

UNIVERSIDADE FEDERAL DO PARÁ  
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

PAULO ROBSON CAMPELO MALCHER

**CATÁLOGO DE ABORDAGENS DE APOIO À  
RASTREABILIDADE DE REQUISITOS BASEADO EM UMA  
REVISÃO SISTEMÁTICA DA LITERATURA**

Belém  
2015

Paulo Robson Campelo Malcher

**CATÁLOGO DE ABORDAGENS DE APOIO À  
RASTREABILIDADE DE REQUISITOS BASEADO EM UMA  
REVISÃO SISTEMÁTICA DA LITERATURA**

Dissertação de Mestrado apresentada para a obtenção do grau de Mestre em Ciência da Computação no Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.

Área de Concentração Engenharia de Software.

Orientador Prof. Dr. Sandro Ronaldo Bezerra Oliveira.

Belém  
2015

Dados Internacionais de Catalogação-na-Publicação (CIP)  
Sistema de Bibliotecas da UFPA

---

Malcher, Paulo Robson Campelo, 1989-

Catálogo de abordagens de apoio rastreabilidade de requisitos baseado em uma revisão sistemática da literatura / Paulo Robson Campelo Malcher. - 2015.

Orientador: Sandro Ronaldo Bezerra Oliveira.

Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Ciências Exatas e Naturais, Programa de Pós-Graduação em Ciência da Computação, Belém, 2015.

1. Engenharia de software-Gerência. 2. Projeto de sistemas. 3. Rastreabilidade de requisitos. 4. Projeto de sistemas-Metodologia-Confabilidade. 5. Engenharia de software baseada em evidências. I. Título.

CDD 22. ed. 005.12

---

Paulo Robson Campelo Malcher

**CATÁLOGO DE ABORDAGENS DE APOIO À  
RASTREABILIDADE DE REQUISITOS BASEADO EM UMA  
REVISÃO SISTEMÁTICA DA LITERATURA**

Dissertação de Mestrado apresentada para a obtenção do grau de Mestre em Ciência da Computação no Programa de Pós Graduação em Ciência da Computação do Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará.

Data da aprovação: Belém-PA.

Banca Examinadora

Prof. Dr. Sandro Ronaldo Bezerra Oliveira  
Programa de Pós Graduação em Ciência da Computação - UFPA – Orientador

Prof<sup>ª</sup>. Dra. Carla Alessandra Lima Reis  
Faculdade de Computação - Instituto de Ciências Exatas e Naturais- UFPA –  
Membro Externo

Prof<sup>ª</sup>. Dra. Marianne Kogut Eliasquevici  
Faculdade de Computação - Instituto de Ciências Exatas e Naturais- UFPA –  
Membro Externo

## AGRADECIMENTOS

Em primeiro lugar quero agradecer a Deus, por me conceder mais essa vitória, por nunca ter me abandonado e também por não permitir que eu abandonasse o trabalho desenvolvido em seu nome.

A minha família tão amada e um dos maiores presentes dado a mim por Deus, Roberto (in memoriam), que sempre se esforçou ao máximo para me ajudar em minhas necessidades e que infelizmente não estará neste momento de grande importância em minha vida e aos meus irmãos Vera e Rossi por seu apoio incondicional.

Aos professores do Mestrado, que contribuíram não somente para minha formação acadêmica, mas também com ensinamentos que levarei para a vida. Em especial ao professor Sandro, pelo carinho, paciência e auxílio de sempre.

Aos colegas de trabalho do projeto SPIDER, que, sem exceção, contribuíram direta ou indiretamente para a realização deste trabalho. Em especial ao Diogo que teve importante participação no desenvolvimento deste trabalho.

Aos meus amigos que sempre estiveram presentes durante essa caminhada, me deram forças nos momentos difíceis e me repreenderam nos momentos de distração. Agradeço também a Kaliu Andrade que esteve comigo na reta final deste trabalho e foi um grande incentivador para sua conclusão.

Enfim, agradeço a todos que acreditaram e torceram por mim.

"Deus não escolhe os capacitados,  
capacita os escolhidos".

Albert Einstein

## RESUMO

Rastreabilidade de Requisitos é uma atividade de grande importância no contexto da Gerência de Requisitos em Projetos de software e pode ser definida como a atividade que acompanha um requisito em todo seu ciclo de vida. Nesse sentido, diversas abordagens (técnicas, modelos de processos, frameworks de processo, ferramenta, metodologias e afins) de apoio a rastreabilidade encontram-se dispersas na literatura para que se possam atingir os níveis de produtividade e qualidade desejados. Neste contexto, este trabalho visa contribuir por meio de um catálogo de abordagens de apoio a atividade de rastreabilidade de requisito, que é um instrumento que reúne estas abordagens a fim de facilitar e ajudar organizações na sua escolha. Para alcançar tal resultado foi utilizado o método da Engenharia de Software Baseada em Evidências chamado Revisão Sistemática da Literatura, devido a este método ser confiável, rigoroso e auditável, tendo assim, como objetivo investigar na literatura tais abordagens e boas práticas de rastreabilidade de requisitos no contexto de projetos de software.

**PALAVRAS-CHAVE:** Engenharia de Software Baseada em Evidências, Gerência de Requisitos, Rastreabilidade de Requisitos, Revisão Sistemática da Literatura.

## **ABSTRACT**

Requirements Traceability is a very important activity in the context of Requirements Management Software Projects and can be defined as the activity that accompanies a requirement throughout its life cycle. In this sense, different approaches (technical, process models, process frameworks, tools, methodologies and related stuffs) to support traceability are dispersed in the literature so that they can achieve the desired levels of productivity and quality. In this context, this work aims to contribute through a catalog of approaches to support the requirement for traceability of activity, which is an instrument that combines these approaches to facilitate and assist organizations in their choice. To achieve this result, we used the method Evidence-Based Software Engineering called Systematic Literature Review, due to this method is reliable, accurate and auditable, thus, the objective of investigating such approaches in the literature and good requirements traceability practices context of software projects.

**KEYWORDS:** Evidence-Based Software Engineering, Requirements Management, Requirements Traceability, Systematic Literature Review



## LISTA DE FIGURAS

Figura 2.1 - Elementos do Processo .....	11
Figura 2.2 - Modelo de atividades Coarse-grain do processo de Engenharia de Requisitos, adaptado de (KOTONYA e SOMMERVILLE, 1998). .....	13
Figura 2.3 - Fases da ER com o acréscimo da Gerência de Requisitos.....	15
Figura 2.4 - Classificação de Rastreabilidade em Pré e Pós-rastreabilidade segundo Gotel e Finkelstein (1994). .....	21
Figura 2.5 - Estrutura do MR-MPS-SW (adaptado de Koscianski e Soares, 2007).....	22
Figura 2.6 - Grupos de processos de acordo com o ciclo de vida de um projeto (adaptado de PMI, 2014) .....	24
Figura 2.7 - Processo para gerenciamento de riscos sugerido pelo padrão ISO/IEC 16085:2006 (IEEE, 2006).....	27
Figura 2.8 - Componentes do MPS.BR (SOFTEX, 2012) .....	29
Figura 2.9 - Equivalência entre Níveis do MR-MPS-SW e níveis do CMMI-DEV (adaptado de SOFTEX, 2012) .....	33
Figura 3.1 - Relação entre Estudos Primários e Estudos Secundários (MAFRA; BARCELOS e TRAVASSOS, 2006).....	39
Figura 3.2 - Número de Estudos retornados a partir da Busca Primária. ....	54
Figura 3.3 - Percentual de estudos retornados a partir da Busca Primária. ....	54
Figura 3.4 - Número de Estudos ao Longo dos Anos .....	60
Figura 3.5 - Número de Estudos por Autor .....	60
Figura 3.6 - Número de Estudos por Instituição.....	61
Figura 3.7 - Número de Estudos por País .....	61
Figura 3.8 - Trabalhos por Tipo de Publicação .....	62
Figura 3.9 - Percentual de Estudos por Categoria de Qualidade .....	68
Figura 4.1 - Relações relacionadas para a classe java.awt.dnd.dragsource.....	78
Figura 4.2 - Níveis de análise de NLP em artefatos textuais (LEAL et al., 2008). .....	98

## LISTA DE QUADROS

Quadro 2.1 - Níveis de maturidade do MR-MPS-SW (SOFTEX, 2012.).....	32
Quadro 3.1 - Seleção de Fontes de Pesquisa .....	49
Quadro 3.2 - Meios que mais Publicaram Estudos.....	63
Quadro 3.3 - Escala de Likert-5 .....	66
Quadro 3.4 - Escala para cada critério de qualidade .....	66
Quadro 4.1 - Informações gerais sobre a técnica ADL.....	75
Quadro 4.2 - Ferramentas e Principais Referências sobre a técnica ADL .....	76
Quadro 4.3 - Informações gerais sobre a técnica Clustering.....	77
Quadro 4.4 - Ferramentas e Principais Referências sobre a técnica Clustering .....	79
Quadro 4.5 - Informações gerais sobre a técnica Cronstaint Networks .....	80
Quadro 4.6 - Principais Referências sobre a técnica Cronstaint Networks .....	80
Quadro 4.7 - Informações gerais sobre a técnica Cross References.....	81
Quadro 4.8 - Ferramentas e Principais Referências sobre a técnica Cross References .....	82
Quadro 4.9 - Informações gerais sobre a técnica EBT .....	83
Quadro 4.10 - Ferramentas e Principais Referências sobre a técnica EBT.....	84
Quadro 4.11 - Informações gerais sobre a técnica ER-Models.....	85
Quadro 4.12 - Ferramentas e Principais Referências sobre a técnica ER-Models.....	85
Quadro 4.13 - Informações gerais sobre a técnica FORT.....	87
Quadro 4.14 - Principais Referências sobre a técnica FORT .....	88
Quadro 4.15 - Informações gerais sobre a técnica GCT.....	89
Quadro 4.16 - Principais Referências sobre a técnica GCT.....	90
Quadro 4.17 - Informações gerais sobre a técnica Graphs.....	91
Quadro 4.18 - Ferramentas e Principais Referências sobre a técnica Graphs.....	91
Quadro 4.19 - Informações gerais sobre a técnica Hyperlinks .....	92
Quadro 4.20 - Ferramentas e Principais Referências sobre a técnica Hyperlinks .....	93
Quadro 4.21 - Informações gerais sobre a técnica KP .....	94
Quadro 4.22 - Ferramentas e Principais Referências sobre a técnica KP .....	94
Quadro 4.23 - Informações gerais sobre a técnica LSI.....	96
Quadro 4.24 - Ferramentas e Principais Referências sobre a técnica LSI .....	97
Quadro 4.25 - Informações gerais sobre a técnica NLP .....	99
Quadro 4.26 - Ferramentas e Principais Referências sobre a técnica NLP .....	99
Quadro 4.27 - Informações gerais sobre a técnica PNM.....	101
Quadro 4.28 - Ferramentas e Principais Referências sobre a técnica PNM .....	101
Quadro 4.29 - Informações gerais sobre a técnica RBT .....	103
Quadro 4.30 - Ferramentas e Principais Referências sobre a técnica RBT.....	103
Quadro 4.31 - Informações gerais sobre a técnica Scenarios Based.....	105
Quadro 4.32 - Ferramentas e Principais Referências sobre a técnica Scenarios Based .....	105
Quadro 4.33 - Informações gerais sobre a técnica Templates .....	106
Quadro 4.34 - Principais Referências sobre a técnica Templates .....	106
Quadro 4.35 - Informações gerais sobre a técnica Traceability Matrix .....	107
Quadro 4.36 - Ferramentas e Principais Referências sobre a técnica Traceability Matrix ....	108
Quadro 4.37 - Informações gerais sobre a técnica VBRT.....	110
Quadro 4.38 - Principais Referências sobre a técnica VBRT.....	110
Quadro 4.39 - Informações gerais sobre a técnica VSM.....	111
Quadro 4.40 - Ferramentas e Principais Referências sobre a técnica VSM.....	111

## LISTA DE TABELAS

Tabela 3.1 - Seleção de Estudos Primários .....	57
Tabela 3.2 - Estatísticas de Aplicação dos Critérios de Exclusão e Inclusão.....	58
Tabela 3.3 – Níveis de Qualidade.....	68

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas e Técnicas
ADL	Allocation Dirichlet Latent
CMMI	Capability Maturity Model Integration
CMMI-DEV	CMMI for Development
EBT	Event-Based Traceability
ER	Engenharia de Requisitos
FORT	Feature Oriented Requirements Traceability
GCT	Goal-centric traceability
GRE	Gerência de Requisitos
IEEE	Institute of Electrical and Electronics Engineers
IEC	International Electrotechnical Commission
IR	Information Retrieval
ISO	International Organization for Standardization
KP	Key Phrases
LSI	Lantent Semantic Indexing
MA-MPS	Método de Avaliação MPS.BR
MN-MPS	Modelo de Negócio MPS.BR
MPS.BR	Melhoria do Processo de Software Brasileiro
MR-MPS-SW	Modelo de Referência MPS para Software
NLP	Natural Language Processing
NRF	Non Requirements Functional
PMBOK	Project Management Body of Knowledge
PMI	Project Management Institute
PNM	Probabilistic Network Model
QAM	Quality Assessment Models
RBT	Rules-based Traceability
RSL	Revisão Sistemática da Literatura
SEI	Software Engineering Institute
SG	Specific Goals
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SP	Specific Practices
SPI	Software Process Improvement

SPIDER	Software Process Improvement: Development and Research
VBRT	Value-Based Requirements Tracing
VSM	Vector Space Model

