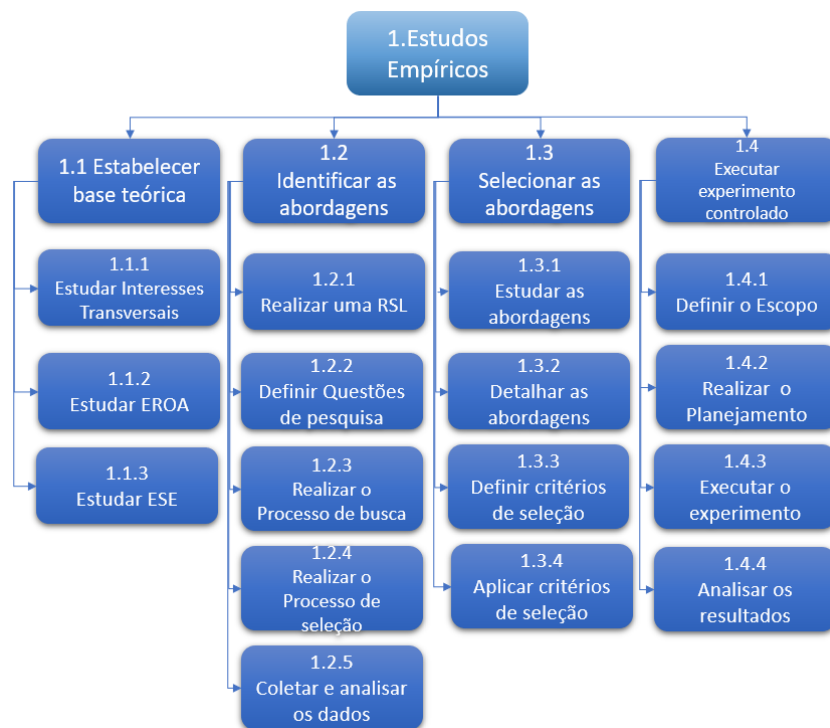
Os objetivos específicos deste trabalho são:

- Estabelecer a base teórica necessária para compreender os Interesses Transversais.
- Identificar as abordagens para identificação de Interesses Transversais existentes.
- Selecionar as abordagens mais relevantes e passíveis de comparação.
- Realizar um experimento controlado com as abordagens selecionadas.

1.3 Metodologia

A Figura 1 apresenta a Estrutura Analítica do Projeto (EAP). Segundo o PM-BOK (2013) uma EAP é composta por uma hierarquia de pacotes entregáveis para um determinado projeto. Para este trabalho, a EAP facilita a forma de visualização do objetivo principal, objetivos específicos e tarefas que compõem cada um desses objetivos. O primeiro objetivo específico é estabelecer uma base teórica para o trabalho. Para isso, foram estudadas as áreas de Interesses Transversais, da EROA e da Engenharia de Software Experimental.

Figura 1 – Estrutura Analítica do Projeto - EAP.



Fonte: Elaborado pelo autor.

O segundo objetivo específico busca a identificação das abordagens mais relevantes para identificação de Interesses Transversais. Para isso, foi realizada uma Revisão Sistemática da Literatura (RSL), em que por intermédio de uma busca por trabalhos disponíveis na literatura foram respondidas as questões de pesquisa propostas no estudo.

Selecionar as abordagens para identificação de Interesses Transversais também está elencado como objetivo específico deste trabalho. Para a realização dessa seleção, foi feito um estudo detalhado das abordagens selecionadas na RSL realizada, com a aplicação de critérios de seleção.

O quarto objetivo específico deste trabalho é a realização de um experimento controlado para comparação da efetividade de abordagens para identificação de Interesses

Transversais na ER. Para a realização desse experimento é definido um escopo, um planejamento e execução, finalizando com a análise dos resultados.

1.4 Contribuições do Trabalho

As principais contribuições deste trabalho estão relacionadas aos resultados obtidos com a condução dos estudos empíricos realizados. São contribuições deste trabalho:

- Criação de um catálogo com as principais abordagens para identificação de Interesses Transversais na ER.
- Evidências empíricas sobre diferenças entre a efetividade das abordagens *ObasCId* e *Theme/Doc*, para o contexto e cenário aplicados.
- Dados relativos à utilidade percebida e facilidade de uso das abordagens *ObasCId* e *Theme/Doc*

1.5 Organização

O restante deste trabalho está organizado com segue. No Capítulo 2 é apresentada a fundamentação teórica do mesmo, onde são explanados os conceitos principais sobre Interesses de Software, EROA e definições sobre Engenharia de Software Experimental, seguido das lições do capítulo. No Capítulo 3 é apresentada a RSL realizada para obter o estado da arte sobre interesses de software e principais abordagens para identificação destes, seguido das lições do capítulo. O Capítulo 4 detalha o catálogo de abordagens para identificação de interesses de software em documentos de requisitos que foi obtido por meio da RSL realizada. Em seguida, o Capítulo 5 apresenta o experimento controlado conduzido para comparação das abordagens selecionadas, contendo o escopo, planejamento, execução e análise dos resultados, concluindo com as lições aprendidas. No Capítulo 6 são apresentadas as considerações finais deste trabalho, onde são retomados os objetivos, motivações, principais contribuições, finalizando com os trabalhos futuros.

2 FUNDAMENTAÇÃO

Este capítulo tem por finalidade apresentar os principais conceitos em que este trabalho está inserido. Na Seção 2.1 são abordadas as definições de Interesses de Software. A Seção 2.2 apresenta as definições sobre EROA. A Seção 2.3 discorre sobre os principais conceitos da Engenharia de Software Experimental. As lições do capítulo são apresentadas na Seção 2.4.

2.1 Interesses de Software

Interesses de Software, do inglês *concerns*, apresentam diferentes definições segundo alguns autores. Para Sommerville (2011), esses interesses refletem os requisitos do sistema e das prioridades definidas pelos envolvidos com o sistema. Para Sutton Jr. (2002), um interesse pode ser definido como qualquer característica em um sistema. Essas características podem estar associadas com o sistema e com o contexto em que ele se insere. Por exemplo, políticas de uso e prioridades organizacionais. A separação desses interesses já é um conceito bastante antigo para a computação, remete à década de 70, no trabalho de Dijkstra (1976). Essa separação visa a decomposição e modularização, ou seja, sistemas de software devem ser decompostos em unidades modulares que lidem isoladamente com apenas um interesse.

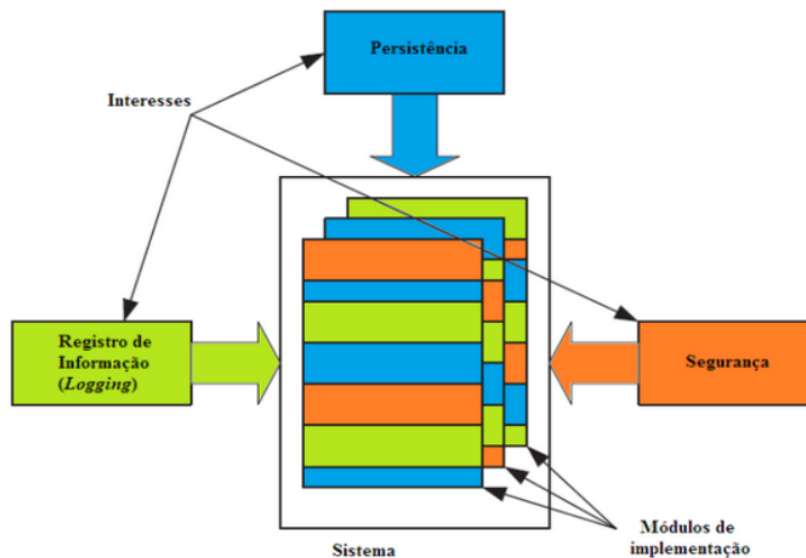
Esses interesses podem estar relacionados com requisitos funcionais e requisitos não funcionais do sistema (BANIASSAD, 2004). Quando um interesse está relacionado à uma funcionalidade do software, por exemplo, “*Cadastrar um produto*”, ele é considerado um Interesse Funcional. Já quando ele está relacionado à uma restrição (requisito não funcional), por exemplo, “*Segurança*”, é considerado um Interesse Não Funcional. Além disso, podem ser separados em dois grupos: base e transversal. No grupo base se encontram os interesses que não estão relacionados com os demais interesses presentes no software. No grupo transversal estão contidos os Interesses Transversais do software, ou seja, aqueles interesses que entrecortam demais interesses do sistema. Por exemplo, no requisito “*Para cadastrar um produto o usuário deverá estar logado no sistema*” há dois interesses presentes um funcional (“*Cadastrar um produto*”) e um não funcional (“*Segurança*”) pois é exigido que o usuário esteja logado e neste caso essa atividade de login está entrecortando o interesse funcional presente nesse requisito.

Embora já existam esses conceitos bem definidos, os problemas relacionados à separação inadequada de interesses de software ainda persistem. Diante disso, estudos avançados sobre a separação de interesses têm sido feitos, tanto no paradigma da Orientação a Objetos quanto na Orientação a Aspectos, onde um interesse é conhecido como aspecto (ROSENHAINER, 2005a). São estudados os interesses transversais ou *cross-cutting concerns*, pois esse tipo de interesse pode estar diretamente relacionado com os fenômenos de entrelaçamento e espalhamento dentro de um software. O entrelaçamento ou interdependência acontece quando há uma ligação entre determinadas característi-

cas do sistema com relação de dependência entre as mesmas. Já o espalhamento ocorre quando há uma característica espalhada por diversas partes do sistema, causando uma diminuição na modularidade do software (IRWIN, 1997).

A Figura 2 apresenta um exemplo da transversalidade entre interesses, onde há os Interesses Transversais persistência, *logging* e segurança com os fenômenos de entrelaçamento e espalhamento em diversos módulos de implementação.

Figura 2 – Exemplo de interesses transversais.



Fonte: Adaptado de Henry (1981).

2.2 Engenharia de Requisitos Orientada a Aspectos

Sommerville (2011) define a ER como sendo o conjunto de técnicas empregadas para levantar, detalhar, documentar e validar os requisitos de um produto, ou seja, está relacionada com a descoberta, análise, documentação e verificação das funcionalidades a serem suportadas pelo software juntamente das suas restrições. Para Chen, Wang e Dong (2003), a ER especifica um problema a ser solucionado, identifica os limites do sistema, analisa as propriedades do ambiente e caracteriza os comportamentos do software resultante. Quatro etapas fundamentais são encontradas na ER. Um estudo de viabilidade é a primeira etapa realizada na ER, onde se busca verificar a real utilidade do sistema e se há condições para sua construção. A partir disso, é realizada a elicitação e análise dos requisitos, onde os mesmos são extraídos e analisados, seguidos da especificação dos requisitos, que busca especificar de uma forma padrão os requisitos do sistema. Por fim, é feita a validação dos requisitos, para verificar se os requisitos realmente definem o esperado

para o software. Uma ER bem estruturada pode garantir qualidade, confiabilidade, e integridade ao produto desenvolvido (PRESSMAN, 2010).

Geralmente as atividades de elicitación e análise de requisitos, subatividades da ER, geram documentos textuais com a descrição dos requisitos do sistema. Para fins de promover uma identificação e modularização, esses documentos são analisados durante a análise de requisitos e os requisitos são estruturados em unidades individuais como casos de uso, cenários, *user stories*, dentre outras (CHITCHYAN; SAWYER, 2005).

A maioria das abordagens tradicionais para ER, como por exemplo, casos de uso e histórias de usuário, baseiam-se no princípio da separação de interesses. Porém, não conseguem tratar corretamente os Interesses Transversais, pois alguns deles não são simples de serem modularizados, uma vez que se encontram espalhados e entrelaçados com demais interesses. Diante disso, surge a necessidade da ER poder tratar adequadamente esse tipo de interesse.

A EROA é baseada na OA. A OA é um paradigma da computação que visa mitigar alguns problemas que outros paradigmas clássicos não tratam adequadamente. Na OA os Interesses Transversais são tratados como Aspectos. Um Aspecto equivale a uma classe da Orientação a Objetos, implementando um interesse em um módulo (PARREIRA Jr., 2018). Com os Interesses Transversais abstraídos em aspectos, é possível reduzir os fenômenos de entrelaçamento e espalhamento mantendo sua modularidade, uma vez que um aspecto é uma unidade modular do sistema. Os Interesses Transversais identificados durante a ER são chamado de *Early Aspects*.

A Tabela 1 exibe um exemplo de aplicação da abordagem proposta por Parreira Jr. (2018) para identificação de interesses de software em documento de requisitos. No primeiro requisito, há cinco interesses, persistência, segurança, reclamação, desempenho e usabilidade, já nos requisitos 2 e 3 há apenas um interesse em cada, usabilidade e desempenho, respectivamente. Esses interesses foram mapeados pelo engenheiro de software ao analisar o documento de requisitos. Na última coluna é possível identificar qual interesse o engenheiro de software definiu como interesse principal, reclamação para o requisito 1, usabilidade para o requisito 2 e desempenho para o requisito 3.

2.3 Engenharia de Software Experimental

Para avaliar as pesquisas na área de Engenharia de Software, é necessário entender os métodos que estão disponíveis para essa atividade, bem como suas limitações e possíveis aplicações, tendo quatro métodos que podem ser utilizados no campo da Engenharia de Software: científico, de engenharia, experimental e analítico (WOHLIN et al., 2012).

No método científico, o mundo é observado e então é criado um modelo, baseado nessa observação, como por exemplo, uma simulação. No método de engenharia, as soluções existentes são estudadas e então são propostas melhorias para em seguida serem avaliadas. No método experimental, um modelo é proposto e avaliado por meio de estu-

Tabela 1 – Exemplo de aplicação de abordagem para EROA.

Requisito RF-01	Interesses	Principal
Tem como propósito realizar a atualização do andamento de uma reclamação . A reclamação deve estar cadastrada, com a situação ABERTA e com o funcionário logado no sistema.	Persistência	
	Segurança	
	Reclamação	X
	Desempenho	
	Usabilidade	
Requisito RNF-02	Interesses	Principal
O sistema deve ter uma interface de fácil utilização , que contemple possíveis restrições de utilização dos usuários. O sistema deve ter um HELP on-line para ser consultado por qualquer pessoa que acesse o sistema.	Usabilidade	X
Requisito RNF-03	Interesses	Principal
O tempo de resposta do sistema não deve ultrapassar 5 segundos por acesso.	Desempenho	X

Fonte: Adaptado de Parreira Jr. (2018).

dos empíricos como por exemplo, estudos de caso e experimentos controlados. Por fim, no método analítico, uma teoria formal é proposta e então comparada com observações empíricas.

Segundo Glass (1994), a abordagem mais utilizada para a experimentação na área de engenharia de software é o método experimental que considera a proposição e avaliação do modelo por meio de estudos empíricos. Porém, os demais métodos também podem ser utilizados. Por exemplo, o método científico pode ser usado para compreender o processo de construção do software para verificação de melhorias nesse processo. O método de engenharia pode, por exemplo, ajudar a comparar o desempenho de ferramentas. O método analítico pode obter provas, como o uso de modelos matemáticos, para conceitos como a complexidade do software, o crescimento da confiabilidade, dentre outros.

Dentro da área da engenharia de software experimental algumas estratégias para realização de estudos empíricos são amplamente utilizadas, como por exemplo, *survey*, estudo de caso, experimento e RSL (WOHLIN et al., 2012).

Um *survey* é utilizado para coletar informações de sujeitos para descrever, comparar ou explicar seu conhecimento sobre um determinado assunto. Nesse tipo de estratégia, é realizada a coleta de dados por intermédio de informações qualitativas ou quantitativas por meio de questionários ou entrevistas.

Segundo Yin (2013), um estudo de caso é uma investigação empírica, apoiada em múltiplas fontes de evidência, para investigar uma instância de um fenômeno da Engenharia de Software em um contexto real. Geralmente, estudos de caso são utilizados na pesquisa de projetos, atividades ou atribuições, onde os dados são coletados com um

propósito específico. É comum que as pesquisas com estudos de caso sejam conduzidas pelo método observacional, ou seja, o pesquisador participa pessoalmente coletando informações durante o processo.

Um experimento controlado é um estudo empírico que manipula um fator ou variáveis de um cenário estudado.

Uma RSL visa coletar e sintetizar evidências empíricas a partir de diferentes fontes.

2.3.1 Revisão Sistemática da Literatura

De acordo com Kitchenham (2007), uma revisão sistemática da literatura tem como principal objetivo identificar, analisar e interpretar as evidências disponíveis na literatura para uma determinada questão de pesquisa. Para garantir uma maior confiabilidade e possibilidade de replicação, uma RSL deve seguir um protocolo rigoroso que estabeleça alguns critérios. O trabalho de Kitchenham (2007) estabelece diretrizes para serem seguidas para execução de uma RSL. Estas diretrizes são divididas em três etapas: planejamento, execução e relato dos resultados.

No planejamento, são definidos os aspectos principais sobre a RSL a ser executada. A primeira etapa do planejamento é a identificar qual a necessidade da revisão a ser realizada, em que o pesquisador busca entender o estado da arte de uma determinada área. A partir disso, são elaboradas as questões de pesquisa e é definido o protocolo a ser utilizado

Durante a execução ocorre a instanciação do que foi definido durante o planejamento. Nessa etapa são especificadas as *strings* de busca e aplicadas nas bases definidas, fazendo a seleção dos trabalhos primários e aplicando os critérios de qualidade. Em seguida, a partir dos trabalhos, são extraídos e apresentados os dados relevantes para a pesquisa.

Na etapa de relato dos resultados o que foi encontrado por meio da RSL deve ser reportado. Deve-se levar em consideração a audiência alvo do estudo, apresentando os dados extraídos de maneira adequada. No Capítulo 3 é apresentada em maiores detalhes a RSL realizada neste trabalho.

2.3.2 Experimento Controlado

Um experimento é um estudo empírico que manipula um fator ou variáveis de um cenário estudado. Baseado em randomização, diferentes tratamentos são aplicados para diferentes sujeitos, enquanto mantém outras variáveis constantes e mede seus efeitos. Em experimentos com pessoas, os sujeitos aplicam diferentes tratamentos para determinados objetos, enquanto em experimentos com tecnologia aplicam diferentes técnicas e tratamentos para diferentes objetos (WOHLIN et al., 2012). Em geral, as experiências são executadas em um laboratório controlado, sendo esse tipo de execução caracterizada

como *in-vitro*. Também é possível a execução de experimentos em ambientes reais da indústria, neste caso são caracterizada como *in-loco* (YIN, 2013).

A primeira etapa em um experimento é a definição do escopo, em que são definidos os seus objetivos. Essa definição de objetivos visa garantir que os aspectos mais importantes do experimento estão sendo contemplados.

Após o escopo estar definido, o planejamento é iniciado. Enquanto o escopo determina os fundamentos principais do experimento, o planejamento detalha como o experimento deve ser conduzido. No planejamento são realizadas as atividades de seleção do contexto, formulação das hipóteses, seleção de variáveis e sujeitos, escolha do tipo de projeto, instrumentação e avaliação da validade do experimento.

Na operação do experimento, o que foi definido no planejamento é executado. A operação engloba as atividades de preparação, execução e validação dos dados do experimento.

Em seguida, é realizada a análise e interpretação dos dados. Essa fase envolve a realização dos testes de hipóteses nas amostras, para que seja possível obter conclusões a partir dos dados coletados.

Para finalizar o experimento, os resultados são reportados por meio de um relato que contém o detalhamento do experimento e dos resultados.

No Capítulo 5 é apresentado em maiores detalhes o experimento controlado realizado neste trabalho.

2.4 Lições do Capítulo

Neste capítulo é apresentada a fundamentação teórica deste trabalho. Esta fundamentação serviu como base de apoio às demais partes do trabalho. Os principais pontos abordados neste capítulo são:

- Definições gerais de ER, EROA e Interesses Transversais.
- Definições gerais de Engenharia de Software Experimental.
- Visão geral sobre RSL.
- Visão geral sobre experimento controlado.

3 REVISÃO SISTEMÁTICA DA LITERATURA

Este capítulo tem por finalidade apresentar a RSL realizada para investigação da área de estudo deste trabalho. O foco principal dessa RSL foi encontrar trabalhos que propõem abordagens para identificação de interesses de software durante a ER. Na Seção 3.1 são apresentados os trabalhos relacionados. Na Seção 3.2 é detalhado o protocolo da revisão, apresentando as questões de pesquisa, o processo de busca, seleção e análise dos dados. A Seção 3.3 detalha a execução da RSL e a Seção 3.4 faz uma análise do resultado do estudo. As lições do capítulo são apresentadas na Seção 3.5.

3.1 Trabalhos Relacionados

Antes da execução da RSL, foi realizada uma busca na literatura por trabalhos que propusessem RSLs relacionadas à área de Interesses Transversais e abordagens para EROA. A busca foi realizada de maneira *ad hoc*.

No trabalho de Durelli et al. (2013) foi realizada uma RSL sobre técnicas para mineração de Interesses Transversais. Através desse estudo foram identificadas 18 técnicas para mineração desses interesses. Como este trabalho é voltado à descoberta de Interesses Transversais na ER, mais especificamente a área de EROA, o trabalho de Durelli et al. (2013) tem um foco diferente do que foi buscado para este trabalho.

No trabalho de Ali et al. (2010), foi realizada uma RSL sobre evidências na literatura sobre a favor da OA em detrimento de outros paradigmas. Para análise dos resultados os estudos foram categorizados para que refletissem os possíveis benefícios de acordo com 6 critérios: desempenho, tamanho do código, modularidade, evolabilidade, congñição e mecanismo de linguagem. O foco desse trabalho, apesar englobar muitos conceitos da área de Interesses Transversais, também difere do objetivo principal da RSL desejada para este trabalho, em que o foco é a identificação de *early aspects*.

Outra RSL relacionada à área de Interesses Transversais foi relatada no trabalho de Silva, Maia e Soares (2014). Essa revisão foca na avaliação do desempenho de técnicas de programação usadas para implementação desses interesses. Como o desejado eram trabalhos que focassem em etapas anteriores à implementação, essa RSL não agrega muitas contribuições a este trabalho.

A RSL proposta por Marçal et al. (2016) analisa as técnicas, relatadas na literatura, para identificação de Interesses Transversais. Apesar do título do trabalho remeter às técnicas para identificação, na prática o estudo foi direcionado à área de Mineração de Aspectos, que por sua vez, foca na extração de Interesses Transversais a partir de linhas de código. Com essa revisão é possível identificar as principais abordagens para identificação de Interesses Transversais a nível de implementação e não especificamente para a EROA.

Os trabalhos de Ali e Kasirun (2008b) e Alshareef, Maatuk e Abdelaziz (2018) apresentam algumas das principais técnicas e abordagens para identificação de Interesses

Transversais na ER. Porém, o primeiro foi executado em 2008 e já surgiram novas abordagens após o trabalho. Já o segundo apesar de ser atual, não segue uma metodologia bem definida para a realização da busca dos trabalhos.

Apesar de alguns estudos indicarem metodologias voltadas à área de EROA, ainda assim se faz necessária uma RSL, tendo em vista o lapso de tempo entre alguns trabalhos e também o fato de poder instanciar a RSL seguindo um protocolo que possibilite sua replicação.

3.2 Protocolo da RSL

A metodologia seguida para a realização da RSL segue o protocolo proposto por Kitchenham (2007).

3.2.1 Questões de Pesquisa

O escopo da revisão foi investigar estudos primários que propusessem abordagens para identificação de Interesses Transversais para ER.

Foram definidas três questões de pesquisa para o estudo:

- *Q1. Quais as motivações dos trabalhos relacionados a Interesses Transversais na Engenharia de Requisitos?*
- *Q2. Quais os problemas relacionados a Interesses Transversais na Engenharia de Requisitos?*
- *Q3. Quais soluções são mais relevantes para a identificação de Interesses Transversais na Engenharia de Requisitos?*

Com relação às questões de pesquisa, a questão 1 foi definida com o objetivo de mapear as motivações relatadas nos trabalhos da área de estudo. Já a questão 2 tem por finalidade descobrir as formas de resolução dos problemas da área. Por fim, a questão 3 busca identificar, dentre as soluções descobertas, quais as mais relevantes no contexto de aplicação de um experimento controlado, ou seja, nesse caso a relevância está relacionada com a possibilidade de experimentação.

3.2.2 Processo de Busca

A partir das questões de pesquisa, o processo de busca por trabalhos foi realizado nas seguintes bases:

- *Association for Computing Machinery (ACM (2017));*
- *IEEE Explore Digital Library (IEEE (2017));*

- *Elsevier Science Direct* (Elsevier (2017));
- *Springer International Publishing* (Springer (2017));

As bases foram escolhidas por apresentarem grande parte dos trabalhos disponíveis na literatura e serem referência para as publicações na área da computação.

3.2.2.1 Association for Computing Machinery

Na base da ACM foram incluídos os seguintes filtros: *English, Computer Science, Software Engineering/Programming and Operating, Systems* e *Article*. Foi utilizada a seguinte *String* de busca: (“*crosscutting concern*” “*cross-cutting concerns*” “*crosscutting concerns*”) AND (*concern concerns*) AND (*aspect aspects*) AND (*identification “identification method” “identification approach”*).

3.2.2.2 IEEE Explore Digital Library

Na base da IEEE não foram incluídos filtros adicionais, sendo usada a seguinte *string* de busca: (“*crosscutting concern*” OR “*cross-cutting concerns*” OR “*crosscutting concerns*”) AND (*concern OR concerns*) AND (*aspect OR aspects*) AND (*identification OR “identification method” OR “identification approach”*).

3.2.2.3 Elsevier Science Direct

Na base da *Elsevier* foram aplicados os filtros: *Review Articles, Research Articles, Journal of Systems and Software* e *Information and Software Technology*. Foi utilizada a seguinte *string* de busca: (“*crosscutting concern*” OR “*cross-cutting concerns*” OR “*crosscutting concerns*”) AND (*concern OR concerns*) AND (*aspect OR aspects*) AND (*identification OR “identification method” OR “identification approach”*).

3.2.2.4 Springer International Publishing

Na base da *Springer* não foram utilizados filtros adicionais, sendo usada a seguinte *string* de busca: (“*crosscutting concern*” OR “*cross-cutting concerns*” OR “*crosscutting concerns*”) AND (*concern OR concerns*) AND (*aspect OR aspects*) AND (*identification OR “identification method” OR “identification approach”*).

3.2.3 Processo de Seleção

Para a realização da seleção dos trabalhos foram definidos critérios. Quando um determinado trabalho atende a um critério de exclusão, o mesmo deve ser excluído do processo. Os critérios de inclusão, por sua vez, definem características necessárias para o trabalho ser selecionado. Critérios de qualidade buscam refinar a busca, possibilitando

encontrar os trabalhos mais relevantes para o objetivo da RSL (WOHLIN et al., 2012). Se o trabalho atende a todos os critérios de qualidade, deve ser selecionado. Os seguintes critérios foram considerados para a seleção dos trabalhos:

a) **Critérios de Exclusão:**

- CE-1: Trabalhos escritos em língua diferente do inglês.
- CE-2: Trabalhos com menos de 6 páginas.

b) **Critérios de Inclusão:**

- CI-1: Trabalhos primários¹ e completos que propõem soluções relacionadas a Interesses Transversais na ER.

c) **Critérios de Qualidade:**

- CQ-1: O trabalho possui uma metodologia bem definida?
- CQ-2: A proposta do trabalho contempla requisitos funcionais e requisitos não funcionais?
- CQ-3: O trabalho apresenta algum tipo de avaliação?
- CQ-4: O trabalho mostra um exemplo de aplicação bem definido?

Como estratégia de avaliação, para os critérios de exclusão, foram excluídos os trabalhos que satisfazem pelo menos um dos critérios. Com relação aos critérios de inclusão, foram incluídos os trabalhos que atendem ao critério de inclusão descrito. Quanto aos critérios de qualidade, as perguntas deveriam ser respondidas com SIM ou NÃO, sendo selecionados apenas os trabalhos que obtiveram resposta SIM para todos os critérios de qualidade.

O processo de seleção foi dividido em cinco etapas: busca inicial, exclusão de redundâncias, aplicação de critérios de exclusão, aplicação de critérios de inclusão e aplicação de critérios de qualidade.

3.2.4 Coleta e Análise de Dados

Para a coleta e análise dos dados dos trabalhos foram extraídas informações do título, resumo e introdução dos artigos. À medida que os artigos iam sendo lidos, as informações relevantes eram transcritas para uma ficha de leitura. Essa ficha continha os campos título, onde deveria constar o título do trabalho, objetivo, que deveria mostrar de maneira resumida o objetivo do artigo, motivação, para compilar a(s) motivação(ões) do trabalho e proposta, para identificar quais soluções os trabalhos propõem para os problemas identificados.

¹ Segundo Kitchenham (2007), um trabalho primário é um estudo empírico que investiga uma questão de pesquisa específica.

Por meio do campo motivação da ficha de leitura, foram obtidas as informações que possibilitaram responder à questão 1. Essas informações foram extraídas e agrupadas por similaridade semântica.

Com relação às questões 2 e 3, que abordam os problemas e as soluções propostas para a área, foram coletadas informações dos campos objetivo e proposta, para que as questões fossem respondidas. Além disso, adicionalmente foi realizada a leitura completa dos trabalhos para que houvesse um entendimento maior das propostas.

3.3 Execução da RSL

A RSL foi realizada entre agosto e novembro de 2018. Na busca inicial foram executadas as *strings* nas bases, onde foram retornados 204 trabalhos, 18 da ACM, 45 da IEEE, 57 da *Springer* e 84 da *Elsevier*. Foram removidos 5 trabalhos duplicados na etapa para exclusão de redundâncias. Com a aplicação dos critérios de exclusão restaram 177 trabalhos, 12 da ACM, 25 da IEEE, 56 da *Springer* e 84 da *Elsevier*. Em seguida, com a aplicação dos critérios de inclusão houve uma redução para 19 trabalhos no total, 4 da ACM, 6 da IEEE, 4 da *Springer* e 5 da *Elsevier*. Por fim, foram aplicados os critérios de qualidade, com isso foram selecionados 6 trabalhos, 3 da IEEE e 3 da *Springer*. Os trabalhos selecionados são os que atendem às questões de pesquisa e aos critérios definidos. Na Tabela 3 são apresentados os resultados obtidos com a busca, de acordo com as etapas definidas, sendo as fases de 1 a 5 obedecendo a ordem de execução: busca inicial, remoção de redundâncias, aplicação de critérios de exclusão, aplicação de critérios de inclusão e aplicação dos critérios de qualidade.

Tabela 2 – Resultado da busca nas bases.

Base	Fase 1	Fase 2	Fase 3	Fase 4	Fase 5
ACM	18	13	12	4	0
IEEE	45	45	25	6	3
Springer	57	57	56	4	3
Elsevier	84	84	84	5	0
Total	204	199	177	19	6

Fonte: Elaborado pelo autor.

Na Tabela 3 estão elencados os resultados da aplicação dos critérios de qualidade nos trabalhos. Essa foi a última etapa antes da escolha dos trabalhos selecionados.

3.4 Análise do Resultado

A partir da coleta das informações relevantes dos trabalhos selecionados, foi possível visualizar os pontos em comum entre eles e como essas informações podem responder às questões de pesquisa da RSL realizada.

Tabela 3 – Aplicação dos critérios de qualidade.

ID	Trabalho	CQ-1	CQ-2	CQ-3	CQ-4
1	Marques, Moreira e Araújo (2008)	X		X	X
2	Kong e Yuan (2009)	X		X	X
3	Antonelli, Rossi e Leite (2010)	X	X		X
4	Sardinha et al. (2012)	X		X	X
5	Marew, Lee e Bae (2009)	X		X	X
6	Nakagawa et al. (2011)	X	X		X
7	d’Amorim e Borba (2012)	X		X	X
8	Henderson-Sellers et al. (2007)	X			X
9	Sánchez et al. (2010)	X		X	X
10	Amel e Amin (2016)	X	X		X
11	Razzaq e Abbasi (2012)	X	X	X	X
12	Ali e Kasirun (2008a)	X	X		X
13	Velan e Priya (2016)	X	X		X
14	He e Tu (2013)	X	X	X	X
15	Baniassad e Clarke (2004)	X	X	X	X
16	Rosenhainer (2005b)	X	X	X	X
17	Sampaio et al. (2005)	X	X	X	X
18	Júnior e Penteado (2018)	X	X	X	X
19	Marin, Moonen e Deursen (2007)	X	X		X

Fonte: Elaborado pelo autor.

3.4.1 Q1. Quais as motivações dos trabalhos relacionados a Interesses Transversais na Engenharia de Requisitos?

A Tabela 4 mostra as principais motivações dos trabalhos relacionados a Interesses Transversais na ER. As motivações descritas variam de mineração de Interesses Transversais a erros nas abordagens existentes para identificação de interesses.

Tabela 4 – Motivações dos trabalhos selecionados.

ID	Descrição
1	Mineração de Interesses Transversais
2	Identificação de <i>early aspects</i>
3	Dificuldade de identificação de Interesses Transversais na fase de análise.
4	Rastreabilidade entre interesses de software.
5	Interesses Transversais a partir de Requisitos Não Funcionais.
6	Classificação e modularização de Interesses Transversais.
7	Erros nas abordagens existentes para identificação de Interesses Transversais.

Fonte: Elaborado pelo autor.

3.4.2 Q2. Quais os problemas relacionados a Interesses Transversais na Engenharia de Requisitos?

a Tabela 5 relaciona os trabalhos selecionados com as motivações. Com essa análise é possível verificar que os principais problemas descritos na área sob estudo são os relacionados à mineração de interesses transversais e à identificação de *early aspects*, por serem os mais recorrentes nos trabalhos selecionados.

Tabela 5 – Síntese do resultado.

Trabalho	Motivações	Proposta
Razzaq e Abbasi (2012)	1	Abordagem baseada em OA
He e Tu (2013)	1, 2	Método baseado em <i>goal model</i>
Baniassad e Clarke (2004)	2, 3, 4	Abordagem baseada em temas
Rosenhainer (2005b)	5	Abordagem baseada em casos de uso
Sampaio et al. (2005)	1, 2, 3	Abordagem baseada em PLN
Júnior e Penteado (2018)	1, 6, 7	Abordagem baseada em ontologia

Fonte: Elaborado pelo autor.