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>1</b>
1.1.	Contexto do Trabalho .....	1
1.2.	Motivação.....	3
1.3.	Objetivos.....	4
1.4.	Metodologia do Trabalho.....	5
1.5.	Estrutura do Trabalho .....	7
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>8</b>
2.1.	Processo de Software: Uma Visão Geral .....	8
2.2.	Engenharia de Requisitos: Uma Visão Geral.....	12
2.2.1.	Gerência de Requisitos .....	15
2.2.2.	Rastreabilidade de Requisitos.....	16
2.3.	Modelos de Qualidade .....	22
2.3.1.	ISO/IEC 12207 .....	23
2.3.2.	O Modelo CMMI-DEV .....	25
2.3.3.	O Modelo MR-MPS-SW .....	27
2.4.	Trabalhos Relacionados .....	33
2.5.	Considerações Finais .....	35
<b>3</b>	<b>UMA REVISÃO SISTEMÁTICA DA LITERATURA PARA RASTREABILIDADE DE REQUISITOS</b> .....	<b>37</b>
3.1.	Engenharia de Software Baseada em Evidências .....	37
3.2.	Revisão Sistemática da Literatura .....	40
3.3.	Metodologia de uma Revisão Sistemática da Literatura .....	42
3.3.1	Planejamento da revisão .....	42
3.3.2	Condução da revisão.....	44
3.3.3	Apresentação da revisão .....	45
3.4.	Planejamento da Revisão .....	45
3.4.1	Objetivo da Revisão.....	45
3.4.2	Questão de Pesquisa Principal .....	46
3.4.3	Estrutura da Questão de Pesquisa Principal .....	46
3.4.4	Questões de Pesquisa Secundárias .....	47
3.4.5	Escopo e Recursos da Pesquisa .....	47
3.4.6	Seleção de Fontes .....	49
3.4.7	Busca Primária.....	49
3.4.7.1	Identificação de Palavras-Chaves e Sinônimos .....	50
3.4.7.2	Geração de Strings de Busca .....	51
3.5.	Condução da Revisão .....	55
3.4.8	Seleção dos Estudos Primários .....	55
3.4.8.1	CrITÉrios de Inclusão e Exclusão dos Estudos Primários.....	56
3.4.8.2	Processo de Seleção dos Estudos Primários .....	57
3.4.9	Avaliação da Qualidade dos Estudos Primários .....	63
3.4.9.1	Processo de Avaliação de Qualidade.....	65
3.4.10	Estratégia de Extração de Informações .....	69
3.4.11	Sumarização dos Resultados .....	70
3.6.	Considerações Finais .....	70

<b>4</b>	<b>UM CATÁLOGO DE ABORDAGENS DE APOIO A RASTREABILIDADE DE REQUISITOS .....</b>	<b>72</b>
<b>4.1.</b>	<b>Catálogo de Abordagens de Apoio a Rastreabilidade de Requisitos .....</b>	<b>73</b>
4.1.1.	Organização do Catálogo.....	73
4.1.2.	Técnicas.....	75
4.1.3.	Aplicação do Catálogo.....	113
<b>4.2.</b>	<b>Avaliação do Catálogo.....</b>	<b>114</b>
<b>4.3.</b>	<b>Considerações Finais .....</b>	<b>117</b>
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>119</b>
<b>5.1.</b>	<b>Sumarização dos Resultados .....</b>	<b>119</b>
<b>5.2.</b>	<b>Contribuições.....</b>	<b>120</b>
<b>5.3.</b>	<b>Limitações. ....</b>	<b>121</b>
<b>5.4.</b>	<b>Trabalhos Futuros.....</b>	<b>121</b>
5.4.1.	Evolução da Revisão Sistemática.....	121
5.4.2.	Aplicação Prática do Catálogo .....	122
5.4.3.	Expansão do Catálogo .....	122
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>123</b>
	<b>APÊNDICE A – PROTOCOLO DE REVISÃO .....</b>	<b>134</b>
	<b>APÊNDICE B – ESTUDOS INCLUÍDOS NA PESQUISA.....</b>	<b>153</b>
	<b>APÊNDICE C – FORMULÁRIO PARA AVALIAÇÃO DE ESPECIALISTA... ..</b>	<b>172</b>

# 1 INTRODUÇÃO

Neste capítulo serão apresentados: a contextualização, a motivação, os objetivos, a metodologia e, finalmente, a estrutura desta dissertação. O objetivo deste capítulo é prover uma visão geral deste trabalho e estabelecer o roteiro a ser seguido durante o desenvolvimento deste.

## 1.1. Contexto do Trabalho

Com o objetivo de tornar as organizações mais competitivas nos negócios as respostas das tecnologias de informação têm a necessidade de serem cada vez mais rápidas, porém, a qualidade do que é entregue, ou influencia essa entrega, é também um fator muito importante para manter essa competitividade (FERREIRA, 2010). Nesse sentido, existem diversas propostas de soluções para auxiliar na qualidade de software (modelos, padrões, metodologias).

A fim de ajudar os engenheiros de software a compreender melhor o problema que eles vão trabalhar para resolver, surgiu então a engenharia de requisitos. A engenharia de requisitos contempla um conjunto de atividades que direcionam ao entendimento da regra de negócio que o software irá impactar, o retorno que o cliente espera do software e da interação final do usuário com o software (PRESSMAN, 2011). Dentre as atividades que compõem a engenharia de requisitos, pode-se dividi-las em quatro áreas: Elicitação, Análise, Documentação e o Gerenciamento (GENVIGIR, 2009). Este trabalho contempla a área de gerenciamento de requisitos, ou seja, a área que abrange toda compreensão e o controle de mudanças nos requisitos de sistema.

Como abordagens que auxiliam na qualidade no desenvolvimento de software, tem-se o padrão da Norma ISO 12207 (ISO, 2008) e os modelos CMMI (SEI, 2010) e MPS.BR (SOFTEX, 2012), onde a Gerência de Requisitos é tida como essencial para o sucesso de um projeto. Um dos motivos para isso, é que erros na especificação de



requisitos e a falta de controle das alterações aplicadas sobre ela fazem com que os erros propaguem-se pelo ciclo de desenvolvimento, sendo encontrados apenas na fase de testes. A correção desses erros de forma tardia, na fase de testes, requer um esforço 30 vezes superior ao de correção durante as etapas iniciais do projeto (BOEHM, 1981).

Devido a essas mudanças que podem ocorrer ao longo da vida dos sistemas, dentro da gestão de requisitos deve haver uma rastreabilidade dos mesmos. A atividade de rastreabilidade de requisitos visa controlar a evolução e a agregação dos requisitos ao projeto de software (GENVIGIR, 2009). Neste contexto, percebe-se uma grande preocupação em se ter uma visão dos relacionamentos que envolvem os requisitos. De fato, o sucesso da gerência de requisitos, conseqüentemente dos projetos de software, depende diretamente de quão bem definidos e conhecidos são estes relacionamentos (LETÉLIER, 2002; COEST, 2011 apud LEAL, 2011).

Segundo a SOFTEX (2012), a gerência de requisitos envolve identificar os requisitos do produto e dos componentes do produto do projeto, bem como estabelecer e manter um acordo entre o cliente e a equipe de projeto sobre esses requisitos. Também é objetivo da gerência de requisitos controlar e tratar as mudanças nos requisitos ao longo do desenvolvimento. Neste sentido, para apoiar o processo de mudança de requisito, é fundamental definir e manter a rastreabilidade destes.

Assim, realizar uma pesquisa que tenha como foco descobrir quais as abordagens descritas na literatura para auxiliar a rastreabilidade de requisitos, pode ser de grande valia para pesquisas futuras, já que pode trazer como resultados: onde os principais autores da área estão concentrando seus esforços; quais as técnicas mais utilizadas; quais padrões e metodologias auxiliam em seu desenvolvimento; quais ferramentas de apoio são utilizadas para sua implantação; e diversas outras informações a respeito do tópico de busca.

O método utilizado para realizar essa pesquisa foi a Revisão Sistemática da Literatura (RSL), uma revisão transparente, devido apresentar toda sua estratégia de pesquisa, restrições e critérios de avaliação. Além disso, é uma revisão estruturada, pois não permite ao autor uma análise ad hoc, nem a presença de vieses por parte deste. Segundo Kitchenham (2007), uma revisão sistemática da literatura é uma forma de avaliar e interpretar todas as pesquisas disponíveis, referentes a uma questão de investigação particular, área temática ou fenômeno de interesse. Podendo definir, assim, esta revisão, como uma revisão abrangente e não tendenciosa. Ele afirma, ainda,

que revisões sistemáticas têm por objetivo apresentar uma avaliação justa de um tópico de investigação, usando uma metodologia confiável, rigorosa e auditável.

É objetivo do projeto aqui descrito definir um catálogo de abordagens para Rastreabilidade de Requisitos disponíveis na literatura, a fim de reuni-las em um único banco de conhecimento e apresentar, com base na literatura, qual o melhor contexto de aplicação que cada uma enquadra-se. Este catálogo, no contexto deste trabalho, será um instrumento onde organizações desenvolvedoras/mantenedoras de softwares ou mesmo pessoas interessadas na área possam encontrar diversas abordagens a respeito de rastreabilidade de requisitos a fim de ajuda-las na definição de seus processos para a implantação desta atividade e na escolha de uma técnica que mais se adeque ao seu contexto.

Este projeto também tem o objetivo de aplicar técnicas de Revisão Sistemática da Literatura, desde seu planejamento até a apresentação dos resultados, para fazer uma análise crítica da literatura, no que diz respeito à rastreabilidade de requisitos e assim reunir as abordagens (por “abordagens” o que se pretende investigar são os padrões, que para fins deste projeto, por padrões entendem-se modelos de processos, frameworks de processo, metodologias, técnicas, ferramentas e afins) descritas na literatura que possam auxiliar essa área, e conseqüentemente na definição do catálogo.

## **1.2. Motivação**

Em Agosto de 2006, foi realizado o Primeiro Workshop de Grandes Desafios para a Rastreabilidade (GCW'06), onde estiveram reunidos membros da comunidade de rastreabilidade da academia, indústria e do governo dos Estados Unidos da América (EUA). Como resultado desse Workshop foi desenvolvido pelo Centro de Excelência para Rastreabilidade um relatório técnico denominado de Grandes Desafios para Rastreabilidade (CLELAND-HUANG et al, 2006).

Este relatório apresenta 14 subáreas de atuação da Rastreabilidade (aspectos importantes da Rastreabilidade), e destaca 40 princípios (consensos na comunidade para cada subárea) e 40 desafios que ainda persistem mesmo nos estudos mais atuais sobre a rastreabilidade. Alguns desses grupos, princípios e desafios serviram de motivação para este trabalho e serão detalhados a seguir.

No que diz respeito à subárea conhecimento sobre a rastreabilidade, pode-se definir como um princípio de que a rastreabilidade é um fator crítico para o sucesso de

projetos de software, porém não há um consenso sobre as melhores técnicas e métodos para sua utilização. Como desafio dessa subárea tem-se a criação de um banco de conhecimento, melhores práticas, terminologias padrões, e estudos de caso sobre rastreabilidade.

A inexistência de um banco de conhecimento a respeito das técnicas utilizadas para prover a Rastreabilidade de Requisitos em projetos de software é uma forte motivação para a realização deste trabalho. Além disso, a Rastreabilidade de Requisitos tem sido um aspecto constantemente presente nos Programas de Melhoria da Qualidade do Processo de Software. Tem o seu papel destacado no Capability Maturity Model Integration for Development (CMMI-DEV) e no programa de Melhoria do Processo de Software Brasileiro (MPS.BR), onde quanto mais madura a instituição, maior o escopo da Rastreabilidade de Requisitos. Este aspecto serviu como outro forte motivador para a realização desse trabalho.

Outro aspecto motivador para a realização deste trabalho está na observação dos benefícios que se tem quando se implementa de fato a rastreabilidade de requisitos, como a agilidade na execução de mudanças solicitadas, o menor índice de retrabalho em decorrência destas mudanças e principalmente a possibilidade de acompanhar os requisitos durante todo processo de software e suas relações com os demais produtos de trabalho. Além do que, vê-se que a falta de conhecimento em diversas organizações a respeito de técnicas para uma efetiva implementação da rastreabilidade é um fator que desmotiva seu uso, já que se tem a ideia de ser algo difícil e oneroso.

### **1.3. Objetivos**

Este trabalho tem como objetivo geral propor um catálogo de abordagens de apoio à atividade de rastreabilidade de requisitos. Este catálogo deve apresentar estas abordagens de acordo com características específicas para rastreabilidade de requisitos, sendo baseado em uma revisão sistemática da literatura.

Para atender ao objetivo geral, devem ser contemplados os seguintes objetivos específicos:

- Aprofundar o conhecimento sobre rastreabilidade de requisitos, a fim de auxiliar um melhor planejamento para realizar a revisão sistemática da literatura;

- Definir previamente especificações, tais como, restrições de pesquisas, critérios de inclusão e exclusão de estudos primários, critérios de qualidade para esses estudos, dentre outras, que nortearão a realização da revisão sistemática da literatura;
- Conduzir a Revisão Sistemática da Literatura seguindo as especificações estabelecidas previamente;
- Analisar os resultados da revisão por meio da caracterização dos estudos selecionados;
- Extrair dados, conforme a estratégia estabelecida a priori, a fim de responder as questões de pesquisa da revisão;
- Definir um catálogo contendo abordagens de apoio para rastreabilidade de requisitos.
- Avaliar o catálogo de abordagens com a ajuda de especialistas;
- Publicar resultados de pesquisa em eventos especializados.

#### **1.4. Metodologia do Trabalho**

A realização deste trabalho ocorreu por meio das etapas descritas a seguir:

##### **a) Etapa de Estudo Inicial**

- Estudo geral de trabalhos na área de Engenharia de Software, que forneceu uma visão sobre Gerência de Requisitos e a atividade de Rastreabilidade de Requisitos e das limitações existentes;
- Estudo geral de modelos, normas e guias para processos de software;
- Estudo aprofundado de trabalhos na área Rastreabilidade de Requisitos que serviram de fundamentação para a elaboração do estado da arte deste trabalho.

##### **b) Etapa da Revisão Sistemática**

- Estudo teórico de revisões sistemáticas da literatura aplicadas na área de Engenharia de Software;
- Definição de um protocolo de revisão sistemática;
- Aplicação do protocolo de revisão sistemática;

- Análise dos resultados da aplicação do protocolo de revisão;
- Obtenção de estudos selecionados pela revisão sistemática.