3.4.3 Q3. Quais soluções são mais relevantes para a identificação de Interesses Transversais na Engenharia de Requisitos?

Com base na análise realizada nos trabalhos selecionados, foi possível verificar que os 6 trabalhos selecionados propõem abordagens para identificação de interesses de software na ER. Essas abordagens possibilitam a identificação desses interesses seguindo uma determinada metodologia. As abordagens propostas pelos trabalhos selecionados estão descritas em detalhes no Capítulo 4. Na Tabela 6 é possível visualizar as propostas dos trabalhos, todos propõem abordagens, exceto o de He e Tu (2013), que propõe um método. Porém, há apenas uma diferença na nomenclatura usada pelo autor, uma vez que o método proposto também pode ser considerado uma abordagem. As abordagens propostas são as seguintes: *AOFA*, *AspectQuery*, *DISCERN*, *EA-Miner*, *ObasCId* e *Theme/Doc*.

Para verificar as soluções mais relevantes para a identificação de Interesses Transversais na ER foi analisado o resultado da aplicação dos critérios de qualidade definidos. Essa análise pode ser vista na Tabela 7 onde é possível visualizar que os 6 trabalhos selecionados atendem a todos os critérios de qualidade definidos.

3.5 Lições do Capítulo

Neste capítulo é apresentada a RSL conduzida para obtenção das principais abordagens para identificação de Interesses Transversais na ER. Os principais pontos aborda-

Tabela 6 – Soluções propostas.

Trabalho	Nome	Proposta
Razzaq e Abbasi (2012)	<i>AOFA</i>	Abordagem baseada em OA
He e Tu (2013)	<i>AspectQuery</i>	Método baseado em <i>goal model</i>
Baniassad e Clarke (2004)	<i>Theme/Doc</i>	Abordagem baseada em temas
Rosenhainer (2005b)	<i>DISCERN</i>	Abordagem baseada em casos de uso
Sampaio et al. (2005)	<i>EA-Miner</i>	Abordagem baseada em PLN
Júnior e Penteado (2018)	<i>ObasCID</i>	Abordagem baseada em ontologia

Fonte: Elaborado pelo autor.

Tabela 7 – Aplicação dos critérios de qualidade nos trabalhos selecionados.

ID	Trabalho	CQ-1	CQ-2	CQ-3	CQ-4
1	Razzaq e Abbasi (2012)	X	X	X	X
2	He e Tu (2013)	X	X	X	X
3	Baniassad e Clarke (2004)	X	X	X	X
4	Rosenhainer (2005b)	X	X	X	X
5	Sampaio et al. (2005)	X	X	X	X
6	Júnior e Penteado (2018)	X	X	X	X

Fonte: Elaborado pelo autor.

dos neste capítulo são:

- Apresentação de trabalhos relacionados que propõem revisões semelhantes.
- Detalhamento do protocolo utilizado na execução da RSL.
- Apresentação e discussão dos resultados obtidos na RSL realizada.

4 CATÁLOGO DE ABORDAGENS PARA IDENTIFICAÇÃO DE INTERESSES

Este capítulo tem por finalidade aprofundar o estudo das abordagens selecionadas na RSL apresentada no Capítulo 3 de forma a dar subsídios para a escolha das abordagens a serem utilizadas no experimento controlado proposto neste trabalho. Na Seção 4.1 é apresentado o detalhamento das abordagens. Na Seção 4.2 é reportada a análise qualitativa das abordagens. Por fim, na 4.3 são apresentadas as lições do capítulo.

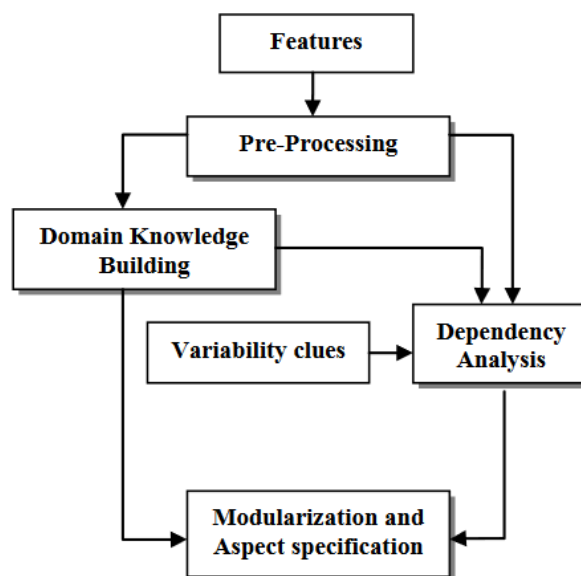
4.1 Detalhamento das Abordagens

Na EROA, abordagens definem técnicas e metodologias para identificação de Interesses Transversais. Diferentes alternativas são propostas para esta finalidade. A seguir são apresentadas em detalhes as seguintes abordagens: *AOFA*, *AspectQuery*, *DISCERN*, *EA-Miner*, *ObasCIId* e *Theme/Doc*.

4.1.1 Aspect-Oriented Feature Analysis

Essa abordagem foi proposta por Razzaq (2012) e tem como objetivo obter informações por meio de processamento de linguagem natural usando técnicas como *stemming*, análise de frequência, *parts-of-speech tagging*, que são técnicas utilizadas para a extração de informações a partir de um determinado texto. A Figura 3 exibe as fases da abordagem. A entrada são as *features* que são as características e termos relevantes para o domínio.

Figura 3 – Etapas da abordagem AOFA.



Fonte: Razzaq (2012).

Na fase *Pre-Processing* é feita uma filtragem para eliminar possíveis falsos positivos, é feito o *stemming*, que consiste em reduzir os termos para a sua raiz. Em seguida, é construído um vocabulário com frequência associativa, onde cada *feature* é representada por um vetor de termos juntamente com sua frequência de ocorrência. Por fim, é realizada a análise de pares, que visa analisar em pares o vocabulário com a descrição e o vocabulário com o título das *features*.

Na fase *Domain Knowledge Extraction* é realizada a construção de uma base de conhecimento do domínio, incluindo os requisitos funcionais e requisitos não funcionais do sistema, com as informações obtidas por meio da análise do domínio. Nessa etapa é feita a análise de *POS (Part-of-speech tagging)* para separar as funções gramaticais. Com isso é criado o *Term Frequency Diagram (TFD)* que contém os termos juntamente com a frequência que aparecem.

A fase *Dependencies and Aspectual Relations Identification* visa padronizar uma base para análise de dependência entre os termos gerando como saída um modelo para visualização das ações similar a *Action View* da abordagem *Theme/Doc*.

A última fase da abordagem AOFA é a *Modularization and Concerns Association* que tem por finalidade modularizar as descrições de acordo com a variabilidade que apresentam, atribuindo Interesses Transversais relacionados com as mesmas, ou seja, é realizada a leitura do artefato de visualização das ações.

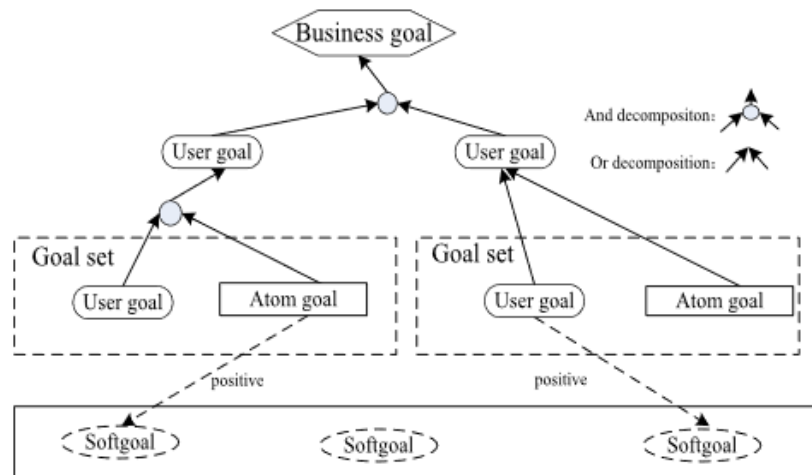
4.1.2 AspectQuery

A abordagem *AspectQuery* foi proposta no trabalho de Chengwan (2013) e é baseada em um modelo de objetivos (*goal model*) que é convertido para um arquivo de extensão XML.

A primeira atividade a ser realizada nessa abordagem é a construção do *goal model* onde o engenheiro de software deve criá-lo de acordo com as especificações do sistema. A Figura 4 exibe um exemplo desse artefato que contém os seguintes termos:

- *Business goal* (hexágono) – Objetivo geral do sistema;
- *User goal* (retângulo arredondado) – Descreve em alto nível o que o usuário espera que o sistema realize;
- *Atom goal* (retângulo) – São os user goals refinados, descrevem um comportamento do sistema;
- *Goal Set* (retângulo pontilhado) – Descreve um conjunto de *user goal* e *atom goal* que não pode mais ser refinado durante a Engenharia de Requisitos;
- *Soft-goals*(elipses) – Requisitos não-funcionais.

Figura 4 – Exemplo de Goal Model.



Fonte: Chengwan (2013).

Em seguida, é feita a análise do modelo onde há quatro possibilidades para identificação de Interesses Transversais:

1. **Um objetivo é positivo para o soft-goal:** Neste caso os *goals* que estiverem ligados ao *soft-goal* são Interesses Transversais.
2. **Um *goal* está relacionado com outros *goals* acima:** Neste caso o *goal* é Interesse Transversal, pois se relaciona com mais de um nível acima dele.
3. **Um *goal* está relacionado com outros *goals* acima:** Igual à anterior, porém depende da implementação de um deles (usa a decomposição OU).
4. **Um *usergoal* é filho de outro *usergoal*:** Aqui há uma relação entre os *usergoals*.

A segunda atividade da abordagem *AspectQuery* é a transformação do *Goal Model* em um arquivo XML. Essa transformação é realizada manualmente pelo engenheiro de software por intermédio de um algoritmo, seguida da identificação dos Interesses Transversais por meio de regras. O algoritmo para transformação e as regras também foram propostos no trabalho de Chengwan (2013).

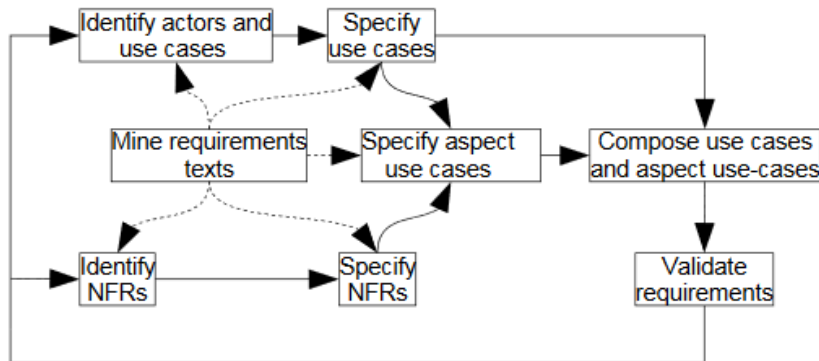
A última atividade da abordagem consiste na composição dos Interesses Transversais onde é possível identificar os *goals* que são afetados por cada Interesse Transversal. Segundo Chengwan (2013) essa atividade é realizada pela ferramenta *XQuery*.

4.1.3 DISCERN

Abordagem proposta por Rosenhainer (2005a) para identificação de Interesses Transversais baseada em casos de uso. A Figura 5 exhibe a visão geral da abordagem

que possui as seguintes atividades:

Figura 5 – Visão Geral da abordagem DISCERN.



Fonte: Chengwan (2013).

A primeira atividade, *Identify actors and use cases*, extrai os atores e casos de uso a partir da análise do domínio ou do documento de requisitos do software. Em seguida, os casos de uso são especificados como cenários e os requisitos não funcionais são extraídos pelo processamento de linguagem natural. Após isso, são construídos os *Aspect Use Cases* que são os casos de uso dos aspectos, com a ideia de aspecto vinda do paradigma da Orientação a Aspectos. Por fim, na atividade *Compose and Validade* são resolvidos os conflitos entre os aspectos e são gerados os diagramas de casos de uso integrados com os aspectos.

4.1.4 EA-Miner

A abordagem/ferramenta *EA-Miner* foi proposta por Sampaio e Rayson (2005) e proporciona a extração automatizada de ITs a partir de um documento de requisitos. Essa abordagem possui 3 passos:

1. *Identificação*: É feito o envio do arquivo de requisitos na ferramenta. A ferramenta envia para o *WMatrix*, que é o processador de linguagem natural utilizado, que fará a análise *POS - Part of speech* e *Semantic Tagging* (marcação semântica). Após isso, o processador *WMatrix* gera um arquivo XML a partir do documento que foi enviado juntamente com os dados da análise realizada. O documento de requisitos é enviado via protocolo HTTP e implementa o padrão de projeto *Observer* para rastrear o estado do processo de *tagging*.
2. *Apresentação dos resultados*: Neste passo o engenheiro de software visualiza os resultados gerados por meio de diagramas e representação textual. Na ferramenta só são apresentados os dados textuais relacionando os Interesses Transversais. Após isso, a ferramenta gera os *Java Objects* relacionados aos termos gerados.

3. *Refinamento dos resultados*: Neste passo os modelos gerados são refinados para exclusão de dados irrelevantes.

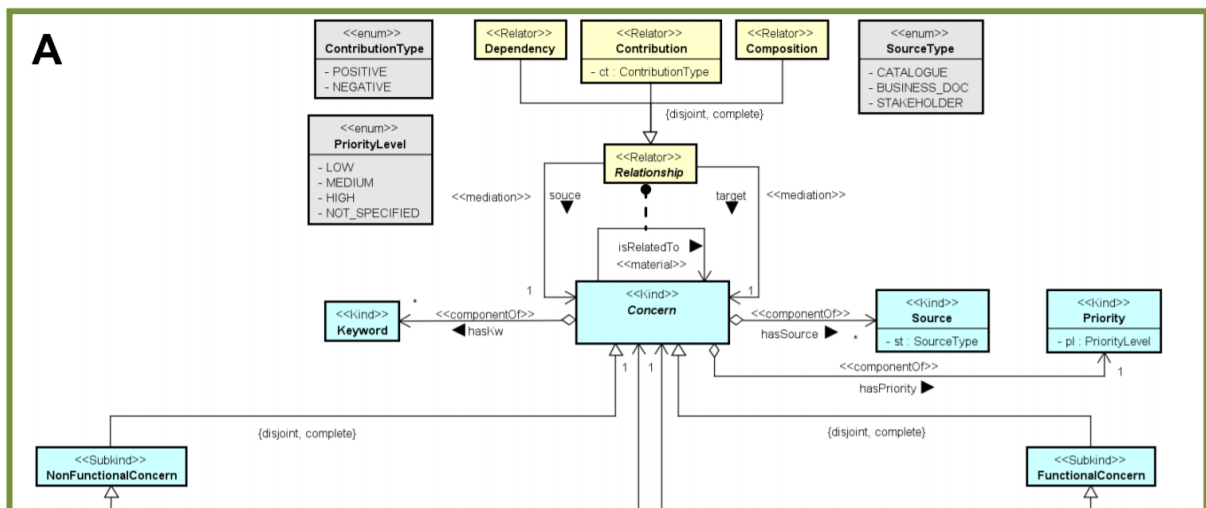
Segundo os desenvolvedores da *EA-Miner*, a mesma se encontra descontinuada e não é possível obtê-la para uso.

4.1.5 ObasCID

A abordagem *ObasCID* (*Ontologically based Concern Identification*) foi proposta por Parreira Jr. (2018). Essa abordagem é baseada em uma ontologia, proposta no mesmo trabalho, para a identificação dos interesses de software. A ontologia base dessa abordagem é chamada *O4C* (*Ontology for Concerns*) e representa os conceitos e relacionamentos bem conhecidos e já documentados na literatura sobre interesses de software no contexto da EROA.

Para facilitar o entendimento dessa ontologia, a mesma é dividida em duas partes principais, A e B. Na Figura 6 está a parte A da ontologia, que visa especificar os conceitos e relacionamentos independentes do contexto sobre qual o processo de EROA está sendo executado. Já na Figura 7 está a parte B da ontologia que visa especificar os conceitos e relacionamentos que dependem do contexto sobre qual o processo de EROA está sendo executado.

Figura 6 – Parte A da ontologia O4C.

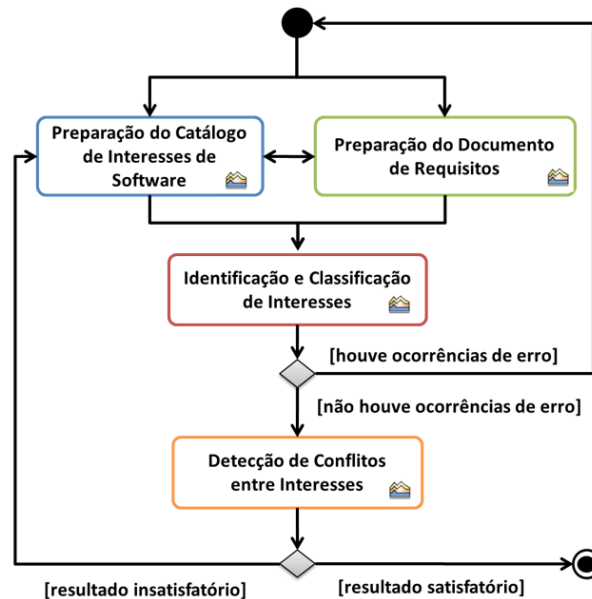


Fonte: Parreira Jr. (2018).

A seguir são explicados mais detalhadamente os principais conceitos envolvidos nessa ontologia:

- **Concern, FunctionalConcern e NonFunctionalConcern**: Representam os indivíduos que atendem às propriedades estabelecidas para um interesse de software

Figura 8 – Fases da ObasCId.



Fonte: Parreira Jr. (2018).

catálogos de interesses de software são gerados a partir dos conceitos da parte A da ontologia *O4C*, Figura 6, correspondendo às instâncias desses conceitos. Na Figura 9 é apresentado um exemplo simples de um catálogo, onde os interesses não funcionais são *Logging*, *Segurança* e *Autorização*. São utilizadas associações do diagrama de classes UML para representar os relacionamentos entre as instâncias.

Figura 9 – Exemplo de um Catálogo de Interesses de Software.



Fonte: Parreira Jr. (2018).

Preparação do Documento de Requisitos do Software: Consiste em criar o documento de requisitos que será utilizado para a identificação de interesses. Na Tabela 8 é exibido o modelo de documento de requisitos adotado pela abordagem *ObasCId*. Nesse modelo há 4 colunas, a primeira mostra o código do requisitos. Na segunda coluna é referenciado se o requisito é do tipo funcional (RF) ou não funcional (RNF). Na terceira coluna é feita a descrição do requisito e finalizando com os requisitos relacionados na

última coluna, onde é colocada uma relação de requisitos que possuem ligação com o requisito em questão.

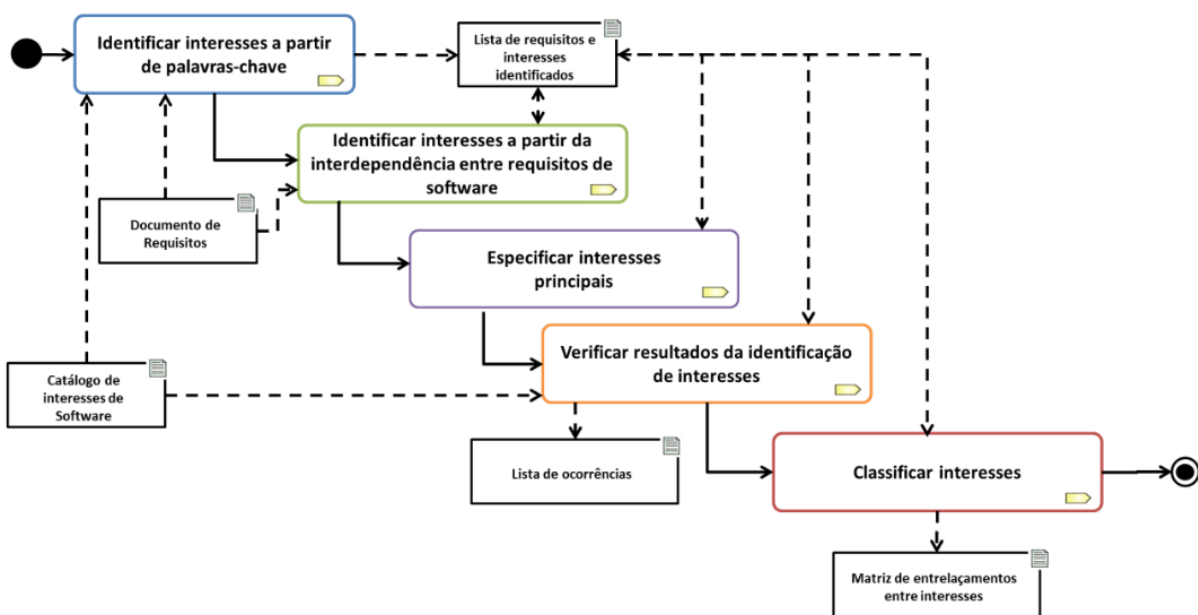
Tabela 8 – Modelo de documento de requisitos adotado pela abordagem ObasCId.

Cód	Tipo	Descrição do Requisito	Requisitos Relacionados
1	RF	Esta é a descrição do requisito funcional 1.	2
2	RNF	Esta é a descrição do requisito não funcional 2.	-
...
<N>	<tipo>	Esta é a descrição do requisito <tipo> <N>.	-

Fonte: Adaptado de Parreira Jr. (2018).

Identificação e Classificação de Interesses: Essa fase visa identificar e classificar os interesses representados no catálogo. Os interesses após identificados podem ser classificados como Interesses Base ou Interesses Transversais. A Figura 10 mostra as atividades realizadas nessa fase.

Figura 10 – Atividades da fase de Identificação e Classificação de Interesses.



Fonte: Parreira Jr. (2018).

Em um primeiro momento é feita a **identificação a partir de palavras-chave** que são buscadas no catálogo de interesses criado anteriormente. Caso alguma das palavras-chave de um determinado interesse apareça na descrição de um requisito de software, diz-se que esse interesse está relacionado com o requisito em questão. Na Tabela 9 está um exemplo de um documento de requisitos com as palavra-chave já identificadas.

As palavras-chave estão sublinhadas e em negrito para melhor visualização. Nesse exemplo, o RF-01 tem destacado *atualização*, *reclamação* e *logado*. Essas palavras permitem a identificação dos interesses *Persistência*, *Segurança* e *Reclamação*.

Tabela 9 – Exemplo de documento de requisitos com as palavras-chave identificadas.

Requisito RF-01	Interesses	Principal
Tem como propósito realizar a <u>atualização</u> do andamento de uma <u>reclamação</u> . A <u>reclamação</u> deve estar cadastrada, com a situação ABERTA e com o funcionário <u>logado</u> no sistema.	Persistência	
	Segurança	
	Reclamação	X
	Desempenho	
	Usabilidade	
Requisito RNF-02	Interesses	Principal
O sistema deve ter uma interface de <u>fácil utilização</u> , visto que o sistema pode ser utilizado por qualquer pessoa que tenha acesso à internet. O sistema deve ter um <u>HELP</u> on-line para ser consultado por qualquer pessoa que acesse o sistema.	Usabilidade	X
Requisito RNF-03	Interesses	Principal
O <u>tempo de resposta</u> do sistema não deve ultrapassar 5 segundos por acesso.	Desempenho	X

Fonte: Adaptado de Parreira Jr. (2018).

A segunda atividade dessa fase é a **identificação a partir da análise de interdependência entre requisitos**. Essa atividade é necessária pois há alguns interesses em que não é possível a identificação apenas por meio de palavras-chave. Sendo assim, nessa atividade é analisado o documento de requisitos para buscar interesses que impactem em um ou mais requisitos. Por exemplo, na Tabela 9, no requisito RNF-03, verifica-se que o mesmo tem impacto em todas as funcionalidades do software. Por isso, o interesse desempenho deverá estar mapeado em todos os requisitos que remetam à funcionalidades.

Em seguida, é realizada a atividade de **especificar os interesses principais** onde é assinalado no documento de requisitos qual o interesse principal de cada requisito. A utilidade da especificação dos interesses principais está relacionada ao processo de classificação em interesses base ou transversais, pois uma vez que um requisito para uma finalidade (interesse) principal, as demais finalidades podem ser consideradas como um comportamento transversal ao interesse (BANIASSAD, 2004).

Após a especificação dos interesses principais, é feita a **verificação dos resultados da identificação de interesses**. Essa atividade visa achar problemas relacionados à identificação de interesses, bem como possíveis soluções para esses problemas.

Finalizando essa fase, é feita a **classificação dos interesses**. Nessa atividade é gerada uma matriz de entrelaçamentos entre interesses a partir do documento de requisitos

e dos interesses identificados. A Tabela 10 mostra a matriz de entrelaçamentos entre interesses. Os interesses principais estão descritos nas linhas e os interesses transversais são detectados nas colunas. Por exemplo, ao analisar a Tabela 9 é possível verificar que o interesse mais afetado por outros interesses é *Reclamação*, que é entrecortado por *Persistência*, *Segurança*, *Usabilidade* e *Desempenho*, o que se reflete nas células marcadas com um X na Tabela 10. Nesse exemplo temos os interesses principais *Persistência*, *Segurança*, *Reclamação*, *Usabilidade* e *Desempenho*, todos também interesses transversais, exceto *Reclamação*.

Detecção de Conflitos entre Interesses: Busca por possíveis conflitos existentes entre os interesses, com base na influência mútua que um pode exercer sobre o outro. Um conflito entre interesses ocorre quando há influência negativa de um interesse sobre outro e os dois foram identificados no software (PARREIRA Jr., 2018). Sendo assim, o engenheiro de software pode tomar algumas decisões:

- assumir o impacto do interesse e prosseguir no desenvolvimento;
- descartar o interesse que impacta negativamente em outro interesse;
- substituir o interesse que impacta negativamente em outro interesse.

Tabela 10 – Exemplo de uma matriz de entrelaçamentos entre interesses.

↓ IP/IT →	1	2	3	4	5
1. Persistência					
2. Segurança					
3. Reclamação	X	X		X	X
4. Usabilidade					
5. Desempenho					

Fonte: Adaptado de Parreira Jr. (2018).

4.1.6 Theme/Doc

Theme é uma abordagem para identificação de interesses para EROA. Foi proposta por Baniassad (2004) e é dividida em dois níveis, *Theme/Doc* que fornece mecanismos para a visualização de interesses por meio de *Action Views* e *Theme/UML* que possibilita ao engenheiro de software a modelagem dos interesses de software.

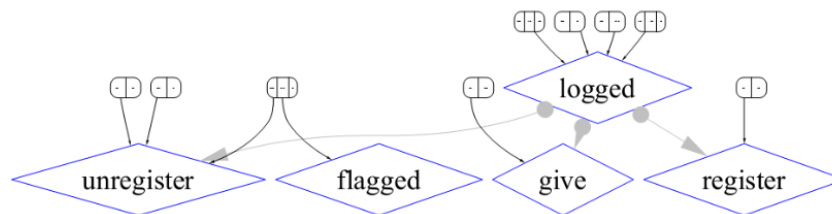
A abordagem *Theme* possui duas atividades: Identificação e Classificação de Interesses e Representação/Composição de Interesses.

Identificação e Classificação de Interesses: Essa atividade tem o apoio das *Action Views*, que são abstrações para os interesses de software. Para gerar uma *Action View* são necessárias duas entradas. A primeira é a lista de ações-chave, que consistem nos verbos e termos identificados pelo engenheiro de software que remetem a interesses. A

outra entrada necessária é o conjunto de requisitos do software. Na Figura 11 é apresentado um exemplo de *Action View* criada a partir de um conjunto de requisitos e lista de ações-chave de um sistema de gerenciamento de cursos apresentado por Baniassad (2004).

As ações-chave são representadas por losangos e os retângulos com bordas arredondadas representam os requisitos do software. Com esse artefato é possível visualizar a associação entre ações-chave e requisitos; e ainda separá-los em dois grupos: o grupo base que é autocontido, ou seja, não possui requisitos que entrecortam demais requisitos e o grupo transversal que possui requisitos que interferem em demais requisitos. O grupo transversal é colocado acima do grupo base e de suas ações parte uma seta, com um círculo no início, para as ações que sofrem influência da ação do grupo transversal. Na Figura 11 a ação *logged* entrecorta as ações *unregister*, *give* e *register*.

Figura 11 – Exemplo de Action View.



Fonte: Baniassad (2004).

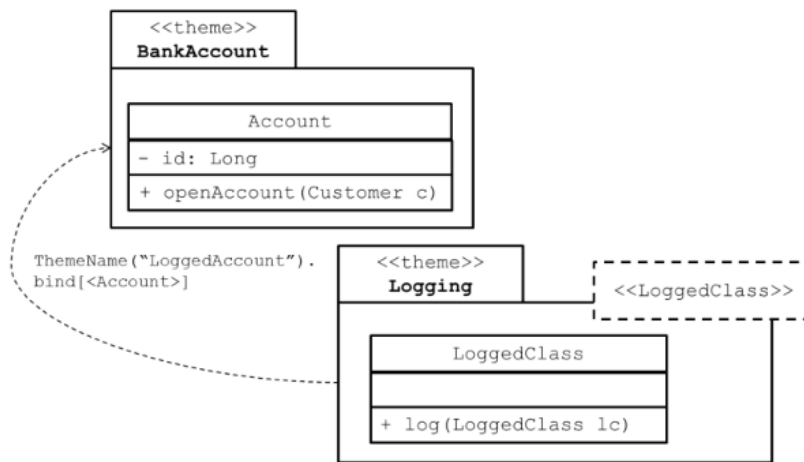
Representação/Composição de Interesses: Essa atividade é realizada no nível *Theme/UML* e tem por finalidade a representação dos interesses pelo estereótipo UML *theme*. Esse estereótipo define temas que podem ser temas base ou temas transversais. A composição está evidenciada na Figura 12 onde há um tema base (*BankAccount*) e um tema transversal (*Logging*) que estão relacionados por uma linha tracejada. Ou seja, nesse exemplo o tema base é afetado pelo tema transversal.

4.2 Análise das Abordagens

Após o estudo detalhado das 6 abordagens mapeadas por meio da RSL executada, duas foram escolhidas para serem alvo de um estudo empírico, mais especificamente um experimento controlado. Para essa escolha foram analisados os seguintes critérios:

- Critério 1: A abordagem deve ser mapeada por meio de uma RSL.
- Critério 2: A abordagem deve ter uma metodologia bem definida.
- Critério 3: A abordagem deve permitir ser executada de forma manual, ou seja, sem o apoio de software.

Figura 12 – Exemplo de Composição de Interesses.



Fonte: Baniassad (2004).

- Critério 4: A abordagem deve abranger interesses de software que contemplem requisitos funcionais e requisitos não funcionais.

A Tabela 11 mostra a relação das abordagens com os critérios definidos. As abordagens *ObasCId* e *Theme/Doc* foram as que atenderam a todos os critérios e por esse motivo são as selecionadas para serem alvo do estudo comparativo detalhado no Capítulo 5.

Tabela 11 – Seleção das abordagens mais relevantes.

Abordagem	Critério 1	Critério 2	Critério 3	Critério 4
AOFA	X	X	-	X
AspectQuery	X	X	-	X
Discern	X	X	-	X
ObasCId	X	X	X	X
Theme/Doc	X	X	X	X

Fonte: Elaborado pelo autor.

4.3 Lições do Capítulo

Neste capítulo são apresentadas as abordagens para identificação de Interesses Transversais mais relevantes no contexto deste trabalho. Os principais pontos abordados neste capítulo são:

- Detalhamento da abordagens: *AOFA*, *AspectQuery*, *DISCERN*, *EA-Miner*, *ObasCId* e *Theme/Doc*, apresentando as respectivas metodologias e exemplos de uso.

- Apresentação do processo de seleção das abordagens selecionadas para o estudo comparativo.

5 EXPERIMENTO CONTROLADO

Este capítulo tem por finalidade apresentar o experimento controlado executado com o objetivo de comparar a efetividade das abordagens para identificação de interesses de software, *ObasCId* e *Theme/Doc* as quais foram detalhadas respectivamente na Subseção 4.1.5 e Subseção 4.1.6. Na Seção 5.1 são apresentados os trabalhos relacionados ao experimento. Também foram coletadas métricas qualitativas relacionadas à utilidade percebida e facilidade de uso das abordagens, as quais são apresentadas em detalhes na Seção 5.6. Na Seção 5.2 é apresentado o escopo do experimento, seguido do planejamento na Seção 5.3. A Seção 5.4 mostra em detalhes a execução do experimento e a Seção 5.5 exhibe os resultados obtidos na realização do mesmo, bem como o teste de hipótese aplicado. Na Seção 5.7 são apresentadas as lições aprendidas neste capítulo.

5.1 Trabalhos Relacionados

Antes da execução do experimento controlado, foi realizada uma busca na literatura por trabalhos que propusessem experimentos relacionados à área de Interesses Transversais e abordagens para EROA. A busca foi realizada de maneira *ad hoc*.

No trabalho de Parreira Jr. (2018) são apresentados três estudos experimentais para avaliar as abordagens *ObasCId* e *Theme/Doc*. Apesar de um estudo proposto nesse trabalho comparar a efetividade das abordagens, ainda se faz necessário uma nova instância do experimento para confirmação dos resultados e correção de alguns pontos do protocolo seguido, como por exemplo, mitigar a ameaça do experimento ter sido conduzido por um dos criadores da abordagem *ObasCId*.

No trabalho de Sardinha et al. (2009) foi realizada uma avaliação empírica da acurácia da ferramenta desenvolvida para resolução de conflitos em requisitos na EROA. A metodologia do estudo não está bem detalhada no trabalho, mostrando apenas os resultados obtidos pela ferramenta para dois conjuntos de requisitos de software. O experimento proposto para este trabalho aborda conceitos diferentes, uma vez que avalia a efetividade das abordagens para identificação de interesses e não a acurácia para a resolução de conflitos.