**c) Etapa de Desenvolvimento do Catálogo de Abordagens**

- Extração de Resultados da Revisão Sistemática da Literatura referentes a Abordagens de apoio à rastreabilidade de requisitos;
- Detalhamento das técnicas encontradas na literatura e relacionamento destas com as demais abordagens;
- Avaliação do catálogo proposto com auxílio de um especialista em gerência de requisitos e melhoria de processo;
- Análise das sugestões propostas pelo especialista que realizou a avaliação.

**d) Etapa de Documentação**

- Redação final da dissertação.

Adicionalmente, Silva e Menezes (2001) apontam que existem diversas formas de se classificar uma pesquisa, com base na literatura especializada. Assim, neste contexto, pode-se caracterizar a pesquisa realizada neste trabalho como sendo:

- **Quanto à natureza:** apresenta-se como uma pesquisa Aplicada, por objetivar a geração de conhecimentos para a aplicação prática a fim de alcançar à solução de problemas específicos;
- **Quanto à abordagem do problema:** define-se como uma pesquisa Quantitativa e Qualitativa, pois em determinados momentos há necessidade de se traduzir em números opiniões e informações obtidas com o uso de questionários e da revisão sistemática da literatura;
- **Quanto aos objetivos:** trata-se de uma pesquisa Exploratória e Descritiva, já que proporciona um maior entendimento do problema, tornando-o mais explícito, além de envolver levantamento bibliográfico, entrevistas com pessoas com experiência prática, e por fim a utilização de questionários como forma de verificar as características de uma população;
- **Quanto aos procedimentos técnicos:** classifica-se como uma pesquisa Bibliográfica, pois a mesma foi elaborada a partir de materiais publicados

como artigos de periódicos e eventos, livros e materiais disponibilizados na Internet.

## **1.5. Estrutura do Trabalho**

Além deste capítulo, que trata sobre a introdução geral do trabalho realizado, identificando o contexto de seu desenvolvimento, os seus objetivos e a metodologia utilizada para a execução deste trabalho, é descrita a seguir a estrutura dos demais capítulos desta dissertação.

O Capítulo 2 apresenta a fundamentação teórica deste trabalho, que inclui trabalhos relacionados e uma descrição detalhada dos padrões de qualidade utilizados.

O Capítulo 3 apresenta a revisão sistemática da literatura, incluindo desde a fundamentação teórica da mesma até os seus resultados preliminares.

O Capítulo 4 apresenta o Catálogo de Abordagens, o qual agrega técnicas relacionadas às demais abordagens retiradas dos resultados da revisão sistemática da literatura, discutida no Capítulo 3.

E, finalmente, o Capítulo 5 apresenta a conclusão e as contribuições deste trabalho, a indicação de trabalhos futuros e as considerações finais desta dissertação.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica desta dissertação, apresentando os seguintes tópicos em relação à Rastreabilidade de Requisitos:

- Processo de Software;
- Engenharia de Requisitos;
  - Gerência de Requisitos;
  - Rastreabilidade de Requisitos;
- Normas, Modelos, e Guias de boas práticas de qualidade de processo de software aplicadas à Rastreabilidade de Requisitos;
- Trabalhos Relacionados;
- Considerações Finais do capítulo;

### 2.1. Processo de Software: Uma Visão Geral

Humphrey (1989) interpreta o conceito de processo de software dando enfoque às necessidades do cliente, definindo então este, como um conjunto de atividades relacionadas de Engenharia de Software, que são necessárias para produzir software a partir dos requisitos do usuário. Já Fuggeta (2000) define como um conjunto de políticas, estruturas organizacionais, tecnologia, procedimentos e artefatos coerentes, necessários para conceber, desenvolver implantar e manter um produto de software. Gruhn (2002) apresenta o processo de software como o conjunto de atividades, métodos, práticas e transformações que guiam pessoas na produção de software. Sendo assim, um processo de software define o conjunto de atividades que serão conduzidas no contexto do projeto, os recursos (software, hardware e pessoas) necessários, os artefatos (insumos e produtos) e os procedimentos a serem adotados na realização de cada uma das atividades.



Para Sommerville (2011) um processo de software é um conjunto de atividades relacionadas que levam a um produto de software. Já para Pressman (2011), é como uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade, que tem sua importância por propiciar estabilidade, controle e organização para uma atividade, que pode se tornar bastante caótica.

Levando em consideração modelos, guias e normas de qualidade para o desenvolvimento de software, o processo de software é definido pelo guia PMBOK – Project Management Body of Knowledge (PMI, 2014), como uma combinação de atividades inter-relacionadas realizadas, com o intuito de atingir um objetivo, como alcançar resultados, produtos ou serviços. Para o modelo de qualidade CMMI – Capability Maturity Model Integration (SEI, 2010), um processo é definido quando existe documentação que detalha o que é feito, quando, por quem, os itens utilizados e os itens produzidos.

Quanto à classificação, Sommerville (2011) classifica os Processos de Software nas seguintes categorias:

- a) **Processos informais:** a organização não possui uma definição concreta do processo de software, não há um modelo estritamente definido. O processo utilizado é escolhido pela equipe de desenvolvimento. Os processos informais podem utilizar procedimentos formais, como o gerenciamento de configuração, mas os procedimentos a serem utilizados e as relações entre eles não são predefinidos.
- b) **Processos gerenciados:** é utilizado um modelo de processo padrão instanciado para cada projeto, com o intuito de direcionar futuras evoluções. O modelo de processo define os procedimentos utilizados, os cronogramas e as relações entre os procedimentos.
- c) **Processos metódicos:** possui alguns métodos definidos, que são utilizados para normatizar o processo de software parcialmente ou totalmente.
- d) **Processos melhorados:** processos de software que possuem orçamento e procedimentos específicos visando sua melhoria, ou seja, têm como principal objetivo a melhoria contínua. Medições quantitativas de processo podem ser introduzidas, como parte dessas melhorias.

A classificação definida por Sommerville (2011) é bastante útil, pois serve de base para a melhoria do processo multidimensional, ou seja, não são exclusivas, já que

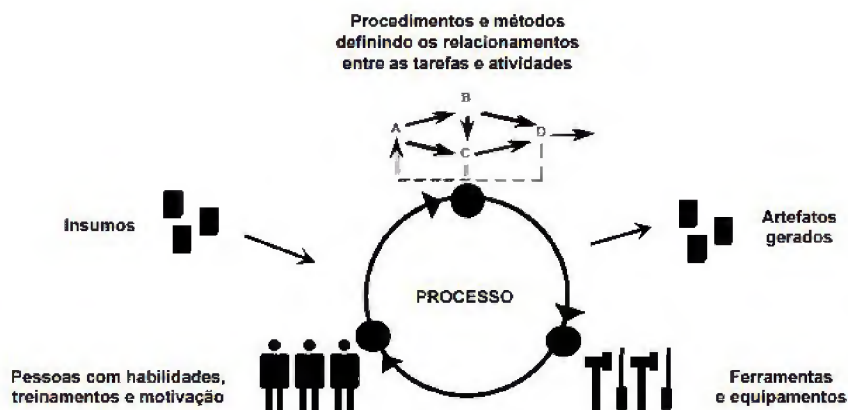


um projeto de uma mesma organização pode ter mais de um tipo de processo, de acordo com as necessidades demandadas.

Alguns principais componentes relacionados à definição de processo de desenvolvimento software são identificados por Falbo (1998) e Travassos (1994), tais como:

- a) **Atividades:** tarefas ou trabalhos a serem realizados, que requerem recursos e podem consumir ou produzir um artefato. Para sua realização, uma atividade pode adotar um procedimento. Uma atividade pode ser decomposta em outras atividades. Além disso, atividades, em qualquer nível, podem depender da finalização de outras atividades.
- b) **Artefatos:** produtos de software produzidos ou consumidos por atividades durante sua realização. Alguns exemplos são: manuais de qualidade, manuais de revisão, diagramas de fluxo de dados, diagramas de objetos, código fonte, entre outros.
- c) **Procedimentos:** são condutas bem estabelecidas e ordenadas para a realização de atividades. Alguns procedimentos podem ser parcialmente automatizados por ferramentas de software. São exemplos de procedimentos: técnicas de avaliação da qualidade, tais como inspeções e walkthroughs, roteiros diversos para a produção de documentos, normas de programação, entre outros.
- d) **Recursos:** pessoas, ferramentas de software, equipamentos ou quaisquer outros recursos necessários à execução de uma atividade. Um recurso humano, especificamente, desempenha um papel na execução das atividades do processo.
- e) **Insumos:** elementos necessários para a realização de uma tarefa ou atividade pode ser um elemento de saída de outras atividades ou tarefas.
- f) **Processos:** agrupamento de atividades relacionadas que têm lugar durante o desenvolvimento de um produto.

Esses componentes podem ser visualizados de melhor forma na Figura 2.1.



**Figura 2.1. Elementos do Processo**

Fonte: PrimeUp - Laboratório de Engenharia de Software – PUC-Rio.

Humphrey (1989) identificou que a busca por um software de qualidade tem sido realizada em duas vertentes: a qualidade do processo e a qualidade do produto. A qualidade do processo tem foco nas tarefas e processos geradores do produto, por meio do controle e gerenciamento do ciclo de vida do software, enquanto a qualidade do produto dedica-se em identificar características tangíveis dos produtos resultantes de um processo, estabelecendo assim critérios que auxiliarão na avaliação do software produzido.

O desenvolvimento de produtos de software confiáveis e dentro do cronograma e custo planejado sempre foi um desafio para a Engenharia de Software (BERTOLLO; FALBO, 2003). O relacionamento entre atividades e componentes pode apresentar grande complexidade na coordenação de tarefas durante o ciclo de vida do software. Assim, a definição de um processo de software possibilita que profissionais da Engenharia de Software trabalhem de forma ordenada (HUMPHREY, 1989). Estudos apontam que uma das principais causas dos problemas enfrentados no desenvolvimento de software é a falta de um processo bem definido e efetivo (FUGGETTA, 2000).

Um processo de software definido permite monitorar o tempo gasto em cada tarefa e o número de defeitos introduzidos ou removidos em cada etapa do desenvolvimento. Esses dados auxiliam a análise do processo e a identificação de problemas e possíveis melhorias (ZAHHRAN, 1998).

A qualidade do processo de software é uma abordagem comumente utilizada para alcançar a qualidade de software (HUMPHREY, 1989), isto acontece através da melhoria do processo de software (Software Process Improvement - SPI).

## 2.2. Engenharia de Requisitos

A fim de ajudar os engenheiros de software a compreender melhor o problema que eles vão trabalhar para resolver, surgiu então a Engenharia de Requisitos (ER). Esta contempla um conjunto de atividades que direcionam ao entendimento da regra de negócio que o software irá impactar, o retorno que o cliente espera do software e da interação final do usuário com o software (PRESSMAN, 2011).

A ER possui grande importância no que tange ao planejamento de um projeto de software, já que devido à alta complexidade dos sistemas desenvolvidos, é necessário um correto entendimento antecipado dos requisitos, antes de um comprometimento de uma solução para o projeto em estudo.

Lamsweerde (2000) define Engenharia de Requisitos como a identificação dos objetivos a serem atingidos pelo futuro sistema, a operacionalização de tais objetivos em serviços e restrições e a atribuição de responsabilidades pelos requisitos resultantes a agentes humanos, dispositivos e software. Já Zave (1997, p. 315) a apresenta como o ramo da engenharia de software que está preocupado com os objetivos do mundo real para as funções e restrições aplicáveis a sistemas de software. Além disso, afirma ainda que a Engenharia de Requisitos também se preocupa com o relacionamento destes fatores para especificações precisas do comportamento do software e com sua evolução no tempo e através de famílias de produtos.

Pressman (2011) afirma que a Engenharia de Requisitos fornece os mecanismos apropriados para entender o que o cliente deseja, analisar suas necessidades, analisar o que é possível ser feito, negociar uma solução razoável, especificar uma solução não ambígua, validar a especificação e gerenciar os requisitos conforme eles transformam-se em um software operante.

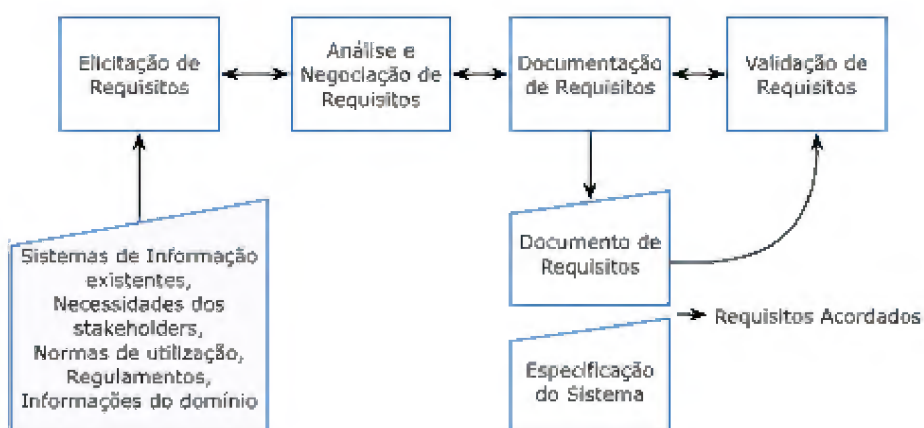
Nuseibeh e Easterbrook (2000) definem Engenharia de Requisitos como a área da engenharia de software que se preocupa com os objetivos reais das funcionalidades do software. Assim, deve agir como uma ponte entre as reais necessidades dos usuários, clientes e outras pessoas afetadas pelo sistema e a solução final.

Segundo Kotonya e Sommerville (1998), a Engenharia de Requisitos consiste nas atividades de descoberta, documentação, validação e manutenção de um conjunto de requisitos de um sistema.

Quanto ao processo da Engenharia de Requisitos, Sommerville e Sawyer (1997)

orientam que este deve ter a definição de quais atividades serão realizadas, a estrutura ou sequência dessas atividades, quem será responsável por cada uma das atividades, a entrada e saída das mesmas e as ferramentas usadas para suporte. Uma boa descrição desse processo fornecerá uma direção para as pessoas envolvidas e reduzirá a probabilidade de que atividades sejam esquecidas. Kotonya e Sommerville (1998) reforçam que o processo de ER pode ser considerado um conjunto estruturado de atividades com o objetivo de derivar, validar e manter um documento de requisitos.