Alguns estudos de caso também são propostos em outros trabalhos como o de Baniassad (2004) e o de Chengwan (2013). Nesses estudos de caso são avaliadas em um ambiente profissional, as abordagens propostas, sem uma intervenção direta do pesquisador, enquanto no experimento proposto neste trabalho há um controle maior justamente por se tratar de um experimento controlado.

5.2 Escopo

O experimento tem como objetivo comparar duas abordagens para identificação de interesses de software a partir de documento de requisitos. Os objetos de estudo são as

abordagens *ObasCId* e *Theme/Doc* e foi comparada a efetividade das mesmas. Seguindo o modelo proposto por Wohlin et al. (2012), que usa o modelo *Goal Question Metric*, proposto por Basili (1999), para estruturar o escopo, segue o resumo da definição:

Analisar o uso das abordagens *ObasCId* e *Theme/Doc*
com o propósito de *comparação*
com respeito à *efetividade*
da perspectiva de *pesquisador*
no contexto de *um grupo de alunos de graduação em Engenharia de Software*
para a *identificação de interesses de software em documentos de requisitos*.

5.3 Planejamento

O planejamento do experimento inclui alguns aspectos fundamentais, suas hipóteses, questões de pesquisa e variáveis. Também é descrito no planejamento a seleção dos participantes, o projeto do experimento e ameaças à validade do estudo (WOHLIN et al., 2012).

5.3.1 Seleção do Contexto

Com relação ao contexto do experimento, foi usada a abordagem *in-vitro*, uma vez que o experimento foi realizado em laboratório sob condições controladas. Os participantes foram alunos a partir do segundo ano de graduação na área de Engenharia de Software. O experimento comparou a efetividade das abordagens *ObasCId* e *Theme/Doc* para a identificação de Interesses Transversais em documentos de requisitos de software. O experimento trata de um problema real, a identificação de interesses de software em documentos de requisitos. O experimento foi aplicado em um contexto específico, alunos de engenharia de software e duas abordagens de identificação de interesses. Porém, as ideias gerais do experimento podem ser aplicadas em outro universo de participantes e outras abordagens que identifiquem interesses em documentos de requisitos.

Para a execução das abordagens foram utilizados como entrada os documentos de requisitos do Sistema de Gerenciamento de Cursos proposto no trabalho de Baniassad (2004) e da ferramenta *ObasCId Tool* proposta por Parreira Jr. (2018). Os softwares têm uma documentação que possibilita a identificação e também já possuem os interesses devidamente identificados e catalogados. Além disso, são equivalentes em nível de complexidade tendo em vista terem o mesmo número de interesses.

5.3.2 Formulação das Hipóteses

A base para a análise estatística de um experimento é o teste de hipóteses. A hipótese é estatisticamente formada, é coletada durante o curso do experimento e é usada

para, se possível, rejeitar a hipótese. Se a hipótese puder ser rejeitada, as conclusões podem ser extraídas (WOHLIN et al., 2012).

Para a formulação das hipóteses foi analisada a seguinte questão:

Qual das abordagens para identificação de Interesses de Software a partir de documento de requisitos é mais efetiva?

O experimento tem as seguintes hipóteses:

$H_0 = Efetividade_{OC} = Efetividade_{TD}$: Não há diferença entre a efetividade das abordagens *ObasCId* e *Theme/Doc*.

$H_1 : Efetividade_{OC} > Efetividade_{TD}$: A abordagem *ObasCId* possui maior efetividade do que a abordagem *Theme/Doc*.

$H_2 : Efetividade_{TD} > Efetividade_{OC}$: A abordagem *Theme/Doc* possui maior efetividade do que a abordagem *ObasCId*.

É possível avaliar os resultados por intermédio de índices de precisão e revocação. Assim, a *Precisão* é a fração dos documentos já examinados que são relevantes e *Revocação* é a fração dos documentos relevantes observada dentre os documentos examinados, (MONTEIRO, 2017).

Para o cálculo das métricas *Precisão* e *Revocação* algumas variáveis são utilizadas:

Verdadeiros Positivos (VP): Quantidade de interesses corretamente identificados com a utilização da abordagem.

Falsos Positivos (FP): Quantidade de interesses identificados incorretamente com a utilização da abordagem.

Falsos Negativos (FN): Quantidade de interesses não identificados com a utilização da abordagem.

A efetividade pode ser verificada pelo cálculo da *F-measure* ou *F-Score*. Esta medida calcula a média harmônica dos índices de precisão e revocação.

- *Precisão (P)*: $P = \frac{VP}{VP+FP}$
- *Revocação (R)*: $R = \frac{VP}{VP+FN}$
- *F-measure (F)*: $F = 2 * ((P * R)/(P + R))$

5.3.3 Seleção dos Sujeitos

Os participantes do experimento foram selecionados por amostragem não probabilística, onde há uma escolha deliberada, por conveniência. Participaram do experimento 36 alunos de graduação a partir do segundo ano do cursos de Engenharia de Software da Universidade Federal do Pampa - *Campus* Alegrete. Os participantes foram alunos matriculados nas disciplinas de Verificação e Validação de Software, Engenharia de Software Experimental e Resolução de Problemas VI.

5.3.4 Desenho do Experimento

Wohlin et al. (2012) define alguns princípios gerais para o experimento: tipo de projeto padrão, bloqueio, balanceamento e randomização.

Tipo de Projeto Padrão: O experimento realizado aplica o padrão de um *Fator* com *Dois Tratamentos*. O *Fator* é a abordagem para identificação de interesses de software e os *Tratamentos* são a *ObasCID* e *Theme/Doc*.

Bloqueio: Como os sujeitos podem possuir diferentes níveis de conhecimento nas áreas de Engenharia de Requisitos e Engenharia de Requisitos Orientada a Aspectos, foi aplicado um questionário de nivelamento para que fosse possível separá-los em grupos homogêneos de acordo com o conhecimento dos sujeitos nas áreas de interesse do experimento.

Balanceamento: Os participantes foram divididos em dois grupos homogêneos, ou seja, com participantes com níveis de conhecimento na área semelhantes.

Randomização: Os sujeitos foram alocados randomicamente para cada grupo e para cada abordagem. Além disso, como os sujeitos executaram todos os tratamentos, caracterizando um *design* pareado, a ordem de execução das abordagens para cada grupo também foi definida randomicamente.

5.3.5 Instrumentação

O nível de experiência dos participantes foi obtido por intermédio do questionário apresentado no Apêndice A, que teve como objetivo verificar o do nível de conhecimento na área que é objeto do estudo. Com essas informações foi possível identificar o perfil dos participantes.

Objetos: dentre os objetos utilizados no experimento estão a documentação de requisitos dos softwares *ObasCId Tool*, do Sistema de Gerenciamento de Cursos, presentes no Apêndice B, e os arquivos para treinamento dos participantes. Os treinamentos foram disponibilizados em vídeos em que foram apresentados os arquivos do Apêndice B, os quais são divididos em 3 partes. O primeiro treinamento aborda os conceitos principais sobre Engenharia de Requisitos e Engenharia de Requisitos Orientada a Aspectos, com a finalidade de nivelar o conhecimento dos participantes. O segundo e terceiro treinamentos explanam as abordagens *Theme/Doc* e *ObasCId*. Também foi disponibilizado um documento impresso contendo todo o material disponibilizado nos vídeos.

A documentação dos requisitos foi devidamente preparada para a identificação dos interesses de software de acordo com as abordagens utilizadas para essa tarefa, conforme Apêndice B. Como a participação no experimento foi de caráter voluntário, foi apresentado um termo de consentimento aos participantes, para registrar a concordância dos alunos em participar do experimento, presente no Apêndice C. Também foram elaborados os gabaritos dos requisitos, conforme Apêndice B, onde constam os interesses que

deveriam ser identificados pelos participantes.

Diretrizes: foram realizadas 3 sessões do experimento e cada sessão foi dividida em duas etapas, treinamento e execução. Durante o treinamento os participantes receberam instruções sobre Interesses de Software e Engenharia de Requisitos, em seguida, instruções sobre as abordagens a serem executadas. A ordem das instruções sobre as abordagens foi definida randomicamente e se manteve a mesma durante todas as sessões do experimento.

Medições: as métricas quantitativas elementos relevantes, elementos selecionados, positivos verdadeiros, falsos positivos, definidas na Subseção 5.3, bem como as métricas qualitativas referentes à percepção do usuário quanto à facilidade de uso e utilidade percebida (MARTINS, 2018), foram coletadas para cada sujeito. Todos os sujeitos realizaram as mesmas tarefas nas mesmas condições, exceto pelo fato das sessões terem sido realizadas em dias diferentes, porém sem diferenças significativas.

5.3.6 Ameaças à Validade

Para que um experimento obtenha um resultado válido, é necessário analisar algumas questões que podem invalidar ou ameaçar a validade do experimento. De acordo com Cook (1979), há quatro categorias em que as ameaças à validade podem se enquadrar: Validade de Conclusão, Validade Interna, Validade de Construção e Validade Externa.

5.3.6.1 Validade de Conclusão

Engloba as questões relacionadas à forma como foram analisados os resultados ou seja, se as conclusões a que se chegou estão corretas.

Baixo poder estatístico e pressupostos violados de testes estatísticos: Para mitigar estas ameaças foi aplicado o teste estatístico *t-test* pareado para comparação dos valores medidos. Como o *t-test* requer que os dados das amostras estejam distribuídos normalmente, foi aplicado o teste *Shapiro-Wilk* para verificar a normalidade dos dados.

Pesca e taxa de erro: Diz respeito à ameaça que tem origem na busca, por meio do pesquisador, de um resultado específico. O fato de não haver relação entre as abordagens analisadas e este trabalho, já auxilia na mitigação desta ameaça.

Confiabilidade das medições: Uma alta confiabilidade nas medições utilizadas tem impacto direto na validade do experimento. Sendo assim, para mitigar essa ameaça serão realizadas medições objetivas que não dependam de julgamento subjetivo, pois será medida a efetividade das abordagens. As métricas qualitativas coletadas para verificação da utilidade percebida e facilidade de uso apenas complementam as métricas quantitativas, não sendo o objetivo principal de medição.

Confiabilidade da implementação do tratamento: Há um risco de que a implementação do tratamento seja diferente para diferentes participantes. Para mitigar essa ameaça foram padronizados os procedimentos para que fossem iguais a todos os partici-

pantes. Todas as sessões do experimento seguiram a mesma ordem de execução, tendo os mesmos artefatos e tempo de duração.

Irrelevâncias aleatórias no ambiente experimental: O experimento foi realizado em um ambiente controlado, evitando interações externas ao experimento, como por exemplo interrupções, saída do ambiente do experimento, acesso a dispositivos eletrônicos, etc. Os participantes foram orientados quanto à proibição do uso de dispositivos eletrônicos e conversas paralelas durante o experimento. Participantes que não seguiram essas orientações tiveram seu resultado desconsiderado. Não foi permitida a saída de nenhum participante do ambiente do experimento durante a execução do mesmo.

Heterogeneidade aleatória dos sujeitos: A variação entre os níveis de conhecimento dos participantes foi levada em consideração como ameaça. Para mitigá-la os participantes foram classificados em grupos homogêneos conforme com o nível de conhecimento na área do experimento de acordo com o questionário presente no Apêndice A.

5.3.6.2 Validade Interna

São as influências que podem afetar as variáveis independentes.

História: Há o risco de algum período em específico ter influência na realização do experimento. Por exemplo, como se trata de um ambiente acadêmico, não é recomendado que seja realizado o experimento em épocas de provas ou outras atividades importantes. Para mitigar essa ameaça o experimento foi realizado em uma época em que os alunos não estavam muito sobrecarregados com trabalhos e provas do semestre.

Teste: No caso de o teste ser repetido, os participantes não deverão saber das respostas que foram dadas. Não houve necessidade de repetição do teste, as tarefas foram executadas uma vez por participante.

Instrumentação: Esse tipo de ameaça está relacionado com a qualidade dos artefatos utilizados no experimento. Para mitigar essa ameaça os artefatos produzidos deverão ser revisados por alguém alheio a elaboração do artefato e com conhecimento na área. Para isso, todos os artefatos foram validados por um especialista com doutorado na área de Engenharia de Requisitos Orientada a Aspectos.

Regressão estatística e seleção: Esta ameaça está relacionada à classificação dos participantes de acordo com experimentos ou estudos de caso anteriores. Para mitigar essa ameaça não foram selecionados participantes que já tivessem participado de experimentos ou estudos de caso semelhantes, inclusive a amostra utilizada no experimento piloto foi descartada para o experimento real.

Mortalidade: Está relacionada ao fato de alguns participantes abandonarem o experimento. Para mitigar essa ameaça foram selecionados indivíduos com conhecimento similar para que na falta de um, tenha outro participante com o mesmo nível de conhecimento para realizar o experimento. Porém, em nenhuma das sessões houve necessidade de substituição dos indivíduos.

5.3.6.3 Validade de Construção

Estão relacionadas ao projeto do experimento ou a fatores sociais.

Explicação pré-operacional inadequada da construção: Esta ameaça está relacionada com o fato do experimento não ter objetivos claros. Para mitigar essa ameaça foi comparada a efetividade de acordo com os conceitos de *Precisão* e *Revocação* definidos por Monteiro (2017).

Interação de diferentes tratamentos: Como o experimento segue um *design* pareado, todos os participantes realizaram os dois tratamentos. Entretanto, este fato não foi uma ameaça pois não houve aprendizado entre a execução das abordagens. Para verificação do aprendizado foi analisado os dados do experimento como um *design* randômico e os resultados mantiveram-se semelhantes. Com isso, ficou evidenciado que não houveram indícios de aprendizado entre a execução das abordagens.

Adivinhação de hipóteses: Se os participantes sabem sobre as hipóteses do experimento, isso pode ser uma ameaça. Para mitigar essa ameaça os participantes não foram informados sobre detalhes do experimento como hipóteses, por exemplo.

Aprensão à avaliação: O fato de estar sendo avaliado pode provocar um certo nervosismo nos participantes, ainda mais por se tratar de um ambiente acadêmico. Para mitigar essa ameaça os participantes foram informados que o experimento não terá impacto nas notas dos alunos e que eles não estavam sob avaliação e sim participando de um experimento.

Expectativas do experimentador: A expectativa do participante com relação ao experimento pode impactar no resultado, sendo conseqüentemente uma ameaça. Para mitigar essa ameaça os participantes não foram fornecidas informações detalhadas sobre o experimento, apenas foram dadas as instruções sobre as tarefas a serem realizadas.

5.3.6.4 Validade Externa

São ameaças que limitam a possibilidade de replicação do experimento.

Sujeitos: Os participantes selecionados para o experimento podem não representar um grupo significativo para a área de estudo. Para mitigar essa ameaça, o estudo foi realizado com estudantes de Engenharia de Software que representam uma amostra significativa para a área uma vez que o principal responsável pela identificação de interesses de software em um projeto é o engenheiro de software. Porém, o fato de realizar o experimento com estudantes é considerado uma ameaça, a qual não foi possível mitigar.

Interação de configuração e tratamento: É a ameaça relacionada ao fato de usar uma configuração ou material não representativo. Para mitigar essa ameaça, foi utilizada uma documentação de requisitos de software em modelo tradicional encontrado na indústria. Além disso, os artefatos foram validados com um especialista com doutorado na área.

5.4 Execução do Experimento

Algumas atividades antecederam a execução do experimento. Primeiramente foi feito contato com os professores das disciplinas escolhidas para a realização das sessões. Após o concorde dos mesmos, foi aplicado o questionário para verificação do nível de conhecimento dos possíveis participantes, conforme Apêndice A. A partir disso, os alunos foram classificados em dois níveis, básico e intermediário, tendo em vista não haver nenhum aluno com nível avançado em Engenharia de Requisitos e Engenharia de Requisitos Orientada a Aspectos. Após a elaboração de todos os artefatos a serem utilizados no experimento, os mesmos foram enviados para validação com um especialista, Doutor na área de Engenharia de Requisitos Orientada a Aspectos, onde foram seguidas as sugestões e realizadas as correções necessárias.

A execução do experimento foi dividida em 3 sessões que ocorreram em dias consecutivos, respectivamente 11, 12 e 13 de setembro de 2018. Para a execução do experimento foram convidados os alunos das disciplinas de Verificação e Validação de Software (sessão 1), Engenharia de Software Experimental (sessão 2) e Resolução de Problemas VI (sessão 3) ambas disciplinas do curso de Engenharia de Software da Universidade Federal do Pampa - Campus Alegrete. A Tabela 12 exibe as informações sobre as sessões do experimento. Na primeira coluna estão as sessões do experimento, seguidas pelas disciplinas onde foram executadas, número de sujeitos que participaram com os bloqueios por níveis básico e intermediário e data de execução.

Na sessão 1, dois participantes tiveram os resultados desconsiderados, pois não seguiram às recomendações quanto à não utilização de dispositivos eletrônicos e quanto a não se comunicar com os demais participantes durante a execução do experimento.

Tabela 12 – Sessões do experimento.

Sessão	Disciplina	Sujeitos	Nível Básico	Nível Intermediário	Data Execução
1	VeV	20	18	2	11/09/2018
2	ESE	12	4	8	12/09/2018
3	RPVI	4	2	2	13/09/2018
Total		36	24	12	

Fonte: Elaborado pelo autor.