Pode-se afirmar que não existe um processo considerado como ideal para a Engenharia de Requisitos, porém, existem várias propostas dispostas na literatura para modelos de processo. Por conta disso, no contexto deste trabalho será utilizado o modelo proposto por Kotonya e Sommerville (1998), que descreve as seguintes fases do processo de requisitos dispostas na figura 2.2: Elicitação; Análise e Negociação; Documentação (Especificação); Validação; e Gerência de Requisitos.



**Figura 2.2. Modelo de atividades Coarse-grain do processo de Engenharia de Requisitos, adaptado de (KOTONYA e SOMMERVILLE, 1998).**

Com relação às fases apresentadas por Kotonya e Sommerville (1998), pode-se definir:

- 1. Elicitação de Requisitos:** processo por meio do qual os clientes e os desenvolvedores de um sistema de software descobrem, revisam, articulam e compreendem a necessidade do usuário e as restrições que o software deverá apresentar. Além de descobrir quais são as necessidades dos usuários, essa atividade também requer uma cuidadosa análise da organização, do domínio de aplicação e dos processos organizacionais (KOTONYA e SOMMERVILLE, 1998). A elicitação é a primeira atividade no ciclo de vida da engenharia de requisitos e seu papel geral é

obter conhecimento relevante para o problema a ser resolvido. Entretanto, a tarefa da elicitaco   identificar os fatos que compem os requisitos do sistema, de forma a fornecer o mais correto entendimento do que   esperado do sistema de software;

2. **An lise de Requisitos:** processo de an lise das necessidades dos usu rios e clientes para chegar a uma defini co de requisitos de software. O principal objetivo dessa fase   fornecer descri ces abstratas dos requisitos que possam ser facilmente interpretadas. Durante os processos de an lise e negocia co so encontrados problemas com os requisitos, tais como: requisitos esquecidos, conflitantes, amb guos e duplicados. A partir da identifica co desses requisitos com problemas, os usu rios devem discutir, priorizar e negociar at  obterem um acordo com poss veis modifica ces e simplifica ces (ALVES, 2001);
3. **Especifica co de Requisitos:** desenvolvimento de um documento que de forma clara e precisa registre cada requisito do sistema de software.   necess rio que o documento seja compreens vel por todos os envolvidos no processo de engenharia de requisitos, pois servir  como um contrato entre usu rios e desenvolvedores. Diferentes linguagens t m sido propostas para expressar e descrever requisitos, desde a linguagem natural at  a l gica;
4. **Valida co de Requisitos:** Processos asseguram que a especifica co de requisitos de software est  de acordo com os requisitos de sistema, conforme padres de documentos das fases de requisitos e   uma base adequada para a arquitetura (preliminar) da fase de projeto. Segundo Loucopoulus (1995), o processo de valida co de requisitos   definido como a atividade na qual se certifica que o documento de requisitos   consistente com as necessidades dos usu rios;
5. **Ger ncia de Requisitos:** O planejamento e controle da elicitaco, an lise, especifica co e verifica co das atividades de requisitos. A ger ncia   constitu da das atividades relacionadas   manuten co de consist ncia e ao controle de altera ces dos requisitos. O gerenciamento de requisitos   uma atividade muito importante no processo de ER, pois   necess rio no somente escrever os requisitos de forma clara e compreens vel, mas

permitir que possam ser rastreados e gerenciados ao longo da evolução do sistema. O rastreamento de requisitos auxilia na definição de uma documentação completa e com integridade, como também ajuda no processo de gerenciamento de mudanças desses requisitos (TORANZO; CASTRO, 1999).

Apesar da área de Gerência de Requisitos não ter sido claramente apresentada por Kotonya e Sommerville (1998), estes e diversos outros autores a consideram como uma área que está presente em todo o processo da ER, ajudando a equipe a identificar, controlar e detectar mudanças de requisitos ao longo da vida de um projeto. Esta área será mais detalhada na próxima seção deste capítulo. Neste sentido a Figura 2.3. apresenta o processo de ER com o acréscimo da gerência de requisitos.

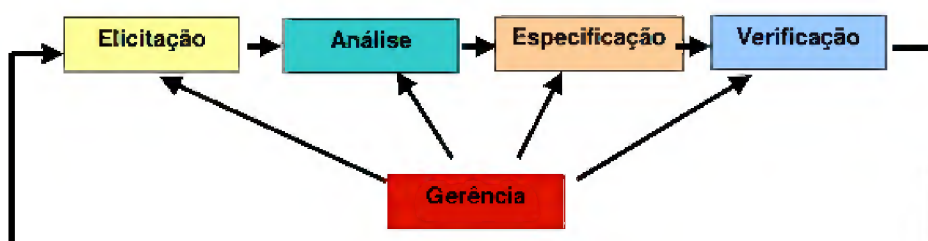


Figura 2.3. Fases da ER com o acréscimo da Gerência de Requisitos

### 2.2.1. Gerência de Requisitos

Este trabalho está focado na área de Gerenciamento de Requisitos. Segundo Spence e Probasco (2000) é a área do processo de desenvolvimento de software que tem como objetivo controlar a criação e evolução dos requisitos. Além de ser, de acordo com Pressman (2011), a área que auxilia a equipe de projeto a identificar, controlar e rastrear os requisitos e suas mutações ao longo do projeto.

A gerência de requisitos tem um importante papel no processo de desenvolvimento de software no que diz respeito ao entendimento comum entre o cliente e a equipe de projeto com relação aos requisitos a serem atendidos pelo. Sendo considerada uma área essencial ao sucesso do projeto. Em vista disso, seus processos são encontrados no primeiro nível de maturidade tanto do MPS.BR (nível G) (SOFTEX, 2012) quanto da representação por estágios do CMMI (nível 2) (SEI, 2010).

O gerenciamento dos requisitos não é uma atividade isolada no início da especificação do sistema, uma vez que corre em paralelo com o processo de desenvolvimento do sistema. Essa atividade implica em gerenciar o escopo para



garantir que todos os requisitos sejam implementados. Além disso, significa controlar alterações nos requisitos com objetivo de evitar alterações de escopo que impactarão nos custos, prazos e na qualidade do projeto.

Pode-se então definir que a qualidade de uma especificação de requisitos é essencial para o adequado andamento de um projeto, mantendo custo, prazo, risco e qualidade do produto dentro do planejado. Além do que, segundo Boehm (1981), a existência de erros na especificação de requisitos de um software podem propagar-se até a implementação, por vezes sendo encontrados apenas na fase de testes, o que requer a revisão de todos os artefatos de desenvolvimento, desde a especificação de requisitos até os casos de teste e implementações.

A fase de gerência de requisitos é a mais dinâmica e atua em todas as fases do processo da ER. De acordo com Cysneiros (2001), uma vez que dificilmente se terá à disposição recursos que permitam satisfazer todos os requisitos dos usuários, estes podem ser classificados como desejáveis ou obrigatórios, utilizando-se um enfoque voltado para a necessidade de priorizar requisitos. Podem também ser classificados como estáveis, quando mudam mais lentamente, ou voláteis, quando mudam mais rapidamente. Essa classificação auxilia a atividade de gerência de requisitos, uma vez que possibilita antecipar mudanças prováveis de requisitos.

Para Kotonya e Sommerville (1998) as principais atribuições da gerência de requisitos são: gerenciar mudanças nos requisitos definidos; gerenciar as relações entre os requisitos; gerenciar as dependências entre o documento de requisitos e outros documentos produzidos ao longo do processo.

Devido a essas mudanças que podem ocorrer ao longo da vida dos sistemas, dentro da gerência de requisitos deve haver uma rastreabilidade dos mesmos. A atividade de rastreabilidade dos requisitos visa controlar a evolução e a agregação dos requisitos ao projeto de software (GENVIGIR, 2009). Esta atividade será detalhada na próxima seção deste trabalho.

### **2.2.2. Rastreabilidade de Requisitos**

A Rastreabilidade de Requisitos é um dos processos que compõem a Gerência de Requisitos (SEI, 2010) e está presente, dentro deste processo, em padrões e modelos de qualidade como a ISO, CMMI e o MPS.BR. Manter a rastreabilidade entre os diversos documentos de um projeto é considerado imprescindível para obtenção de um

produto de software de qualidade (CLELAND-HUANG, 2003; RAMESH E JARKE, 2001; GOTEL E FINKELSTEIN, 1995). No entanto, esta atividade apresenta uma série de dificuldades relativas à sua aplicação. Suas técnicas manuais são custosas, suscetíveis a erro e difíceis de ser mantidas ao longo do tempo, dificuldades apontadas por Grechanik et al. (2007).

#### 2.2.2.1. Definições

Gotel e Finkelstein (1994) definem Rastreabilidade de Requisitos como a habilidade de descrever e seguir o ciclo de vida de um requisito tanto para frente quanto para trás, ou seja, de sua origem, por meio de sua elicitación, para fases subsequentes de implantação e uso. Desta forma, pode-se afirmar, ainda segundo os autores, que a rastreabilidade de requisitos refere-se à habilidade de identificar as origens de uma especificação de requisitos (reuniões, ideias, decisões de clientes e usuários) e relacioná-las aos produtos de trabalho delas derivados (modelos de análise e projeto, código-fonte, modelos de persistência, casos de teste, manuais, entre outros).

Ferreira (2010) define que o objetivo da rastreabilidade é relacionar entre si os elementos produzidos durante o desenvolvimento de software, permitindo entender e seguir a origem dos requisitos pelos refinamentos realizados durante o ciclo de desenvolvimento. O autor ainda apresenta diversos benefícios que podem ser percebidos pelo uso da rastreabilidade, como: maior facilidade de entender o código de um software e, por isso, ter maior facilidade para analisar o impacto de mudanças e efetivamente alterá-lo; maior visibilidade das regras de negócio e do andamento do projeto; dentre outras que serão abordadas mais adiante.

A especificação da norma IEEE std 830 considera que uma boa especificação de requisitos deve ser rastreável. O próprio instituto define rastreabilidade como o grau em que o relacionamento pode ser estabelecido entre dois ou mais produtos de desenvolvimento de software, especialmente os que tenham uma relação de predecessor e sucessor ou de mestre e subordinado; por exemplo, o grau em que requisitos e projeto (design) de um determinado componente de software combinam (IEEE, 1990 apud SOFTEX, 2013).

Para o SEI (2010) quando os requisitos são bem gerenciados, a rastreabilidade pode ser estabelecida, desde um requisito fonte, passando por todos os níveis de decomposição do produto até seus requisitos de mais baixo nível e destes até o seu



requisito fonte. Tal rastreabilidade é dita bidirecional e auxilia a determinar se todos os requisitos fonte foram completamente tratados e se todos os requisitos de mais baixo nível podem ser rastreados para uma fonte válida.

#### 2.2.2.2. Benefícios

A Rastreabilidade de Requisitos apresenta diversas técnicas para sua execução, e quando são utilizadas de forma correta, alcançam diversos benefícios que envolvem também direta e indiretamente diversos profissionais do projeto de desenvolvimento de software, assim como usuários finais.

Quanto aos benefícios que a rastreabilidade provê a empresas desenvolvedoras/mantenedoras de software pode-se elencar uma série deles disponíveis na literatura, tais como:

- Melhor compreensão do software, já que segundo Antoniol et al. (1999), rastreando documentos de requisitos, análise e projeto de um software para o código-fonte pode-se auxiliar desenvolvedores na compreensão do software;
- Análise de riscos e impacto mais eficiente quanto às alterações realizadas no processo de desenvolvimento de software, já que os requisitos de software frequentemente são alterados durante e após seu desenvolvimento, tendo em vista que a rastreabilidade pode oferecer informações sobre o impacto de mudanças, baseando decisões sobre a realização da mudança e estimativas (ANTONIOI et al., 1999); (EGYED; GRUENBACHER, 2002); (KELLEHER; SIMONSSON, 2006);
- Facilidade na manutenção, já que ao realizar a devida manutenção no produto de software, possíveis impactos serão mais facilmente identificados e tratados. Identificação de componentes ao reuso, já que é possível acompanhar todo o desenvolvimento de um requisito desde sua especificação até sua definição como parte do produto de software e sua ligação com os demais requisitos. Quando a documentação de um software é rastreada para seu código-fonte, é proporcionado um importante auxílio para a localização de componentes candidatos a reuso;
- Auxílio na Verificação e validação, segundo Kannenberg e Saiedian (2009), é nas etapas de verificação e validação que a rastreabilidade demonstra seus

maiores benefícios, pois facilita a identificação de onde os requisitos foram documentados, implementados e testados;

- Maior Visibilidade, a rastreabilidade oferece um importante recurso para que a origem de cada requisito seja corretamente entendida e propagada para os demais produtos de trabalho, indicando a importância desse requisito, como foi implementado e testado. Com esse avanço, ocorre uma melhora na comunicação entre os clientes e a equipe técnica, reduzindo os problemas decorrentes de uma comunicação deficiente (KANNENBERG e SAIEDIAN, 2009);
- Auditoria constante, que em um projeto de desenvolvimento de software tem como objetivo garantir que um conjunto de produtos de trabalho está compatível com critérios específicos, por exemplo, requisitos (SEI 2010). Dentre as atividades realizadas, pode ser avaliado se os requisitos foram devidamente implementados e validados, garantindo que o cliente receberá o produto que solicitou. Também pode ser avaliado se existem casos de teste adequados aos requisitos, garantindo que o produto terá a qualidade esperada. Com ligações de rastreabilidade que demonstrem o cumprimento dos requisitos e testes, pode-se comprovar que os requisitos estão validados e foram testados, chegando a uma melhor avaliação na auditoria (WATKINS; NEAL, 1994);
- Redução de custos de correção de defeitos, visto que a rastreabilidade auxilia no entendimento e visibilidade dos requisitos. Com isso, inconsistências são encontradas e podem ser resolvidas antes de provocarem maiores problemas na documentação, na implementação e nos testes (WATKINS; NEAL, 1994). Ela permite que possíveis erros sejam detectados de maneira antecipada, e corrigidos em fases ou iterações iniciais (LINDVAL; SANDAHL, 1996). Isto gera uma economia de recursos, pois os custos de manutenção são maiores nas fases mais avançadas do processo (LEE; TEPFENHART, 2001);
- Aumento da reutilização a partir da identificação de componentes utilizando as características descritas nos requisitos e os links entre os mesmos;
- Testes contínuos sobre os requisitos durante todo o ciclo de desenvolvimento (CLELAND-HUANG, 2006).