A primeira atividade realizada foi a disponibilização do Termo de Consentimento para a execução do experimento, conforme Apêndice C. Foram selecionados para participar das demais atividades os alunos que assinaram o referido termo. Em seguida, os participantes foram informados do caráter "não avaliativo" do experimento, ou seja, que os participantes não estavam sob avaliação. Após isso, os presentes foram divididos em dois grupos homogêneos, A e B, de acordo com o resultado do questionário presente no Apêndice A, aplicado anteriormente.

Após a divisão dos participantes nos grupos, teve início o treinamento. Na primeira etapa foi exibido o vídeo do treinamento sobre Separação de Interesses de Software. Essa atividade teve uma duração aproximada de 12 minutos. Em seguida, foi exibido, após escolha randômica, o treinamento sobre a abordagem *Theme/Doc*. Essa atividade teve a duração aproximada de 9 minutos. A última atividade do treinamento foi a apresentação do vídeo sobre a abordagem *ObasCId*, que teve duração aproximada de 12 minutos. É importante ressaltar que a diferença de tempo de duração entre os treinamentos das abordagens se deve às diferenças existentes entre elas, pois a abordagem *ObasCId* exige que mais atividades sejam realizadas para a identificação dos interesses de software e isso impactou no aumento da duração do treinamento da mesma.

Ao fim da etapa de treinamento teve início a etapa de execução. Para isso, foi entregue aos sujeitos os artefatos necessários para a realização das tarefas. A atividade principal a ser realizada era identificar interesses de software em documentos de requisitos, utilizando as abordagens que foram apresentadas. Na primeira parte da execução o GRUPO A deveria usar a abordagem *ObasCId* e o GRUPO B a abordagem *Theme/Doc*, na segunda parte da execução o GRUPO A usou a abordagem *Theme/Doc* e o GRUPO B a *ObasCId*. Para isso, foram entregues aos participantes o material presente nas apresentações dos treinamentos em um documento impresso. Além disso, o GRUPO A recebeu os requisitos do Sistema de Gerenciamento de Cursos, conforme Apêndice B e o Catálogo de Interesses, conforme Apêndice A. Já para o GRUPO B foram entregues os requisitos da ferramenta *OC Tool*, também conforme o Apêndice B. Na segunda parte do experimento o GRUPO A recebeu os requisitos da ferramenta *OC Tool* e o GRUPO B os requisitos do Sistema de Gerenciamento de Cursos mais o Catálogo de Interesses de Software. Após a realização da atividade de identificação de interesses os participantes responderam o questionário de percepção de utilidade e facilidade de uso (MARTINS, 2018). Cabe ressaltar que as três sessões foram realizadas da mesma maneira, seguindo as mesmas atividades na mesma ordem de execução.

5.5 Resultados do Experimento

Após os participantes do experimento realizarem as atividades de identificação de interesses utilizando as abordagens em análise, passou-se a etapa de correção, onde foram confrontados os interesses identificados pelos sujeitos com os interesses presentes no gabarito. A correção foi realizada em pares e se buscou chegar em um consenso.

A Figura 13 exhibe os resultados do experimento. As caixas em amarelo referem-se à abordagem *ObasCId* e as caixas na cor verde mostram os resultados da abordagem *Theme/Doc*. A figura mostra também os resultados de acordo com os níveis, básico e intermediário.

Analisando os resultados é possível notar que em geral a abordagem *Theme/Doc* obteve melhores índices para a *F-measure*, independente do bloqueio realizado. Porém,

a diferença entre os tratamentos é menor entre os sujeitos classificados no nível intermediário. A abordagem *ObasCId* apresenta uma variabilidade menor comparada com a *Theme/Doc*. O limite inferior para o nível básico é bastante semelhante, 0.20 para a *ObasCId* e 0.22 para a *Theme/Doc*, já no nível intermediário a diferença é maior, 0.33 para a *ObasCId* e 0.72 para a *Theme/Doc*.

O grupo que apresenta uma maior variabilidade entre o primeiro e terceiro quartil é o de nível básico com a abordagem *Theme/Doc*. Já o grupo com menor dispersão é o de nível intermediário que executou a abordagem *Theme/Doc*, com uma diferença de 0.22 entre o primeiro e terceiro quartil. Além disso, este grupo também apresentou o valor mais alto para o limite inferior, 0.72. Outro dado interessante é que em todos os níveis e tratamentos o limite superior foi o valor máximo possível, ou seja, em todos os grupos houveram sujeitos que conseguiram identificar todos os interesses presentes nos softwares sob análise.

Com base nos dados apresentados, é possível verificar que ambos os tratamentos obtiveram melhores resultados com grupos com maior experiência em Engenharia de Requisitos e Engenharia de Requisitos Orientada a Aspectos e que os tratamentos apresentam diferenças significativas com relação à sua efetividade.

Tabela 13 – Resultados da identificação de interesses.

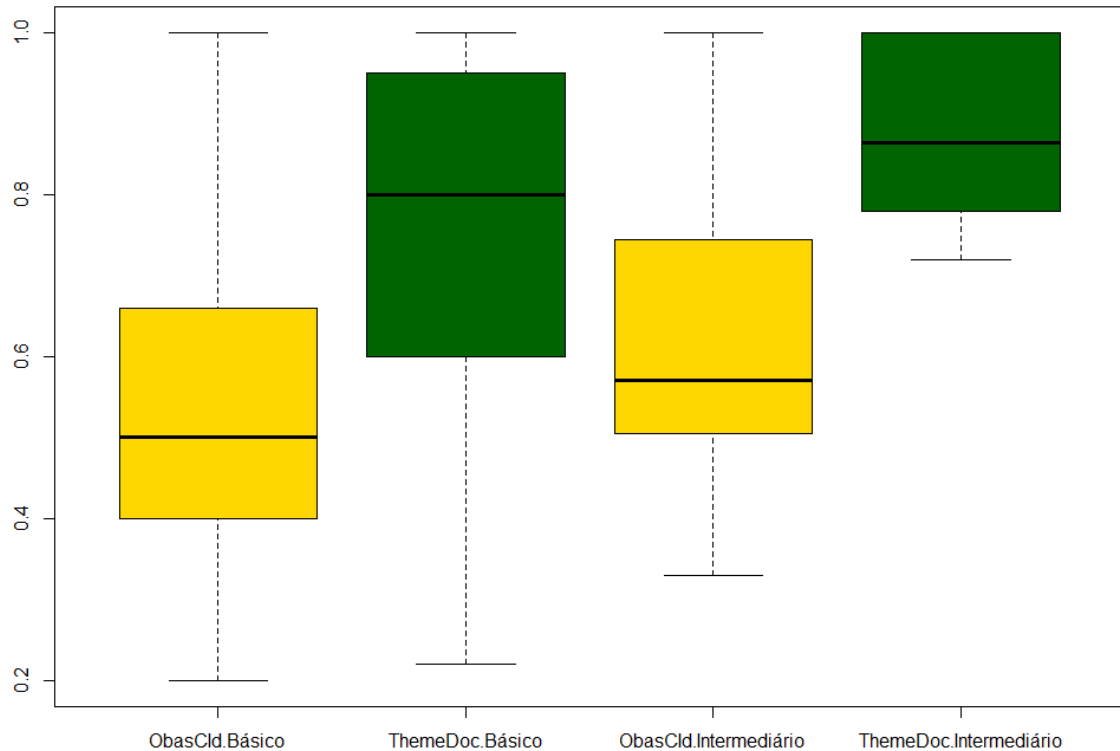
	OC.Básico	TD.Básico	OC.Intermediário	TD.Intermediário
Limite Inferior	0.20	0.22	0.33	0.72
Primeiro Quartil	0.40	0.60	0.505	0.78
Mediana	0.50	0.80	0.570	0.865
Terceiro Quartil	0.66	0.95	0.745	1
Limite Superior	1	1	1	1

Fonte: Elaborado pelo autor.

5.5.1 Teste de Hipótese

Segundo Graybill, Iyer e Burdick (1998) para ser possível aceitar ou rejeitar uma determinada hipótese, é necessário realizar um procedimento denominado Teste de Hipótese ou Teste de Significância. O primeiro procedimento realizado foi a análise das amostras do experimento para verificação da normalidade entre a distribuição dos dados. Para esta tarefa foi usado o teste *Shapiro-Wilk*, em que se busca rejeitar a hipótese nula para inferir a normalidade dos dados (WOHLIN et al., 2012). A Tabela 14 exhibe os valores de W e p -value obtidos com o teste *Shapiro-Wilk* para o experimento. Como p -value $>$ 0.05 e $W_{(\text{calculado})} < W_{(0,05;10)}$, é possível afirmar com nível de significância de 5% que a amostra provém de uma população normal, ou seja, há 95% de chances da amostra ser normal.

Figura 13 – Resultados do experimento.



Fonte: Elaborado pelo autor.

Com base nos resultados do teste de normalidade aplicado, optou-se por realizar o *T-test* Pareado para testar a H_0 do experimento. Aplicando este teste foi obtido um $p\text{-value} = 2.001e-05$ ($p\text{-value} < 0.05$), ou seja, é possível rejeitar a H_0 e afirmar que há diferenças entre a efetividade das abordagens *ObasCId* e *Theme/Doc*. De acordo com os dados exibidos na Figura 13, podemos aceitar a H_2 : A abordagem *Theme/Doc* possui maior efetividade do que a abordagem *ObasCId*.

É válido ressaltar que os resultados apresentados foram obtidos para uma determinada amostra de uma população específica sob um contexto específico, não podendo ser generalizados.

Tabela 14 – Resultados teste *Shapiro-Wilk*.

	Sessão 1	Sessão 2	Sessão 3	Geral
W	0.97115	0.9616	0.89359	0.92798
p-value	0.7789	0.8064	0.3999	0.2013

Fonte: Elaborado pelo autor.

5.6 Utilidade Percebida e Facilidade de Uso

Foram coletadas métricas qualitativas referentes à utilidade percebida e facilidade de uso das abordagens, por meio de um questionário disponível em Martins (2018). O modelo de referência utilizado para a formulação das questões presentes no questionário foi o TAM (*Technology Acceptance Model*), proposto por Davis (1993).

O questionário foi organizado com 12 questões, 6 para cada uma das abordagens sob análise. As 6 primeiras referindo-se à abordagem *ObasCId* e as demais à abordagem *Theme/Doc*. Na Figura 14 são apresentados os resultados obtidos com a aplicação do questionário.

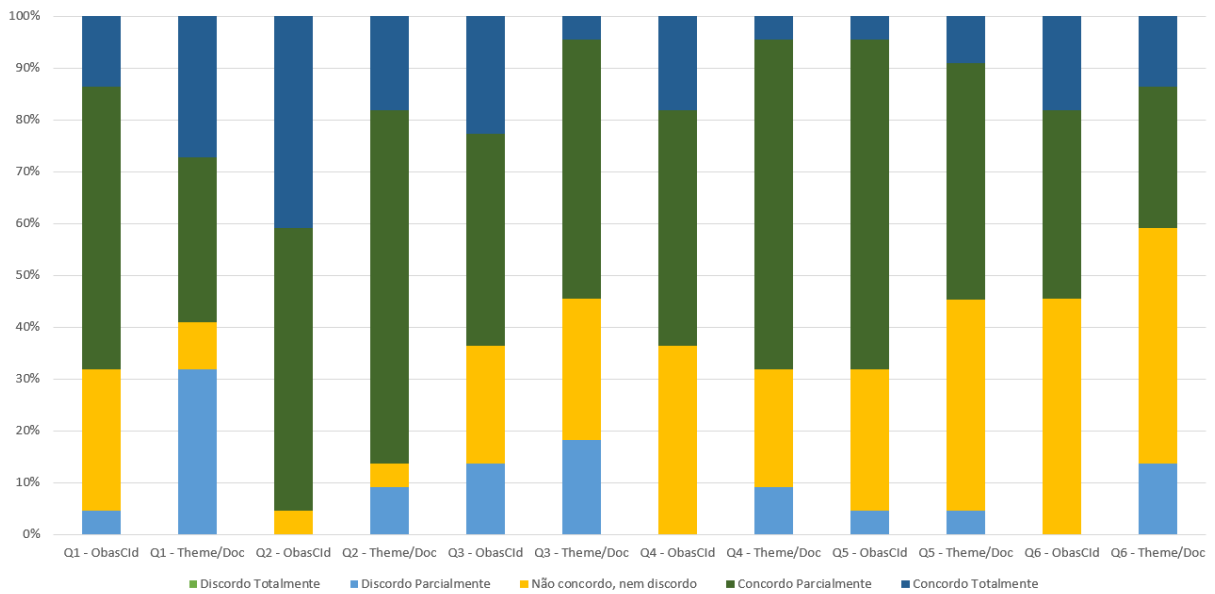
Para responder o questionário o participante deveria escolher uma dentre 5 respostas possíveis: Discordo Totalmente, Discordo Parcialmente, Não Concordo, nem Discordo, Concordo Parcialmente e Concordo Totalmente. Um total de 22 participantes do experimento responderam o questionário. Em geral, a abordagem *ObasCId* apresentou melhores resultados pois apresenta uma porcentagem maior de respostas que obtiveram concordância total, ou seja, respostas assinaladas com "Concordo Totalmente". Ambas abordagens tiveram uma boa aceitação, o que se evidenciou por meio das questões 6 e 12, as quais remetem à recomendação de uso das abordagens. A abordagem *Theme/Doc* teve melhores resultados nas perguntas relacionadas à facilidade de uso, questões 1 e 7, obtendo 27,3% de respostas *Concordo Totalmente* contra 13,6% da *ObasCId*. Nas questões 5 e 11, que tratam sobre os resultados esperados, apesar da abordagem *Theme/Doc* obter um resultado maior de respostas "Concordo Totalmente" (9,1%), teve um número expressivo de sujeitos que mantiveram-se neutros, respondendo "Não Concordo, nem Discordo" e aliado ao fato de 63,6% dos participantes responderem "Concordo Parcialmente" para a mesma questão, porém relacionada a abordagem *ObasCId*, pode representar que esta última tenha melhores índices quanto aos resultados esperados.

5.7 Lições do Capítulo

Neste capítulo é apresentado o experimento controlado conduzido com a finalidade de comparar a efetividade das abordagens *ObasCId* e *Theme/Doc* juntamente da avaliação de utilidade percebida e facilidade de uso das abordagens analisadas. Os principais pontos abordados neste capítulo são:

- Definição do escopo do experimento.
- Detalhamento do planejamento e execução do experimento.
- Discussão dos resultados do experimento.
- Definições das métricas qualitativas relacionadas à utilidade percebida e facilidade de uso.

Figura 14 – Resultados do questionário.



Fonte: Elaborado pelo autor.

- Discussão dos resultados obtidos para a utilidade percebida e facilidade de uso das abordagens analisadas.

6 CONSIDERAÇÕES FINAIS

O objetivo geral deste trabalho é a realização de estudos empíricos sobre abordagens para identificação de interesses de software. Esse estudo compreende uma RSL, onde analisou-se o estado da arte da área e as abordagens mais relevantes foram selecionadas, e um Experimento Controlado para medição e comparação da efetividade das abordagens para identificação de Interesses Transversais.

Através dos estudos realizados, bem como por meio dos resultados obtidos com a RSL e experimento controlado, é possível verificar que o objetivo principal deste trabalho foi atingido. Além disso, como principais contribuições deste trabalho destacam-se a obtenção de um catálogo das principais e mais relevantes abordagens para identificação de Interesses Transversais; o detalhamento dessas abordagens incluindo os principais conceitos e exemplos de uso; e a obtenção de evidências empíricas que indicam que para um determinado cenário e contexto, há diferenças significativas entre a efetividade das abordagens *ObasCId* e *Theme/Doc*. Os resultados obtidos por intermédio da coleta de métricas qualitativas relacionadas à utilidade percebida e facilidade de uso das abordagens também podem ser elencados como contribuição deste trabalho, pois não foram encontrados na literatura trabalhos que realizem essas medições para o contexto e cenário apresentados.

Alguns desafios se apresentaram para a realização deste trabalho. A definição do protocolo e execução de uma RSL bem como de um experimento não são tarefas triviais, pois inúmeros aspectos devem ser mapeados e controlados para que esses estudos sejam realizados satisfatoriamente. Por meio de protocolos bem definidos e seguindo metodologias consolidadas para a condução desses estudos empíricos, foi possível atingir ao que foi proposto, superando os desafios que se apresentaram no decorrer deste trabalho.

6.1 Trabalhos Futuros

Como sugestão de trabalho futuro é possível ampliar os resultados do experimento realizado com uma amostra diferente, por exemplo, replicando-o no meio profissional, tendo em vista o experimento ter sido aplicado no meio acadêmico. Também podem ser analisadas as abordagens selecionadas na RSL para que sejam propostos mecanismos para automatização e melhoria de alguns processos das mesmas. Outra sugestão de trabalho futuro é a replicação da RSL realizada, que pode ser instanciada novamente em momento oportuno, visando a atualização dos resultados. Também é possível realizar estudos empíricos na área de Interesses Transversais em outras etapas do ciclo de vida do software, como por exemplo, na implementação.