### 2.2.2.3. Dificuldades

Apesar de a rastreabilidade oferecer os benefícios citados na seção anterior e ser uma prática indicada por padrões como o CMMI e o MPS.BR, nem sempre ela é tida como um processo obrigatório para a produção de um software, muitas vezes tratada como uma atividade de suporte, que influencia na qualidade do produto final. Com a pressão pelo cumprimento de prazos, a atividade de manutenção de ligações de rastreabilidade acaba sendo deixada de lado (GOTEL e FINKELSTEIN, 1994). Diversos autores afirmam que a rastreabilidade é complexa de implantar e cara de ser mantida, já que abrange o controle de um grande número de associações entre artefatos, de diversos tipos de informações e de várias partes envolvidas (CLELAND-HUANG, 2003); (RAMESH E JARKE, 2001); (GOTEL E FINKELSTEIN, 1995). Isso aliado à imaturidade das técnicas e ferramentas de rastreabilidade automática atuais, ao grande esforço e suscetibilidade a erros das técnicas manuais de rastreabilidade e à falta de treinamento e conscientização dos profissionais envolvidos, torna a rastreabilidade um processo que recebe pouca adesão das organizações.

### 2.2.2.4. Classificação

A classificação por forma de rastreio é feita de acordo com a capacidade de rastrear um requisito até seus refinamentos, podendo ser subdividida em:

- **Rastrear para frente** (Forwards): capacidade de rastrear um requisito de sua origem até seu refinamento;
- **Rastrear para trás** (Backwards): capacidade de rastrear um requisito de seu refinamento até sua origem.

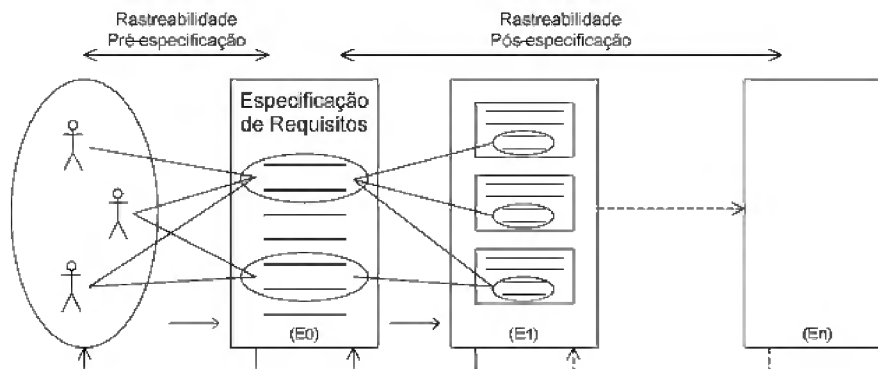
Porém, segundo Genvigir (2009), quanto à forma de rastreio a rastreabilidade deve ser bidirecional, ou seja, permitir o rastreio para frente e para trás. Essa propriedade é necessária a todas as técnicas de rastreabilidade, caso contrário, seria considerado um recurso de utilização limitada.

Em relação ao ciclo de vida do requisito, pode-se classificar a rastreabilidade de duas formas (DAVIS, 1993):

- **Pré-rastreabilidade:** rastreio que vai da especificação dos requisitos, em direção à origem dos mesmos, ou vice-versa;
- **Pós-rastreabilidade:** rastreio que vai da especificação dos requisitos em

direção à implementação, é chamada de pós-rastreabilidade.

Gotel e Finkelstein (1994) definem que a classificação em relação ao ciclo de vida de um requisito está relacionada à especificação de requisitos, como fica claro na Figura 2.4.



**Figura 2.4. Classificação de Rastreabilidade em Pré e Pós-rastreabilidade segundo Gotel e Finkelstein (1994).**

Ao levar em conta a dimensionalidade, pode-se classificar a rastreabilidade de acordo com Genvigir (2009):

- **Rastreabilidade Horizontal:** refere-se aos links criados entre itens que estão no mesmo nível de abstração, tal como o relacionamento entre dois ou mais requisitos;
- **Rastreabilidade Vertical:** refere-se aos links criados entre itens que estão em níveis de abstração distintos, tal como o relacionamento entre casos de uso e casos de teste.

A Figura 2.5 mostra uma visão ampla da classificação de rastreabilidade, segundo Genvigir (2009).

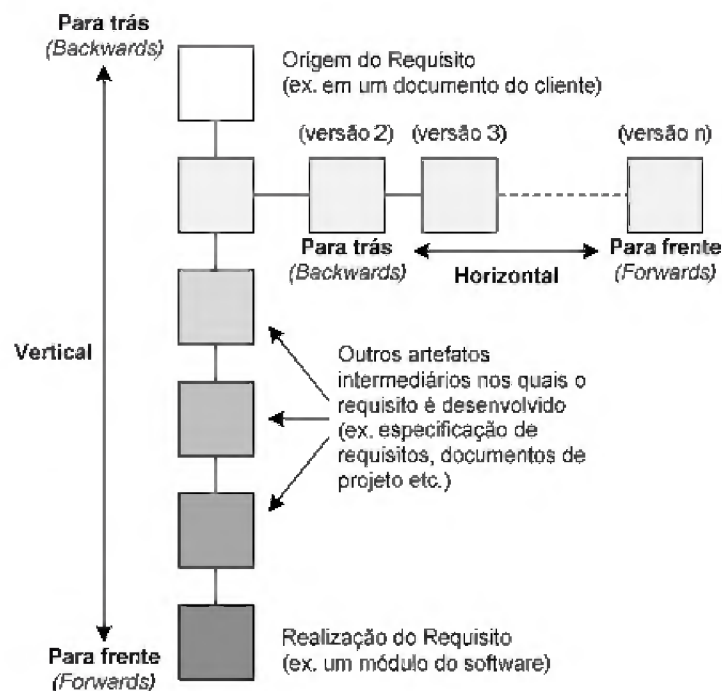


Figura 2.5. Classificação de Rastreabilidade em Horizontal e Vertical por Genvigir (2009).

Para De Lucia et al. (2008), a rastreabilidade vertical fornece apenas uma visão limitada sobre os artefatos afetados por uma mudança. Sistemas complexos necessitam de modelos com vários níveis de abstração, como por exemplo, o código-fonte. Desta forma, muitas abordagens acabam estendendo a rastreabilidade vertical com a horizontal, permitindo gerenciar dependências entre requisitos e artefatos de design, entre requisitos e o código-fonte, entre requisitos e casos de teste, e entre artefatos de design e o código-fonte.

### 2.3. Modelos de Qualidade

Diversos métodos, padrões e boas práticas foram criados pela comunidade de Engenharia de Software no que tange a qualidade no desenvolvimento de software, e acredita-se que o uso destes possa ajudar as empresas a melhorar seu desempenho quanto a custo, prazo, qualidade, entre outros. Porém, Nogueira (2006) constatou que normalmente as organizações só implementam tais práticas quando são exigidas em avaliações de processos. Dentre os padrões que oferecem as boas práticas para o processo de gerência de requisitos, onde se encontra a rastreabilidade de requisitos, pode-se citar: o CMMI-DEV (Capability Maturity Model Integration for Development)

(SEI, 2010), a norma ISO/IEC 12207 (ABNT, 2009), e o MR-MPS-SW (Modelo de Referência do MPS.BR para Software) (SOFTEX, 2012). Estes modelos, normas e guias de boas práticas são apresentados nesta seção, situando a rastreabilidade de requisitos em seu contexto.

Nas subseções a seguir, a rastreabilidade de requisitos será descrita como um componente de cada um dos padrões citados, para definir a base de conhecimentos para os capítulos seguintes.

### **2.3.1.ISO/IEC 12207**

A ISO/IEC 12207 (ABNT, 2009) é uma norma de referência, e não de certificação, que tem como objetivo definir o processo de ciclo de vida de software. Ela tenta resolver o problema da proliferação de normas, procedimentos e métodos de Engenharia de Software que surgem atualmente, pelo estabelecimento de uma estrutura comum que possa ser utilizada para estabelecer uma linguagem comum nos processos de software (ABNT, 2009). Nesse contexto, ela fornece um arcabouço único de conceitos de Engenharia de Software para que os profissionais da área possam se comunicar por meio de termos padronizados. Pode ser aplicada em: aquisição de sistemas e produtos de software e de serviços; fornecimento, desenvolvimento, operação, manutenção e desativação de produtos de software (ABNT, 2009).

A ISO/IEC 12207 não exige a utilização de quaisquer métodos, ferramentas, treinamentos, métricas ou tecnologias. Isto possibilita que a norma seja utilizada mundialmente e possa acompanhar a evolução da Engenharia de Software nas diversas culturas organizacionais. A norma pode ser utilizada com qualquer modelo de ciclo de vida, método ou técnica de Engenharia de Software e linguagem de programação. Dada a sua relevância, esta característica foi incorporada a outros modelos e normas relativos à Engenharia de Software como o MR-MPS-SW (SOFTEX, 2012) e o CMMI-DEV (SEI, 2010).

Os processos da ISO/IEC 12207 são agrupados por afinidade em dois grupos: (i) Processos Contextuais de Sistema; (ii) Processos Específicos de Software. Os do primeiro grupo fornecem um contexto de sistema para lidar com produto ou serviço de software ou um sistema de software independente. Em contrapartida, os do segundo apresentam processos que podem ser utilizados na implementação de um produto ou serviço de software que seja elemento de um sistema maior. Com relação a estes



grupos, o subprocesso de análise de requisito de software encontra-se nos processos específicos de software como pode ser visto na Figura 2.6.

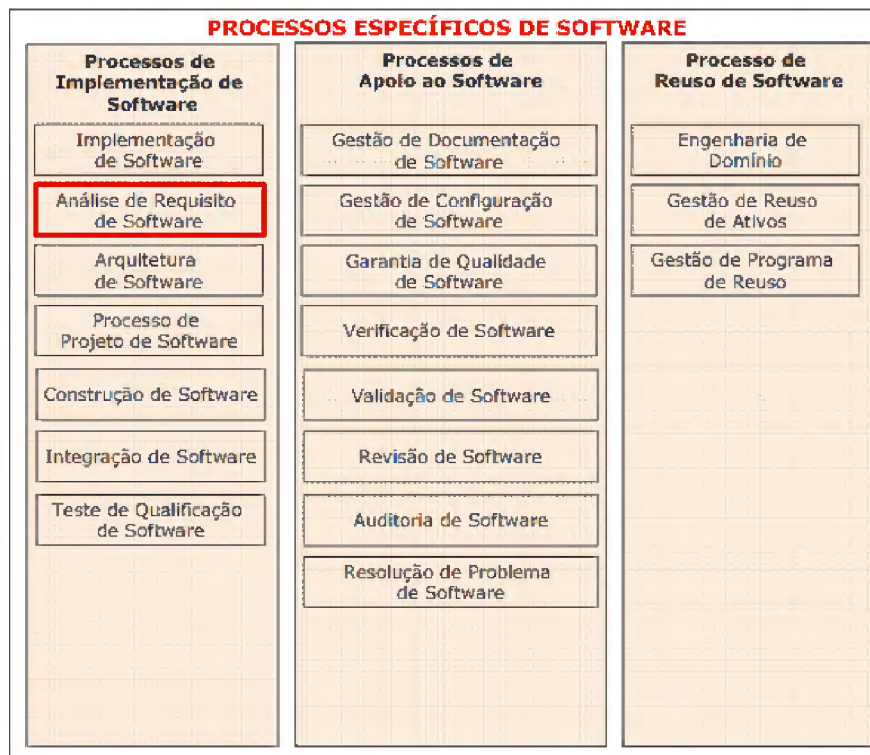


Figura 2.6. Processos Específicos de software na ISO/IEC 12207 (ABNT, 2009).

### 2.3.1.1. Rastreabilidade de Requisitos e a Norma ISO/IEC 12207

O Processo de Análise de Requisitos de Software apresentado na norma ISO/IEC 12207 tem como propósito “Estabelecer os requisitos dos elementos de software do sistema.” (ABNT, 2009).

Este processo está subdividido em duas atividades: (I) Definição dos Requisitos; e (II) Análise de corretude, completitude, testabilidade, rastreabilidade (consistência) e prioridade. Estas atividades são constituídas por um identificador e por uma descrição, conforme consta no guia da ISO/IEC 12207 (ABNT, 2009). Vale ressaltar que o identificador das atividades é composto por 5 números, por exemplo, 6.2.4.3.1, ao passo que o identificador das tarefas é composto por seis números:

- 7.1.2.3.1 Análise de requisitos de software;
  - 7.1.2.3.1.2 O implementador deve avaliar os requisitos de software considerando os critérios listados abaixo:
    - a) Rastreabilidade entre os requisitos do sistema e projeto do sistema;

- b) Consistência externa com os requisitos do sistema;
- c) Consistência interna;
- d) Capacidade de teste;
- e) Viabilidade do projeto de software;
- f) Viabilidade da operação e manutenção.

### 2.3.2. O Modelo CMMI-DEV

O Capability Maturity Model Integration (CMMI) é um modelo de qualidade de processo de software concebido pelo Software Engineering Institute (SEI), cuja principal meta é ajudar a integrar funções organizacionais tradicionalmente separadas, definir objetivos e prioridades na melhoria de processo, prover orientação para processos de qualidade e prover um ponto de referência para avaliação de processos (SEI, 2010).

O CMMI é composto por três modelos (SEI, 2010):

- CMMI for Development (CMMI-DEV) publicada em agosto de 2006. Foca no processo de desenvolvimento de produtos e serviços;
- CMMI for Acquisition (CMMI-ACQ) publicada em novembro de 2007. Foca nos processos de aquisição e terceirização de bens e serviços;
- CMMI for Services (CMMI-SVC) publicada em fevereiro de 2009. Foca nos processos de empresas prestadoras de serviços.

O CMMI-DEV possui duas representações: contínua e por estágios. Que permitem à organização utilizar diferentes caminhos para a melhoria de acordo com seu interesse.

A melhoria contínua possibilita à organização utilizar a ordem de melhoria que melhor atende aos objetivos de negócio da empresa. Cada processo pode ser caracterizado na representação contínua nos seguintes níveis de capacidade (SEI, 2010):

- Nível 0 ou Incompleto (caótico);
- Nível 1 ou Executado: o processo é executado de modo a completar o trabalho necessário para produzir o trabalho necessário;
- Nível 2 ou Gerenciado: planejar a execução do processo e analisar a aderência entre o planejado e o executado;



- **Nível 3 ou Definido:** o processo é construído sobre as diretrizes do processo existente, e é mantido uma descrição do processo;

A representação contínua é indicada quando a empresa deseja tornar apenas alguns processos mais maduros, quando já utiliza algum modelo de maturidade contínua ou quando não pretende usar a maturidade alcançada como modelo de comparação com outras empresas.

A representação por estágios, por sua vez, disponibiliza uma sequência pré-determinada de processos para melhoria baseada em níveis de maturidade. A vantagem de utilizá-las reside no fato de que cada nível serve de base para o próximo, como se pode ver na Figura 2.7.

Nível		Capacidade
<b>5</b>	<b>Melhoria Contínua de Processo</b>	Gestão de Processo Organizacional Análise Causal e Resolução
<b>4</b>	<b>Gerenciamento Quantitativo</b>	Desempenho de Processo Organizacional Gerenciamento Quantitativo de Projeto
<b>3</b>	<b>Padronização de Processo</b>	Desenvolvimento de Requisitos Solução Técnica Integração de Produto Verificação Validação Foco de Processo Organizacional Definição de Processo Organizacional Treinamento Organizacional Gerenciamento Integrado de Projeto Gerenciamento de Riscos Análise de Decisão e Resolução
<b>2</b>	<b>Gerenciamento Básico de Projeto</b>	Gerenciamento de Requisitos Planejamento de Projeto Acompanhamento e Controle de Projeto Gerenciamento de Acordo com Fornecedor Medição e Análise Garantia da Qualidade de Processo e Produto Gerência de Configuração
<b>1</b>	<b>Esforço Heróico</b>	

**Figura 2.7. Níveis de Maturidade e Processos Correlatos (adaptado de SEI, 2010).**

Nesta representação a maturidade é medida por um conjunto de áreas de processos. Assim, por exemplo, é necessário que todas as áreas de processos atinjam nível de capacidade dois para que a empresa seja certificada com nível dois.

### 2.3.2.1. Rastreabilidade de Requisitos e o CMMI-DEV

O CMMI-DEV possui uma área de processo relacionada à Gerência de Requisitos que está no nível 2 de maturidade. Esta área de processo tem como objetivo “fornecer subsídios para gerenciar os requisitos dos produtos e componentes de produto do projeto e identificar inconsistências entre esses requisitos e os planos e produtos de trabalho do projeto.” (SEI, 2010). As recomendações do CMMI-DEV para esta área de processo são (adaptado de SEI, 2010):

- SG 1: Gerenciar Requisitos:
  - SP 1.1 Obter Entendimento dos Requisitos;
  - SP 1.2 Obter Comprometimento com os Requisitos;
  - SP 1.3 Gerenciar Mudanças nos Requisitos;
  - SP 1.4 Manter Rastreabilidade Bidirecional dos Requisitos;
  - SP 1.5 Identificar Inconsistências entre Produtos de Trabalho, Planos de Projeto e Requisitos.

Dentre as recomendações para a Gerência de Requisitos, no contexto deste trabalho serão levadas em conta as que tratam direta ou indiretamente de rastreabilidade de requisitos que seriam a SP 1.3, SP 1.4, SP 1.5.

### 2.3.3. O Modelo MR-MPS-SW

Este modelo faz parte do programa denominado Melhoria do Processo de Software Brasileiro (MPS.BR), coordenado pela SOFTEX - Associação para Promoção da Excelência do Software Brasileiro. Tem como principal meta a de “definir e aprimorar um modelo de melhoria e avaliação de processo de software, visando preferencialmente às micros, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software” (SOFTEX, 2012).

O MPS.BR objetiva ser adequado ao perfil de empresas com diferentes tamanhos e características, públicas e privadas, embora com especial atenção às micro, pequenas e médias empresas. Também se espera que o modelo MPS seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda a competência existente nos padrões e modelos de melhoria

de processo já disponíveis (SOFTEX, 2012). Dessa forma, ele tem como base os requisitos de processos definidos nos modelos de melhoria do processo e atende a necessidade de implantar os princípios de Engenharia de Software de forma adequada ao contexto das empresas. O modelo de qualidade tem o intuito de ser aderente às principais abordagens internacionais para definição, avaliação e melhoria dos processos de software.

Para atender a sua principal meta, o MPS.BR estrutura-se em dois modelos de processos de software, um modelo de avaliação de processos e um modelo de negócio, descritos respectivamente pelos componentes: Modelo de Referência do MPS.BR para Software (MR-MPS-SW); Modelo de Referência do MPS.BR para Serviços (MR-MPS-SV); Método de Avaliação (MA-MPS); e Modelo de Negócio (MN-MPS). Os modelos e o método propostos estão baseados nas normas e nos modelos (SOFTEX, 2012):

- NBR ISO/IEC 12207 – Processo de Ciclo de Vida de Software;
- ISO/IEC 12207, emendas 1 e 2 da norma internacional;
- ISO/IEC 15504;
- ISO/IEC 20000;
- CMMI-DEV;
- CMMI-SVC.

A Figura 2.8 ilustra os componentes do programa MPS.BR e os documentos relacionados.

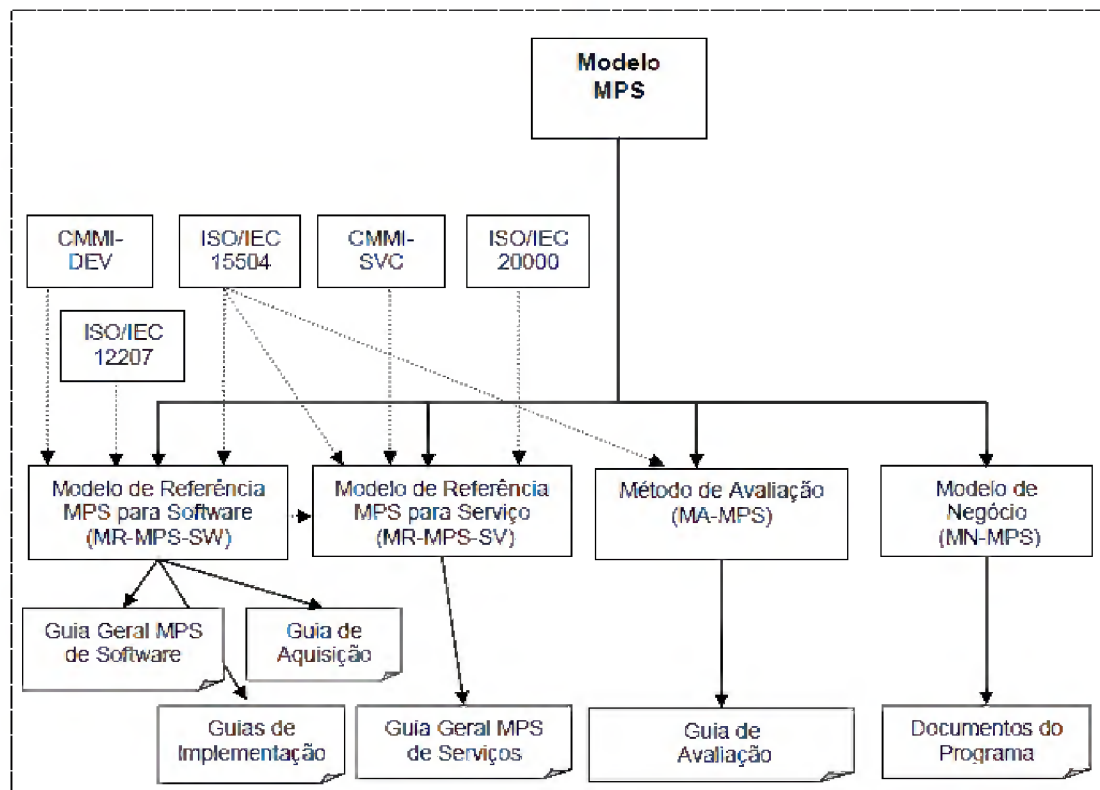


Figura 2.8 - Componentes do MPS.BR (SOFTEX, 2012)

Este modelo de qualidade, portanto, não se propõe a implementar novas diretrizes para a Engenharia de Software. Sua proposta é implementar paradigmas de avaliação diferentes, adaptados à realidade nacional.

O Modelo de Referência MR-MPS-SW contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o mesmo, que por sua vez atende aos requisitos dos modelos de referência de processo da Norma Internacional ISO/IEC 12207 (SOFTEX, 2012).

O Modelo de Referência do MPS.BR para Serviços, ou MR-MPS-SV, tem como referências o Modelo de Referência MR-MPS-SW (SOFTEX, 2012), a Norma Internacional ISO/IEC 20000, a Norma Internacional ISO/IEC 15504 e o modelo CMMI-SVC. Tem como objetivo o detalhamento da definição dos níveis de maturidade, seus processos e capacidade, além dos resultados esperados referentes a serviços.

O Modelo de Avaliação MA-MPS descreve os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA). O processo e o método de avaliação MA-MPS estão em conformidade com a Norma Internacional ISO/IEC 15504-2 (SOFTEX, 2012).

O Modelo de Negócio MN-MPS (SOFTEX, 2012) descreve regras de negócio

para: implementação do MR-MPS-SW e MR-MPS-SV pelas Instituições Implementadoras (II); avaliação seguindo o MA-MPS pelas Instituições Avaliadoras (IA); organização de grupos de empresas pelas Instituições Organizadoras de Grupos de Empresas (IOGE) para implementação do MR-MPS-SW e MR-MPS-SV e avaliação MA-MPS; certificação de Consultores de Aquisição (CA); e programas anuais de treinamento do MPS.BR por meio de cursos, provas e workshops.

Os documentos de apoio que aparecem na Figura 2.8 compõem a base técnica do MPS.BR, são descritos abaixo. Vale ressaltar que os guias de implementação ainda não foram atualizados e se referem apenas ao MR-MPS-SW (SOFTEX 2012):

- Guia Geral MPS de Software:2012;
- Guia Geral MPS de Serviços:2012;
- Guia de Avaliação:2012;
- Guia de Aquisição:2011;
- Guia de Implementação – Parte 1: Fundamentação para Implementação do Nível G do MR-MPS:2011;
- Guia de Implementação – Parte 2: Fundamentação para Implementação do Nível F do MR-MPS:2011;
- Guia de Implementação – Parte 3: Fundamentação para Implementação do Nível E do MR-MPS:2011;
- Guia de Implementação – Parte 4: Fundamentação para Implementação do Nível D do MR-MPS:2011;
- Guia de Implementação – Parte 5: Fundamentação para Implementação do Nível C do MR-MPS:2011;
- Guia de Implementação – Parte 6: Fundamentação para Implementação do Nível B do MR-MPS:2011;
- Guia de Implementação – Parte 7: Fundamentação para Implementação do Nível A do MR-MPS:2011;
- Guia de Implementação – Parte 8: Implementação do MR-MPS:2011 (Níveis G a A) em organizações que adquirem software;

- Guia de Implementação – Parte 9: Implementação do MR-MPS:2011 (Níveis G a A) em organizações do tipo Fábrica de Software;
- Guia de Implementação – Parte 10: Implementação do MR-MPS:2011 (Níveis G a A) em organizações do tipo Fábrica de Teste.
- Guia de Implementação – Parte 11: Implementação e Avaliação do MR-MPS-SW:2012 em Conjunto com o CMMI-DEV v1.3.

O MR-MPS-SW, assim como o CMMI-DEV estagiado, é dividido em níveis de maturidade, os quais estabelecem patamares de evolução de processos. Estes patamares caracterizam os estágios de melhoria da implementação de processos na organização. “O nível de maturidade em que se encontra uma organização permite prever o seu desempenho futuro ao executar um ou mais processos” (SOFTEX, 2012). O MR-MPS-SW tem apenas o equivalente à avaliação por estágios do CMMI-DEV, não apresentando em seu modelo de referência uma abordagem semelhante à avaliação contínua.

Existem sete níveis de maturidade no MR-MPS-SW, cada um contém processos e atributos de processo (SOFTEX, 2012), como mostra o Quadro 2.1.

Os sete níveis de maturidade e suas denominações são em ordem crescente de maturidade: nível G – Parcialmente Gerenciado; nível F – Gerenciado; nível E – Parcialmente Definido; nível D – Largamente Definido; nível C – Definido; nível B – Gerenciado Quantitativamente; nível A – Em Otimização. Cada nível de maturidade é composto pelos seus próprios processos em adição aos de níveis de maturidade inferiores. Estes níveis possuem a relação de equivalência com o CMMI-DEV apresentada na Figura 2.7. Vale ressaltar que o MR-MPS-SW não apresenta classificação equivalente ao nível 1 do CMMI-DEV, referente ao caos e esforço heroico predominantes.

#### **2.3.3.1. Rastreabilidade de Requisitos e o MR-MPS-SW**

O processo de Gerência de Requisitos encontra-se no Nível G, ou Parcialmente Gerenciado, do MR-MPS-SW. O nível G do MPS.BR é o primeiro nível de maturidade do MR-MPS. Sua implementação deve ser executada com cautela por estabelecer o início dos trabalhos em implantação de melhoria dos processos de software na organização. Ao final da implantação desse nível a organização deve ser capaz de



gerenciar parcialmente seus projetos de desenvolvimento de software.