REFERÊNCIAS

- ACM. **ACM Digital Library**. 2017. Disponível em: <<https://dl.acm.org/>>. Acesso em: 8.8.2017. Citado na página 34.
- ALI, B. S.; KASIRUN, Z. M. Developing tool for crosscutting concern identification using nlp. In: IEEE. **Information Technology, 2008. ITSIM 2008. International Symposium on**. 2008. v. 3, p. 1–8. Disponível em: <<https://ieeexplore.ieee.org/document/4632039>>. Citado na página 38.
- ALI, B. S.; KASIRUN, Z. M. A review on approaches for identifying crosscutting concerns. In: IEEE. **Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on**. 2008. p. 855–859. Disponível em: <<https://ieeexplore.ieee.org/document/4737078>>. Citado na página 33.
- ALI, M. S. et al. A systematic review of comparative evidence of aspect-oriented programming. **Information and software Technology**, Elsevier, v. 52, n. 9, p. 871–887, 2010. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0950584910000819>>. Citado na página 33.
- ALSHAREEF, S. F.; MAATUK, A. M.; ABDELAZIZ, T. M. Aspect-oriented requirements engineering: approaches and techniques. In: ACM. **Proceedings of the First International Conference on Data Science, E-learning and Information Systems**. 2018. p. 13. Disponível em: <https://www.researchgate.net/publication/327732449_Aspect-Oriented_Requirements_Engineering_Approaches_and_Techniques>. Citado na página 33.
- AMEL, B.; AMIN, B. A multi-level interaction dealing approach with aspects in requirement engineering phase. In: **2016 International Conference on Engineering MIS (ICEMIS)**. [s.n.], 2016. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/document/7745365>>. Citado na página 38.
- ANTONELLI, L.; ROSSI, G.; LEITE, J. C. S. do P. Early identification of crosscutting concerns in the domain model guided by states. In: **Proceedings of the 2010 ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2010. (SAC '10), p. 275–280. ISBN 978-1-60558-639-7. Disponível em: <<http://doi.acm.org/10.1145/1774088.1774147>>. Citado na página 38.
- BANIASSAD, E.; CLARKE, S. Theme: An approach for aspect-oriented analysis and design. In: IEEE COMPUTER SOCIETY. **Proceedings of the 26th International Conference on Software Engineering**. 2004. p. 158–167. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.5495&rep=rep1&type=pdf>>. Citado 3 vezes nas páginas 38, 39 e 40.
- BANIASSAD, S. C. E. **Theme: An Approach for Aspect-Oriented Analysis and Design**. Proceedings of the 26th International Conference on Software Engineering (ICSE'0, 2004. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.5495&rep=rep1&type=pdf>>. Citado 10 vezes nas páginas 24, 27, 49, 50, 51, 52, 55, 56, 90 e 91.
- BASIL, V. R. **The Goal Question Metric Approach**. 1999. Disponível em: <https://www.researchgate.net/publication/239596452_The_goal_question_metric_approach>. Citado na página 56.

- CHEN, H.; WANG, J.; DONG, W. High confidence software engineering technologies. **Acta Electronica Sinica**, v. 31, n. 12A, p. 1933–1938, 2003. Disponível em: <https://www.researchgate.net/publication/228367578_High_confidence_software_for_cyber-physical_systems>. Citado na página 28.
- CHENGWAN, H. **AspectQuery: A Method for Identification of Crosscutting Concerns in the Requirement Phase**. 2013. Disponível em: <https://www.jstage.jst.go.jp/article/transinf/E96.D/4/E96.D_897/_article/-char/en>. Citado 5 vezes nas páginas 24, 42, 43, 44 e 55.
- CHITCHYAN, R.; SAWYER. **aspect oriented requirements engineering, architectures and design**. Lancaster University, 2005. Disponível em: <https://www.researchgate.net/publication/228541921_Aspect-Oriented_Requirements_Engineering>. Citado na página 29.
- COOK, D. T. C. T. D. **Quasi-experimentation: Design and Analysis Issues for Field Settings**. Houghton Mifflin, 1979. Disponível em: <<http://www.worldcat.org/title/quasi-experimentation-design-analysis-issues-for-field-settings/oclc/7593552>>. Citado na página 59.
- DAVIS, F. D. User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. **International Journal of Man-Machine Studies**, v. 38, n. 3, p. 475 – 487, 1993. ISSN 0020-7373. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0020737383710229>>. Citado na página 66.
- DIJKSTRA, E. W. **A Discipline of Programming**. Prentice Hall, 1976. Disponível em: <<https://dl.acm.org/citation.cfm?id=550359>>. Citado 2 vezes nas páginas 23 e 27.
- DURELLI, R. S. et al. A systematic review on mining techniques for crosscutting concerns. In: ACM. **Proceedings of the 28th Annual ACM Symposium on Applied Computing**. 2013. p. 1080–1087. Disponível em: <<http://www2.dc.ufscar.br/~lawasp/2012/artigos/09.pdf>>. Citado na página 33.
- D'AMORIM, F.; BORBA, P. Modularity analysis of use case implementations. **Journal of Systems and Software**, v. 85, n. 4, p. 1012 – 1027, 2012. ISSN 0164-1212. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0164121211002950>>. Citado na página 38.
- ELSEVIER. **Science Direct**. 2017. Disponível em: <<https://www.sciencedirect.com>>. Acesso em: 8.8.2017. Citado na página 35.
- GLASS, R. L. **The software research crisis**. IEEE Softw. 11, 1994. Disponível em: <<https://ieeexplore.ieee.org/document/329400>>. Citado na página 30.
- GRAYBILL, F.; IYER, H.; BURDICK, R. **Applied Statistics: A First Course in Inference**. Prentice Hall, 1998. (Data Warehousing Institute Series from). ISBN 9780136214670. Disponível em: <<https://books.google.com.br/books?id=Hqqfn3u8-T8C>>. Citado na página 64.
- HE, C.; TU, C. Aspectquery: A method for identification of crosscutting concerns in the requirement phase. **IEICE TRANSACTIONS on Information and Systems**, The Institute of Electronics, Information and Communication Engineers, v. 96, n. 4, p.

897–905, 2013. Disponível em: <https://www.researchgate.net/publication/269862640_AspectQuery_A_Method_for_Identification_of_Crosscutting_Concerns_in_the_Requirement_Phase>. Citado 3 vezes nas páginas 38, 39 e 40.

HENDERSON-SELLERS, B. et al. A method engineering approach to developing aspect-oriented modelling processes based on the open process framework. **Information and Software Technology**, v. 49, n. 7, p. 761 – 773, 2007. ISSN 0950-5849. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584906001121>>. Citado na página 38.

HENRY, D. K. S. **Software Structure Metrics Based on Information Flow**. IEEE Transactions on, vol. SE-7, 1981. Disponível em: <<https://ieeexplore.ieee.org/document/1702877>>. Citado na página 28.

IEEE. **Standard Glossary of Software Engineering Terminology**. 1990. Disponível em: <<https://ieeexplore.ieee.org/document/159342>>. Citado na página 23.

IEEE. **IEEE Explore Digital Library**. 2017. Disponível em: <<https://ieeexplore.ieee.org>>. Acesso em: 8.8.2017. Citado na página 34.

IRWIN, G. K. J. L. A. M. C. M. C. L. J. L. L. **Aspect-oriented programming**. Springer Berlin Heidelberg, 1997. Disponível em: <<https://www.cs.ubc.ca/~gregor/papers/kiczales-ECOOP1997-AOP.pdf>>. Citado 2 vezes nas páginas 23 e 28.

JÚNIOR, P. A. P.; PENTEADO, R. A. D. Obasid (-tool): an ontologically based approach for concern identification and classification and its computational support. **Journal of the Brazilian Computer Society**, Springer, v. 24, n. 1, p. 3, 2018. Disponível em: <https://www.researchgate.net/publication/322410610_ObasCIId-Tool_an_ontologically_based_approach_for_concern_identification_and_classification_and_its_computational_support>. Citado 3 vezes nas páginas 38, 39 e 40.

KITCHENHAM, B. A. **Kitchenham, B.: Guidelines for performing Systematic Literature Reviews in software engineering**. EBSE Technical Report EBSE-2007-01. [s.n.], 2007. Disponível em: <<https://userpages.uni-koblenz.de/~laemmel/ese/course/slides/slr.pdf>>. Citado 3 vezes nas páginas 31, 34 e 36.

KONG, L.; YUAN, T. Use case modeling approach for early aspect acquisition. **SIGSOFT Softw. Eng. Notes**, ACM, New York, NY, USA, v. 34, n. 4, p. 1–6, jul. 2009. ISSN 0163-5948. Disponível em: <<http://doi.acm.org/10.1145/1543405.1543417>>. Citado na página 38.

MARÇAL, I. et al. Techniques for the identification of crosscutting concerns: A systematic literature review. In: **Information Technology: New Generations**. Springer, 2016. p. 569–579. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-32467-8_50>. Citado na página 33.

MAREW, T.; LEE, J.-S.; BAE, D.-H. Tactics based approach for integrating non-functional requirements in object-oriented analysis and design. **Journal of Systems and Software**, v. 82, n. 10, p. 1642 – 1656, 2009. ISSN 0164-1212. SI: YAU. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0164121209000533>>. Citado na página 38.

- MARIN, M.; MOONEN, L.; DEURSEN, A. van. An integrated crosscutting concern migration strategy and its application to jhotdraw. In: IEEE. **Source Code Analysis and Manipulation, 2007. SCAM 2007. Seventh IEEE International Working Conference on**. 2007. p. 101–110. Disponível em: <<https://ieeexplore.ieee.org/document/4362902>>. Citado na página 38.
- MARQUES, A. G.; MOREIRA, A.; ARAÚJO, J. a. Multi-dimensional composition by objective in aspect-oriented requirements analysis. In: **Proceedings of the 13th International Workshop on Early Aspects**. New York, NY, USA: ACM, 2008. (EA '08), p. 19–26. ISBN 978-1-60558-032-6. Disponível em: <<http://doi.acm.org/10.1145/1370828.1370834>>. Citado na página 38.
- MARTINS, G. L. **Questionário de utilidade percebida e facilidade de uso**. 2018. Disponível em: <<https://goo.gl/forms/Ps5FHVwrhcuo9FFG3>>. Citado 3 vezes nas páginas 59, 63 e 66.
- MEYER, B. **Object-oriented Software Construction**. Prentice Hall, 1997. Disponível em: <<https://sophia.javeriana.edu.co/~cbustaca/docencia/POO-2016-01/documentos/Object20Oriented20Software20Construction-Meyer.pdf>>. Citado na página 23.
- MONTEIRO, S. D. **Sistemas de recuperação da informação e o conceito de relevância nos mecanismos de busca: semântica e significação**. Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação, v. 22, n.50, p. 161-175, 2017. Disponível em: <<https://periodicos.ufsc.br/index.php/eb/article/view/1518-2924.2017v22n50p161>>. Citado 2 vezes nas páginas 57 e 61.
- NAKAGAWA, E. Y. et al. An aspect-oriented reference architecture for software engineering environments. **Journal of Systems and Software**, v. 84, n. 10, p. 1670 – 1684, 2011. ISSN 0164-1212. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0164121211001038>>. Citado na página 38.
- PARNAS, D. L. **On the criteria to be used in decomposing systems into modules**. Communications of ACM, 1972. Disponível em: <https://www.win.tue.nl/~wstomv/edu/2ip30/references/criteria_for_modularization.pdf>. Citado na página 23.
- PARREIRA Jr., R. A. D. P. Obascid(-tool): an ontologically based approach for concern identification and classification and its computational support. **Journal of the Brazilian Computer Society**, v. 24, n. 1, p. 3, Jan 2018. ISSN 1678-4804. Disponível em: <<https://doi.org/10.1186/s13173-017-0067-6>>. Citado 14 vezes nas páginas 24, 29, 30, 45, 46, 47, 48, 49, 50, 55, 56, 87, 89 e 91.
- PMBOK, G. Um guia do conhecimento em gerenciamento de projetos. **Quarta Edição**, v. 123, 2013. Disponível em: <<https://brasil.pmi.org/brazil/PMBOKGuideAndStandards.aspx>>. Citado na página 25.
- PRESSMAN, R. **Software Engineering: A Practitioner's Approach**. 7. ed. New York, NY, USA: McGraw-Hill, Inc., 2010. ISBN 0073375977, 9780073375977. Disponível em: <http://dinus.ac.id/repository/docs/ajar/RPL-7th_ed_software_engineering_a_practitioners_approach_by_roger_s._pressman_.pdf>. Citado 2 vezes nas páginas 23 e 29.

RAZZAQ, A.; ABBASI, R. Automated separation of crosscutting concerns: Earlier automated identification and modularization of cross-cutting features at analysis phase. In: **2012 15th International Multitopic Conference (INMIC)**. [s.n.], 2012. p. 471–478. Disponível em: <<https://ieeexplore.ieee.org/document/6511500>>. Citado 3 vezes nas páginas 38, 39 e 40.

RAZZAQ, R. A. A. **Automated Separation of Crosscutting Concerns: Earlier**. 2012. Disponível em: <<https://ieeexplore.ieee.org/document/6511500>>. Citado 2 vezes nas páginas 24 e 41.

RESENDE, A. M. P. de. **Um método para Identificação e Definição de Aspectos Iniciais**. Tese (Doutorado) — Instituto Tecnológico de Aeronáutica, São José dos Campos, Brasil, 2007. Citado na página 23.

ROSENHAINER, L. **The DISCERN Method: Dealing Separately with Crosscutting Concerns**. 2005. Disponível em: <https://www.researchgate.net/publication/228947290_The_discern_method_Dealing_separately_with_crosscutting_concerns>. Citado 3 vezes nas páginas 24, 27 e 43.

ROSENHAINER, L. The discern method: dealing separately with crosscutting concerns. In: CITESEER. **Early Aspects Workshop at OOPSLA-2005**. 2005. Disponível em: <https://www.researchgate.net/publication/228947290_The_discern_method_Dealing_separately_with_crosscutting_concerns>. Citado 3 vezes nas páginas 38, 39 e 40.

SAMPAIO, A. et al. Ea-miner: a tool for automating aspect-oriented requirements identification. In: ACM. **Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering**. 2005. p. 352–355. Disponível em: <https://www.researchgate.net/publication/220883449_EA-Miner_a_tool_for_automating_aspect-oriented_requirements_identification>. Citado 3 vezes nas páginas 38, 39 e 40.

SAMPAIO, C.; RAYSON. **EA-Miner: A tool for automating aspect-oriented requirements identification**. 2005. Disponível em: <https://www.researchgate.net/publication/220883449_EA-Miner_a_tool_for_automating_aspect-oriented_requirements_identification>. Citado 2 vezes nas páginas 24 e 44.

SANTANNA, C. **On the Modularity of Aspect-Oriented Design: a concern-driven measurement approach**. Tese (Doutorado) — Centre for Telematics and Information Technology, University of Twente, Enschede, The Netherlands, 2008. Disponível em: <<http://www.loa.istc.cnr.it/Guizzardi/SELMAS-CR.pdf>>. Citado na página 23.

SARDINHA, A. et al. Ea-analyzer: automating conflict detection in aspect-oriented requirements. In: IEEE COMPUTER SOCIETY. **Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering**. 2009. p. 530–534. Disponível em: <<https://ieeexplore.ieee.org/document/5431741>>. Citado na página 55.

SARDINHA, A. et al. Ea-tracer: Identifying traceability links between code aspects and early aspects. In: **Proceedings of the 27th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2012. (SAC '12), p. 1035–1042. ISBN

- 978-1-4503-0857-1. Disponível em: <<http://doi.acm.org/10.1145/2245276.2231938>>. Citado na página 38.
- SILVA, R. F. da; MAIA, M. de A.; SOARES, M. S. A systematic review on performance evaluation of aspect-oriented programming techniques used to implement crosscutting concerns. In: **ICEIS (2)**. [s.n.], 2014. p. 5–13. Disponível em: <<http://lascam.facom.ufu.br/cms/userfiles/downloads/2014/ICEISRodrigo2014.pdf>>. Citado na página 33.
- SOMMERVILLE, I. **Engenharia de Software**. [S.l.]: Pearson Prentice Hall, 2011. Citado 3 vezes nas páginas 23, 27 e 28.
- SPRINGER. **Springer Link**. 2017. Disponível em: <<https://link.springer.com/>>. Acesso em: 8.8.2017. Citado na página 35.
- SUTTON Jr., I. R. S. M. **Modeling of Software Concerns in Cosmos**. 1st International Conference on Aspect Oriented Software Development - ACM Press, 2002. Disponível em: <<https://dl.acm.org/citation.cfm?id=508402>>. Citado 2 vezes nas páginas 23 e 27.
- SÁNCHEZ, P. et al. Model-driven development for early aspects. **Information and Software Technology**, v. 52, n. 3, p. 249 – 273, 2010. ISSN 0950-5849. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584909001669>>. Citado na página 38.
- VELAN, S. S.; PRIYA, R. V. S. Identification and removal of semantic interference during the analysis phase of aspect oriented software development. In: **2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)**. [s.n.], 2016. p. 709–714. Disponível em: <<https://ieeexplore.ieee.org/document/7831732>>. Citado na página 38.
- WOHLIN, C. et al. **Experimentation in Software Engineering**. Norwell, MA, USA: Springer, 2012. Disponível em: <<https://www.springer.com/la/book/9783642290435>>. Citado 8 vezes nas páginas 29, 30, 31, 36, 56, 57, 58 e 64.
- YIN, R. K. **Case study research: Design and methods**. Sage publications, 2013. Disponível em: <<https://us.sagepub.com/en-us/nam/author/robert-k-yin>>. Citado 2 vezes nas páginas 30 e 32.