**Quadro 2.1. Níveis de maturidade do MR-MPS-SW (SOFTEX, 2012)**

Nível	Processos	Atributos de Processo
A		AP 1.1, AP 2.1, AP 2.2, AP 3.1, AP 3.2, AP 4.1, AP 4.2, AP 5.1 e AP 5.2
B	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2, AP 4.1 e AP 4.2
C	Gerência de Riscos – GRI	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Desenvolvimento para Reutilização – DRU	
	Gerência de Decisões – GDE	
D	Verificação – VER	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Validação – VAL	
	Projeto e Construção do Produto – PCP	
	Integração do Produto – ITP	
	Desenvolvimento de Requisitos – DRE	
E	Gerência de Projetos – GPR (evolução)	AP 1.1, AP 2.1, AP 2.2, AP 3.1 e AP 3.2
	Gerência de Reutilização – GRU	
	Gerência de Recursos Humanos – GRH	
	Definição do Processo Organizacional – DFP	
	Avaliação e Melhoria do Processo Organizacional – AMP	
F	Medição – MED	AP 1.1, AP 2.1 e AP 2.2
	Garantia da Qualidade – GQA	
	Gerência de Portfólio de Projetos – GPP	
	Gerência de Configuração – GCO	
	Aquisição – AQU	
G	Gerência de Requisitos – GRE	AP 1.1 e AP 2.1
	Gerência de Projetos – GPR	

O processo de Gerência de Requisitos tem como propósito de “gerenciar os requisitos do produto e dos componentes do produto e identificar inconsistências entre os requisitos, os planos do projeto e os produtos de trabalho do projeto” (SOFTEX, 2012).



**Figura 2.9. Equivalência entre Níveis do MR-MPS-SW e níveis do CMMI-DEV (adaptado de SOFTEX, 2012).**

Este processo conta com cinco resultados esperados. São eles (SOFTEX, 2012):

- GRE 1: Os requisitos são entendidos, avaliados e aceitos junto aos fornecedores de requisitos, utilizando critérios objetivos;
- GRE 2: Obtenção do comprometimento da equipe técnica com os requisitos aprovados;
- GRE 3: A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho é estabelecida e mantida;
- GRE 4: Revisões em planos e produtos de trabalho do projeto são realizadas visando identificar e corrigir inconsistências em relação aos requisitos;
- GRE 5: Mudanças nos requisitos são gerenciadas ao longo do projeto.

Dentre os resultados esperados os que estão incluídos no escopo deste trabalho são os que tratam direta ou indiretamente de rastreabilidade de requisitos que seriam os resultados esperados GRE3, GRE4 e GRE5.

## 2.4. Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados a esta dissertação. Foram



pesquisados trabalhos que tratassem de Revisões Sistemáticas da Literatura realizadas sobre a Rastreabilidade de Requisitos. Vale ressaltar que os trabalhos aqui apresentados foram selecionados através de uma pesquisa ad hoc na máquina de busca do Google Acadêmico e em diversas bases de dados relevantes para o tema. Sendo encontrados dois trabalhos já realizados intitulados como:

- “Requirements traceability state-of-the-art: A systematic review and industry case study” (TORKAR et al, 2012);
- “A Review of Traceability Research at the Requirements Engineering Conference<sup>RE@21</sup>” (NAIR, 2013);

O primeiro estudo trata de uma Revisão Sistemática e um Estudo de Caso na Indústria sobre o estado da arte da Rastreabilidade de Requisitos no período de 1997 a 2007. As questões de pesquisas da revisão podem ser resumidas na busca pela definição de Rastreabilidade de Requisitos, seus principais desafios, ferramentas e técnicas que pudessem auxiliar em sua aplicação. A pesquisa foi realizada em cinco fontes de busca (IEEE, ACM, Springer Link, Inspec e Compendex). Com base nos critérios de inclusão e exclusão, foram selecionados 52 estudos primários que foram utilizados para responder as questões de pesquisa da revisão.

Nesta revisão realizada pode-se observar que, mesmo sendo um trabalho publicado no ano de 2012, o ano limite de busca foi o ano de 2007, ou seja, há sete anos e devido a isso novas abordagens já podem estar disponíveis na literatura. Outro ponto importante a ser destacado em relação ao trabalho é que Tokar et al. (2012) não apresentam a qualidade dos estudos primários selecionados, apenas apresentam no protocolo de revisão os critérios definidos de qualidade, mas não relatam as suas aplicações. O trabalho realizado nesta pesquisa se diferencia deste relacionado em relação ao seu objetivo e a sua questão de pesquisa, já que este busca encontrar na literatura não apenas a caracterização de rastreabilidade de requisitos, os principais desafios em sua implementação, principais técnicas e ferramentas disponíveis na literatura, mas sim de uma forma mais ampla quaisquer abordagens que sejam referentes à rastreabilidade de requisitos, como processos, framework, modelos, meta-modelos, entre outros. Esta diferença pode ser evidenciada quanto a grande diferença entre a quantidade de trabalhos selecionados nas duas pesquisas. Quanto aos resultados, este se diferencia no que diz respeito à elaboração do catálogo de técnicas e suas principais utilizações. Outro fator diferencial deste trabalho é quanto à abrangência, já que a busca será realizada em um número maior em fontes de

pesquisas, inclusive fontes de pesquisas nacionais.

Já o segundo estudo trata de uma revisão sobre rastreabilidade no contexto da “Requirements Engineering ConferenceRE@21” realizada em um período de vinte anos, porém no trabalho não é apresentada claramente a data de início e fim desse período em seu planejamento, mas pela disposição dos resultados pode-se supor que o período realizado foi de 1993 a 2013. O principal objetivo para a realização da revisão foi o de verificar como a pesquisa em Rastreabilidade na conferência citada tem contribuído para a área, nesse sentido pode-se resumir como questões de pesquisa dessa revisão, a descoberta de temas sobre rastreabilidades, desafios, ferramentas, artefatos, métodos e por fim os principais autores e instituições da área encontrados na literatura desta conferência.

Vale ressaltar que o estudo não define a revisão realizada como uma revisão sistemática da literatura, porém ele apresenta algumas definições que são utilizadas nesse tipo de estudos. Com base na estratégia de busca definida no trabalho foram encontrados 76 trabalhos, porém houve a exclusão de seis estudos, resultando assim em 70 estudos primários a serem levados em consideração na pesquisa. Não foram apresentados critérios objetivos para inclusão ou exclusão de estudos, o que proporciona um caráter mais subjetivo a esta revisão, algo que não é bem visto em uma revisão sistemática que preza pela eliminação de vieses por parte de seus autores. Outro ponto a ser destacado é que não foi realizada uma avaliação da qualidade nos estudos secundários, além do que, a revisão realizada concentra-se em apenas uma conferência o que pode ser apresentada como uma limitação do trabalho.

Com relação aos dois trabalhos relacionados encontrados, pode-se verificar que os objetivos das revisões sistemáticas realizadas são diferentes deste, já que as anteriores buscavam caracterizar a rastreabilidade de requisitos e o trabalho desenvolvido nesta pesquisa não tem apenas essa finalidade, mas também fazer um estudo detalhado nas técnicas que são relatadas na literatura para apoio à rastreabilidade de requisitos a fim de ajudar na elaboração de um catálogo contendo as mesmas e suas especificações. Quanto aos resultados, este trabalho pretende além de caracterizar o fenômeno, mostrar tendências relacionadas à rastreabilidade de requisitos por meio das abordagens encontradas na literatura.

## **2.5. Considerações Finais**

Este capítulo apresentou os principais conceitos sobre a Rastreabilidade de Requisitos e sua relação com os modelos e normas para a melhoria do processo de software.

A Rastreabilidade de Requisitos é uma atividade que pertence ao processo de Gerência de Requisitos, que por sua vez faz parte da Engenharia de Requisitos que é uma área que possibilita a coleta, o entendimento e o acompanhamento de um requisito em todo o ciclo de vida de desenvolvimento de um software. Ela tem como objetivo seguir um requisito durante todo o seu ciclo de vida, acompanhando em suas transformações e relacionamentos com outros requisitos ou mesmo com outros artefatos. Corroborando com esta ideia, modelos de qualidade para o processo de software têm práticas correlatas à Rastreabilidade de Requisitos por meio de processos de Gerência e Análise de Requisitos. Portanto, pode-se inferir a importância da Rastreabilidade em organizações desenvolvedoras de software.

Diversas pesquisas apontam ferramentas, técnicas, processos e modelos, e com isto é observável a riqueza de opções de pesquisa nesta área. No entanto, estas pesquisas ainda se encontram dispersas e sem uma análise geral na literatura e conseqüentemente sem um instrumento que agrupe essas diversas abordagens. Outro ponto a ser destacado é que o desenvolvimento dessas abordagens de apoio à rastreabilidade tem evoluído com o passar do tempo, saindo de métodos manuais e chegando a métodos totalmente automáticos. A dissertação aqui apresentada pretende contribuir neste sentido para as pesquisas na área.

### **3 UMA REVISÃO SISTEMÁTICA DA LITERATURA PARA RASTREABILIDADE DE REQUISITOS**

Este capítulo apresenta uma Revisão Sistemática da Literatura (RSL) que teve como intuito descobrir abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software. As estratégias, metodologias e resultados preliminares da RSL também são apresentados com base nos seguintes tópicos:

- Engenharia de Software Baseada em Evidência;
  - Revisões Sistemáticas da Literatura (RSL);
- Metodologia de uma RSL;
  - Planejamento;
  - Condução;
  - Apresentação;
- Resultados Preliminares;
  - Planejamento;
  - Condução;
- Considerações Finais do Capítulo.

Vale ressaltar que os estudos identificados no contexto desta RSL serão utilizados para fundamentar o Catálogo de abordagens de apoio à rastreabilidade de requisitos, o qual será apresentado e discutido no Capítulo 4. Outro ponto a ser destacado é fato de Ferreira (2014) descrever resultados parciais deste capítulo em seu trabalho de conclusão de curso. O autor participou como apoio da revisão sistemática da literatura.

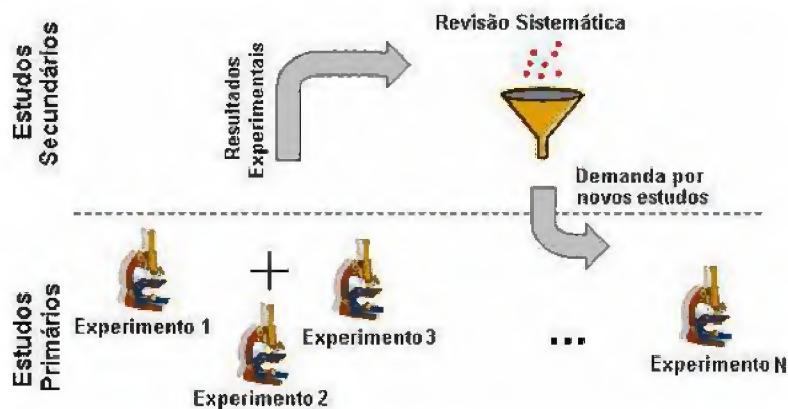
#### **3.1. Engenharia de Software Baseada em Evidências**

Cochrane (2003) apud Mafra e Travassos (2006) já afirmava que a importância da aplicação de uma metodologia baseada em evidência na pesquisa científica pode ser ilustrada pela experiência vivenciada na Medicina, que durante muito tempo, esteve repleta de revisões que não utilizavam métodos para identificar, avaliar e sintetizar a informação existente na literatura. Além disso, chamou atenção que uma provável consequência dessa falta de rigor pôde ser observada no final dos anos 80, onde estudos apontaram que, de um lado, a falha em organizar a pesquisa médica em revisões sistemáticas podia custar vidas. Desde então, o reconhecimento da necessidade da condução de revisões de forma sistemática e formal em Medicina tem crescido rapidamente. Este fato pode ser comprovado pela grande quantidade de revisões formais publicadas a cada ano na área médica (NHSCRD, 2003).

Segundo Mafra e Travassos (2006) a experimentação em Engenharia de Software surgiu por volta da década de 1980 e desde então diversos trabalhos têm sido publicados ao longo dos anos a este respeito. A experimentação permite avaliar a atividade humana de modo sistemático, disciplinado, computável e controlado (TRAVASSOS; GUROV; AMARAL, 2002). Kitchenham et al. (2004) foram os primeiros a realizar um trabalho que estabelecesse um paralelo entre Medicina e Engenharia de Software, no que diz respeito a uma abordagem baseada em evidência. Segundo os autores, a Engenharia de Software Baseada em Evidência deve prover meios pelos quais melhores evidências provenientes da pesquisa possam ser integradas com experiência prática e valores humanos no processo de tomada de decisão, considerando o desenvolvimento e a manutenção do software.

Com o objetivo de atingir um nível adequado de evidência a respeito da caracterização de uma determinada tecnologia em uso, a Engenharia de Software Baseada em Evidência, segundo Mafra e Travassos (2006), deve fazer uso basicamente de dois tipos de estudos: estudos primários e estudos secundários. Os estudos primários são definidos como aqueles que visam caracterizar uma tecnologia em uso dentro de um contexto específico. Nessa categoria encontram-se os estudos experimentais, entre os quais experimentos, estudos de caso e surveys. Já os estudos secundários são aqueles que visam identificar, avaliar e interpretar todos os resultados relevantes a um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa.

Mafrá, Barcelos e Travassos (2006) apresentam que resultados obtidos por diversos estudos primários atuam como fonte de informação para estudos secundários, essa relação é mostrada na Figura 3.1.



**Figura 3.1** Relação entre Estudos Primários e Estudos Secundários (MAFRA; BARCELOS e TRAVASSOS, 2006).

Ainda em comparativo com a medicina e com dados baseados na pesquisa de Monteiro (2010), pode-se afirmar que as práticas médicas mudaram drasticamente na última década, pois, estudos mostraram que a não realização de revisões sistemáticas poderia custar vidas. Costa (2010) discute que esta afirmação pode ser válida também para o contexto da Engenharia de Software, devido ao aumento da importância de softwares em diversas áreas, já que os mesmos têm sido utilizados em dispositivos médicos, sistemas de controle de voos, construção civil, entre outros.