Apêndices

APÊNDICE A – QUESTIONÁRIO PARA DEFINIÇÃO DOS GRUPOS

Este questionário tem como objetivo verificar seu nível de conhecimento em Engenharia de Requisitos e Interesses de Software. Este instrumento não é uma ferramenta avaliativa e os resultados não serão divulgados.

Responda às seguintes perguntas:

1. Como você classifica seu conhecimento em Engenharia de Requisitos?
 - a) () Nível 1: nunca ouvi falar em Engenharia de Requisitos.
 - b) () Nível 2: meu conhecimento sobre Engenharia de Requisitos foi adquirido em disciplinas de graduação e pós-graduação.
 - c) () Nível 3: além de aprendido no meio acadêmico, já trabalhei com Engenharia de Requisitos de forma ativa na área acadêmica ou no mercado de trabalho.

2. Qual o seu tempo de experiência em Engenharia de Requisitos?
 - a) () menos de 6 meses.
 - b) () entre 6 meses e 1 ano.
 - c) () entre 1 e 3 anos.
 - d) () mais de 3 anos.

3. Você conhece alguma abordagem para Identificação de Interesses em documento de requisitos?
 - a) () sim.
 - b) () não.

Se sim, qual(is)?

4. Você já trabalhou com alguma abordagem para identificação de interesses em documento de requisitos?
 - a) () sim.
 - b) () não.

Se sim, qual(is)?

GABARITO DO QUESTIONÁRIO

Questão 1: A (0 pontos), B (1 ponto), C (2 pontos).

Questão 2: A (0 pontos), B (1 ponto), C (2 pontos), D (3 pontos).

Questão 3: A (2 pontos), *ObasCId* (2 pontos), *Theme/Doc* (2 pontos), B (0 pontos).

Questão 4: A (2 pontos), *ObasCId* (2 pontos), *Theme/Doc* (2 pontos), B (0 pontos).

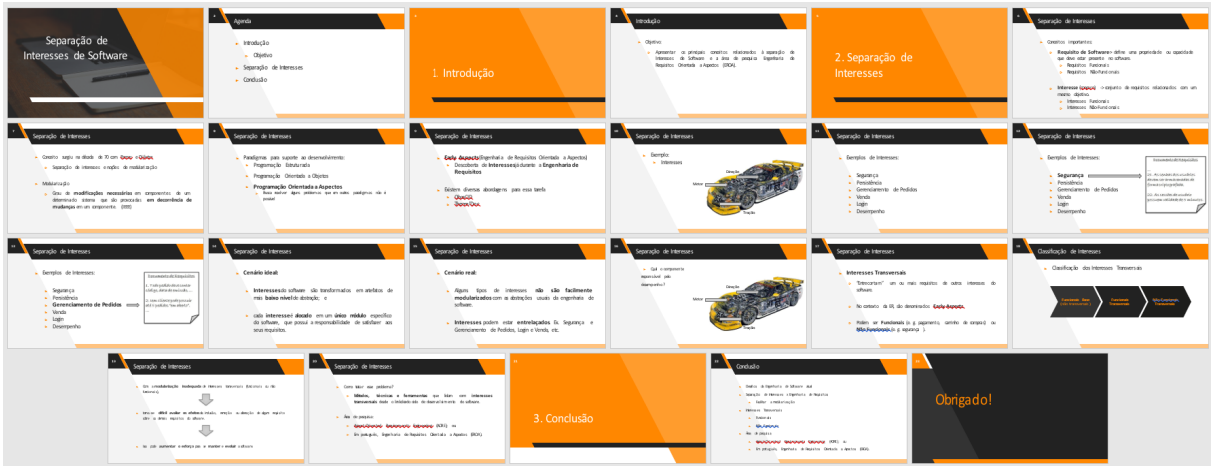
Resultado 0, 1 ou 2 = Nível Básico.

Resultado 3, 4 ou 5 = Nível Intermediário.

Resultado maior que 5 = Nível Avançado.

APÊNDICE B – DOCUMENTAÇÃO DOS TREINAMENTOS REALIZADOS

Figura 15 – Treinamento Interesses de Software.



Fonte: Elaborado pelo autor.

Figura 16 – Treinamento abordagem ObasCid.



Fonte: Elaborado pelo autor.

Figura 17 – Treinamento abordagem Theme/Doc.

The figure displays a sequence of 20 presentation slides for a training on the Theme/Doc approach. The slides are arranged in a grid and cover the following content:

- Slide 1:** Title slide: "Abordagem Theme/Doc".
- Slide 2:** "Agenda" with bullet points: "Introdução", "Objetivo", "Abordagem Theme/Doc".
- Slide 3:** "1. Introdução".
- Slide 4:** "Introdução" with "Objetivo": "Aprender a abordagem Theme/Doc para identificação de interesses de software a partir de documentos de requisitos".
- Slide 5:** "1. Abordagem Theme/Doc".
- Slide 6:** "Abordagem Theme/Doc" with bullet points: "Theme/Doc é uma abordagem para identificar os requisitos orientada a Apoiar os seus interesses de software a partir de documentos de requisitos.", "Baseia-se no **Active View** (visualização de ações)".
- Slide 7:** "Abordagem Theme/Doc" with bullet points: "O que é uma ação-chave? Uma unidade de abstração para gerar uma **Ação-Chave**", "Lista de ações-chave (também identificadas pelo significado no contexto do documento de requisitos). As ações-chave podem ser: Requisitos Funcionais e Requisitos Não-Funcionais", "Documento de Requisitos".
- Slide 8:** "Abordagem Theme/Doc" with bullet points: "Exemplo: Software de Controle de Tráfego", "O que é uma ação-chave?", "O que é uma ação-chave de ação?", "O que é uma ação-chave de ação?", "Requisitos de software (requisitos de requisitos)".
- Slide 9:** "Abordagem Theme/Doc" with bullet points: "Exemplo: Identificação de ações-chave", "O que é uma ação-chave?", "O que é uma ação-chave de ação?", "O que é uma ação-chave de ação?", "Requisitos de software (requisitos de requisitos)".
- Slide 10:** "Abordagem Theme/Doc" with bullet points: "Identificação das ações-chave", "O que é uma ação-chave em um documento de requisitos?", "Requisitos de software (requisitos de requisitos)", "O que é uma ação-chave de ação?".
- Slide 11:** "Abordagem Theme/Doc" with bullet points: "Lista de ações-chave:", "R", "I", "S", "O", "B".
- Slide 12:** "Abordagem Theme/Doc" with bullet points: "O que é uma ação-chave?", "As ações-chave são apresentadas em um formato de matriz", "Os requisitos são apresentados em um retângulo", "Diagrama de uma matriz de ações-chave".
- Slide 13:** "Abordagem Theme/Doc" with bullet points: "O que é uma ação-chave?", "Relacionar com uma lista de requisitos de ação", "Diagrama de uma matriz de ações-chave".
- Slide 14:** "Abordagem Theme/Doc" with bullet points: "O que é uma ação-chave?", "Relacionar com uma lista de requisitos de ação", "Diagrama de uma matriz de ações-chave".
- Slide 15:** "Abordagem Theme/Doc" with bullet points: "No documento Theme/Doc, os requisitos são identificados analisando esse tipo de requisito", "Diagrama de uma matriz de ações-chave".
- Slide 16:** "Abordagem Theme/Doc" with bullet points: "A ideia é separar e lidar ações e requisitos em dois grupos base e transversal.", "Base/Transversal, ou seja, não possui requisitos que se referem a ações de outro grupo.", "Transversal possui requisitos que se referem a ações de outro grupo.", "Diagrama de uma matriz de ações-chave".
- Slide 17:** "Abordagem Theme/Doc" with bullet points: "Para lidar uma ação-chave em grupo, devemos:", "requisitos de requisitos que se referem a mais de uma ação para determinar qual é a ação principal dos requisitos", "No requisito de **base** a ação principal", "Aprender qual requisito de uma matriz de requisitos de **base**", "O requisito de **transversal** não possui base e é **base**, no grupo base.", "Diagrama de uma matriz de ações-chave".
- Slide 18:** "Abordagem Theme/Doc" with bullet points: "No documento Theme/Doc, os requisitos são identificados analisando esse tipo de requisito", "Diagrama de uma matriz de ações-chave".
- Slide 19:** "Abordagem Theme/Doc" with bullet points: "No documento Theme/Doc, os requisitos são identificados analisando esse tipo de requisito", "Diagrama de uma matriz de ações-chave".
- Slide 20:** "Obrigado!".

Fonte: Elaborado pelo autor.

APÊNDICE C – TERMO DE CONSENTIMENTO

Figura 18 – Termo de Consentimento.

Avaliação de duas Abordagens para Identificação de Interesses de Software na Engenharia de Requisitos

Termo de Consentimento Livre e Esclarecido

Alegrete, 12 de setembro de 2018

Você está sendo convidado(a) para participar, como voluntário(a), de um estudo para avaliação de duas abordagens para identificação de interesses de software na engenharia de requisitos. Este estudo é parte integrante do trabalho de conclusão de curso para graduação de Guilherme Legramante Martins, o qual está sob orientação do prof. Me. João Pablo Silva da Silva.

Por meio deste documento, a qualquer momento, você poderá solicitar esclarecimentos adicionais sobre o estudo ou sobre a pesquisa. Também poderá retirar seu consentimento ou interromper a participação a qualquer momento, sem sofrer qualquer tipo de penalidade ou prejuízo. Ao participar deste estudo você não terá nenhum custo, nem receberá qualquer vantagem financeira. Seu nome e identidade serão mantidos em sigilo. Os resultados deste estudo serão armazenados pelo pesquisador responsável e poderão ser divulgados em publicações científicas.

Para participar deste estudo, você precisará: assistir aos treinamentos oferecidos, ler o material fornecido e realizar as atividades solicitadas. Ao participar deste estudo você corre o risco de frustrar-se por não conseguir realizar as atividades solicitadas. Esperamos que os resultados deste estudo contribuam para minimização da complexidade da identificação de interesses de software na engenharia de requisitos.

Após esses esclarecimentos, solicitamos o seu consentimento de forma livre para que participe deste estudo. Para tanto, preencha os itens que se seguem.

Eu, _____, tendo em vista as informações acima apresentadas, de forma livre e esclarecida, aceito participar deste estudo.

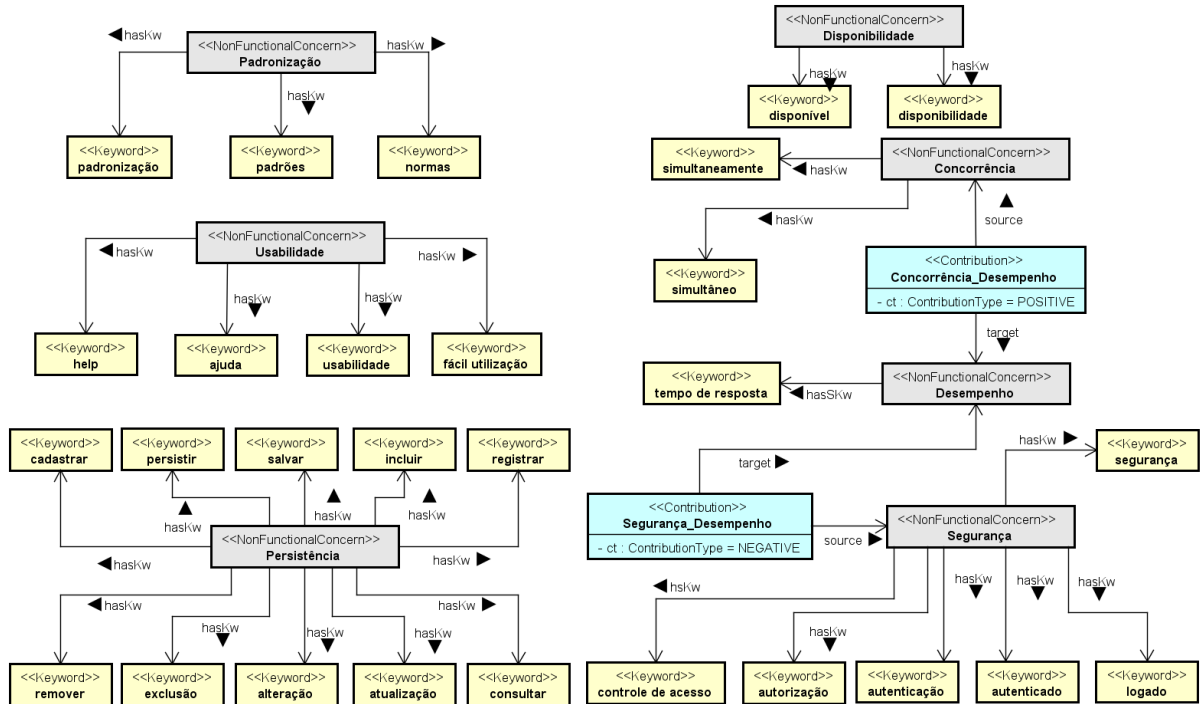
Assinatura do Participante

Fonte: Elaborado pelo autor.

Anexos

ANEXO A – CATÁLOGO DE INTERESSES DE SOFTWARE UTILIZADO NA ABORDAGEM OBASCID

Figura 19 – Catálogo de Interesses de Software.



Fonte: Parreira Jr. (2018).

ANEXO B – DOCUMENTAÇÃO DE REQUISITOS - EXPERIMENTO

Figura 20 – Requisitos ObasCId Tool.

Requisitos ObasCId Tool

1. Com o auxílio do catálogo de requisitos não funcionais, identifique os interesses de software e o interesse principal nos requisitos a seguir:

Requisito R-01	Interesses	Principal
A interface de todas as funções do software deve ser responsiva, de forma que os elementos da mesma se adaptem a dispositivos com telas menores (tais como, smartphones e tablets) com no mínimo 5 polegadas.		

Requisito R-02	Interesses	Principal
Para utilizar as funções do software, o usuário deve estar autenticado por meio de seu <u>email</u> e senha.		

Requisito R-03	Interesses	Principal
O software deve permitir a importação/exportação das palavras-chave de um interesse por meio de arquivos de texto.		

Requisito R-04	Interesses	Principal
O software deve permitir o cadastramento, alteração e exclusão de catálogos de interesses de software. Cada catálogo deve conter nome, descrição (opcional) e tipo de licença, que pode ser pública ou privada.		

Requisito R-05	Interesses	Principal
O software deve permitir a alteração do perfil de pesquisadores. Todos os dados de um pesquisador, com exceção do seu e-mail, podem ser atualizados.		

2. Construa a matriz de entrelaçamento de acordo com mostrado no treinamento;

Fonte: Parreira Jr. (2018).

Figura 21 – Requisitos Sistema de Gerenciamento de Cursos.

Requisitos Sistema de Gerenciamento de Cursos		
1. Identifique as ações chave e ação principal nos requisitos a seguir:		
Requisito R-01	Ações Chave	Ação Principal
Os alunos podem se inscrever nos cursos.		
Requisito R-02	Ações Chave	Ação Principal
Os alunos podem cancelar a inscrição nos cursos.		
Requisito R-03	Ações Chave	Ação Principal
Para um aluno se inscrever em um curso ele deverá estar logado.		
Requisito R-04	Ações Chave	Ação Principal
Para um aluno se cancelar a inscrição em um curso ele deverá estar logado.		
Requisito R-05	Ações Chave	Ação Principal
Os professores podem cancelar a inscrição nos cursos.		
Requisito R-06	Ações Chave	Ação Principal
Para um professor cancelar a inscrição de um aluno em um curso ele deverá estar logado.		
Requisito R-07	Ações Chave	Ação Principal
Para um professor cancelar a inscrição de um aluno em um curso ele deverá marcá-lo como especial.		
Requisito R-08	Ações Chave	Ação Principal
Professores podem dar nota para cursos.		
Requisito R-09	Ações Chave	Ação Principal
Para um professor dar nota para um curso ele deverá estar logado.		
1. Construa a Action View de acordo com mostrado no treinamento:		

Fonte: Baniassad (2004).

Figura 22 – Gabarito dos requisitos utilizados no experimento.

Software – Sistema de Gerenciamento de Cursos

Elementos Relevantes:

1. Marcar (Transversal)
2. Cancelar a inscrição
3. Logar (Transversal)
4. Inscrever
5. Dar nota

Software – ObasCId Tool

Elementos Relevantes:

1. Login (Transversal)
2. Usabilidade (Transversal)
3. Importar
4. Cadastrar
5. Alterar

Fonte: Parreira Jr. (2018) e Baniassad (2004).

ÍNDICE

ACM, 35, 37

AOFA, 24, 41, 52

EAP, 25

ER, 9, 23, 24, 26, 28, 29, 32–34, 36, 38,
39

EROA, 15, 23–27, 29, 30, 32–34, 41, 55

IEEE, 19, 34, 35, 37

OA, 23, 24, 29, 33, 39, 40

PLN, 39, 40

RSL, 9, 25, 26, 30–34, 36, 37, 39–41, 51,
69