A essência do paradigma baseado em evidência é coletar e analisar sistematicamente todos os dados disponíveis sobre determinado fenômeno para obter uma perspectiva mais completa e mais ampla do que se pode captar por meio de um estudo individual (COSTA, 2010).

Um dos principais métodos da Engenharia de Software Baseada em Evidências é a Revisão Sistemática da Literatura, ou RSL, classificada como um estudo secundário, já que dependem dos estudos primários utilizados para revelar evidências e construir conhecimento (KITCHENHAM et al. 2004; BIOLCHINI et al. 2005).

A precisão e a confiabilidade proporcionadas pela condução de estudos secundários contribuem para a melhoria e para o direcionamento de novos tópicos de pesquisa, a serem investigados por estudos primários, em um ciclo iterativo. Ou seja, é válido ressaltar que estudos secundários não podem ser considerados uma abordagem



alternativa para a produção primária de evidências, representada pelos estudos primários (BIOLCHINI et al. 2005).

### **3.2. Revisão Sistemática da Literatura**

Segundo Kitchenham (2007), uma Revisão Sistemática da Literatura é uma forma de avaliar e interpretar todas as pesquisas disponíveis, referentes a uma questão de investigação particular, área temática ou fenômeno de interesse. Podendo definir, assim, esta revisão, como uma revisão abrangente e não tendenciosa. Ele afirma, ainda, que revisões sistemáticas têm por objetivo apresentar uma avaliação justa de um tópico de investigação, usando uma metodologia confiável, rigorosa e auditável.

Pode-se afirmar que a principal meta de uma revisão sistemática da literatura consiste em realizar pesquisa exaustiva na literatura, em busca de evidências que possam apoiar uma determinada hipótese, ou simplesmente a busca por conhecimento aprofundado acerca de certo fenômeno de interesse. Para tal, a revisão sistemática faz uso de estudos previamente publicados e validados relacionados ao tópico de interesse: os estudos primários, estudos de natureza experimental que envolvem hipóteses e resultados obtidos com pesquisas e experimentação, a partir de diferentes métodos, como surveys, estudo de caso e experimentos. (MAFRA; TRAVASSOS, 2006).

Mafra e Travassos (2006) também Na Engenharia de Software, os estudos experimentais consistem em uma importante maneira de se obter maiores informações a respeito de novas tecnologias, metodologias e boas práticas quanto ao desenvolvimento de software

Mulrow (1994) define a RSL como uma técnica científica objetiva, eficiente e reprodutível, que permite extrapolar achados de estudos independentes, avaliar a consistência de cada um deles e explicar as possíveis inconsistências e conflitos. Com base nisso, uma RSL reúne, de forma organizada, o resultado de vários estudos realizados em uma determinada área do conhecimento.

Clarke e Oxman (2000) apresentam como objetivo de uma RSL o de responder uma pergunta claramente formulada utilizando métodos sistemáticos e explícitos para identificar, selecionar e avaliar criticamente pesquisas relevantes, e coletar e analisar

dados de estudos incluídos na revisão. Sendo assim, é possível considerar uma RSL como uma revisão transparente, já que a mesma deve apresentar toda sua estratégia de pesquisa, restrições e critérios de avaliação

A revisão sistemática possui como característica: a abrangência, já que engloba todos ou, pelo menos, a grande maioria dos estudos relevantes à questão de pesquisa; ser não tendenciosa, pois possui um protocolo de revisão, não sendo dirigida por interesses pessoais de seus pesquisadores; e ser passível de replicação, por possuir um protocolo de revisão definido a priori (MAFRA; TRAVASSOS, 2006).

A ciência é uma atividade cooperativa e social e o conhecimento científico é resultado do processo cumulativo dessa cooperação (BIOLCHINI et al., 2005). Nesse sentido, Mafra e Travassos (2006) afirmam que a revisão na literatura é o meio pelo qual o pesquisador pode identificar o conhecimento específico em uma determinada área de forma a planejar sua pesquisa, evitando duplicação de esforços e a repetição de erros passados. Uma revisão sistemática difere-se de uma revisão informal nos seguintes aspectos:

- **Inconsistência dos resultados:** para Biolchini et al. (2005), nas revisões informais a presença de inconsistência é um fator negativo, já que dificulta o entendimento e julgamento do pesquisador. Já nas revisões sistemáticas, essa inconsistência é um fator de incentivo para o pesquisador, já que por meio dessa inconsistência pode ser percebida a falta de robustez para o fenômeno em estudo, o que pode auxiliar no apoio ou não de suas hipóteses;
- **Propósito:** para Mafra e Travassos (2006) uma revisão sistemática não é simplesmente uma revisão da literatura conduzida conforme um planejamento. A revisão de literatura é parte integrante de uma revisão sistemática, ou seja, o objetivo de uma revisão sistemática é maior do que o de uma revisão de literatura informal, tendo em vista que o propósito de uma revisão sistemática é a análise dos dados coletados com o objetivo da geração de evidências para o fenômeno em questão.

No contexto da Engenharia de Software, Mafra e Travassos (2006) afirmam que frequentemente as revisões de literatura são realizadas de maneira informal, sem planejamento e critérios estabelecidos, o que geralmente ocasiona em uma revisão pouco abrangente, não passível de repetição, pouco confiável e dependente de



revisores. Aspectos esses que não são encontrados em uma revisão sistemática, já que a mesma estabelece um processo formal. Vale ressaltar, ainda, nesse aspecto, que uma revisão sistemática não é apenas um agrupamento de estudos e sim uma abordagem metodológica para se realizar pesquisa com finalidades experimentais.

### **3.3. Metodologia de uma Revisão Sistemática da Literatura**

Uma revisão sistemática deve ser conduzida por meio de um protocolo pré-estabelecido para garantir que essa revisão tenha de fato valor científico e possibilidade de repetição, pois caso isso não aconteça, as revisões tornam-se informais e dependentes dos revisores que a conduziram, diminuindo o grau de confiabilidade das mesmas.

A revisão sistemática segue uma metodologia específica e peculiar, apresentada por Kitchenham (2007), em que define: que as revisões sistemáticas começam pela definição de um protocolo de revisão, que especifica a questão de pesquisa a ser tratada e os métodos que serão usados para sua realização; no qual é apresentada uma estratégia de busca definida pelo pesquisador, a qual objetiva identificar o máximo possível de estudos relevantes à questão de pesquisa; onde esta estratégia é documentada nos mínimos detalhes, para que o leitor possa avaliar seu rigor, completude e replicabilidade; além do que, é exigido a definição de critérios explícitos de inclusão e exclusão de estudos primários, de forma a avaliar a necessidade destes estudos para a pesquisa; e por fim é especificada a forma que serão extraídos os dados a serem obtidos a partir de cada estudo primário, incluindo critérios de qualidade para avaliação de cada estudo.

Desta forma, Kitchenham (2007) resume as etapas de uma revisão sistemática em três fases principais: planejamento, condução e apresentação. Essas etapas são explicadas nas subseções seguintes.

#### **3.3.1. Planejamento da revisão**

A fase de planejamento de uma RSL é definida pela definição de seu Protocolo de Revisão, onde são definidos diversos pontos importantes que nortearam todo o processo de revisão, tais como a definição da questão de pesquisa, critérios objetivos de avaliação de estudos, métodos de seleção, avaliação e extração.

A formulação das questões de pesquisa a serem respondidas é a base para uma revisão sistemática, sendo assim, a fase mais importante da etapa de planejamento, já que todos os outros aspectos do processo da revisão dependem delas (DYBA et al., 2007).

Para Kitchenham (2007), antes de realizar uma revisão sistemática, os pesquisadores devem assegurar que a mesma é necessária e o protocolo deve ser capaz de responder a algumas questões:

- Quais são os objetivos desta revisão?
- Que fontes foram pesquisadas para identificar os estudos primários? Houve alguma restrição?
- Quais foram os critérios de inclusão / exclusão e como foram aplicados?
- Que critérios foram utilizados para avaliar a qualidade dos estudos primários?
- Como foram aplicados os critérios de qualidade?
- Como os dados foram extraídos dos estudos primários?
- Como os dados foram sintetizados?
- Quais foram as diferenças entre os estudos pesquisados?
- Como os dados foram combinados?

Travassos (2007), por meio de alguns passos, confirma e acrescenta novas informações sobre o que a fase de planejamento deve contemplar:

- Objetivos da pesquisa devem ser listados;
- Questões de pesquisa formuladas (strings de busca preparadas);
- Métodos que serão utilizados para executar a revisão e analisar os dados obtidos devem ser definidos;
- As fontes e a seleção de estudos devem ser planejadas;
- Um protocolo de revisão deve ser definido, documentado e disponibilizado.

Outro fator importante a ser levado em consideração é que depois de definido, o protocolo deve passar por uma avaliação para garantir a viabilidade da pesquisa. É sugerido que essa avaliação seja realizada por um especialista.

### 3.3.2. Condução da revisão

A etapa de condução da revisão pode ser dividida nas seguintes fases: busca primária, seleção dos estudos primários, avaliação da qualidade dos estudos primários e extração dos dados.

A fase de busca primária é definida pela utilização de strings de busca nas fontes de pesquisa definidas, tendo como resultado um conjunto de estudos primários potenciais para a pesquisa.

A fase de seleção dos estudos primários é onde ocorre a execução do processo de seleção definido no protocolo de revisão, com a aplicação dos critérios de inclusão e exclusão pré-definidos nos estudos retornados da busca primária, tendo como resultado o conjunto de estudos que serão utilizados na pesquisa.

A fase de avaliação da qualidade dos estudos primários ocorre após a seleção, onde estes são avaliados inicialmente com base em critérios de qualidade, previamente estabelecidos. Depois dessa avaliação é atribuída uma nota final para cada estudo e com base em uma tabela com faixas de notas é atribuído um nível de qualidade.

A fase de extração dos dados pode ser realizada por meio de formulários, podendo ser apoiadas por uma ferramenta computacional, para a categorização desses dados obtidos dos estudos primários selecionados.

Para Travassos e Biolchini (2007), o processo para realizar a etapa de condução de uma revisão sistemática é definido pelos seguintes passos:

- Realização das Buscas nas fontes definidas: o processo deve ser transparente, repetível e documentado, assim como as mudanças que ocorrem no processo.
- Seleção dos Estudos Primários com os critérios de inclusão e exclusão definidos.
- Extração dos dados, desde informações gerais dos estudos às respostas para as questões de pesquisa. Formulários são um bom meio para registrar todos os dados necessários e o uso de uma ferramenta computacional pode apoiar a extração e registro dos dados e posterior análise.

- Avaliação da qualidade dos estudos é importante para balancear a importância de diferentes estudos, reduzir o viés (tendência a produzir “resultados tendenciosos” que se separam sistematicamente dos resultados verdadeiros), maximizar a validade interna e externa e guiar recomendações para pesquisas futuras.
- Síntese dos dados é realizada de acordo com as questões de pesquisa, por meio de quadros para realçar as similaridades e diferenças entre estudos. Se dados quantitativos estão disponíveis, pode-se considerar fazer uma meta-análise.

### **3.3.3. Apresentação da revisão**

A etapa de apresentação da revisão consiste na sumarização dos resultados, com base na análise e síntese dos dados realizada durante a etapa de extração.

A sumarização pode ser realizada por meio da escrita de um relatório, em que são apresentados os resultados com informações tabuladas de forma que auxiliem ou respondam as questões de pesquisa.

Vale ressaltar que as etapas apresentadas, não necessariamente, precisam ser realizadas de forma sequencial, algumas delas podem ser realizadas de forma concorrente.

## **3.4. Planejamento da Revisão**

A etapa de planejamento da revisão sistemática da literatura realizada neste trabalho culminou na criação de um protocolo de revisão (Apêndice A).

O protocolo de revisão é o documento em que se encontra o planejamento de uma revisão sistemática. Nele são definidos os objetivos, o escopo, as restrições, os critérios, entre outras especificações para que uma revisão sistemática seja conduzida com sucesso. Nas subseções seguintes são apresentadas essas especificações.

### **3.4.1. Objetivo da Revisão**

Esta revisão sistemática teve o objetivo de identificar abordagens para apoiar a atividade de Rastreabilidade de Requisitos, no contexto de projetos de software, no que

tange a processos, modelos, metodologias, técnicas, ferramentas e afins. Desta forma, têm-se a seguinte estrutura para o objetivo, conforme proposto em Santos (2010):

- **Analisar:** relatos de experiência e publicações científicas através de um estudo baseado em revisão sistemática.
- **Com o propósito de:** identificar abordagens para apoiar atividades de Rastreabilidade de Requisitos.
- **Com relação à:** definição e uso de processos, frameworks, metodologias, ferramentas e demais instrumentos empregados para a implantação e execução das atividades de Rastreabilidade de Requisitos em organizações de desenvolvimento de software.
- **Do ponto de vista:** de pesquisadores e organizações desenvolvedoras/mantenedoras de software.
- **No contexto:** acadêmico e industrial.

### 3.4.2. Questão de Pesquisa Principal

Com base no objetivo de investigação desta revisão foi definida a seguinte questão de pesquisa, a qual guiou esta Revisão Sistemática da Literatura:

- **(Q1) Quais os padrões existentes para apoiar as atividades de Rastreabilidade de Requisitos?**

#### 3.4.2.1. Estrutura da Questão de Pesquisa Principal

A estrutura da questão de pesquisa foi organizada conforme a estrutura Population, Intervention, Context, Outcomes, Comparison (PICOC), recomendada por Kitchenham (2007). Entretanto, apenas os itens Population, Intervention e Outcomes (PIO), que traduzidos para o português são População, Intervenção e Resultados, foram considerados relevantes para a pesquisa. Comparison – Comparação – foi excluído do contexto, pois esta revisão sistemática da literatura não tem como objetivo comparar duas possíveis intervenções. Context – Contexto – não foi considerado no escopo da pesquisa, pois se refere ao contexto no qual a comparação acontece, portanto ao excluir Comparação, foi necessário também excluir o Contexto.

Tal restrição, segundo Santos (2010), caracteriza esta pesquisa como uma Revisão QUASI Sistemática da Literatura. Ainda segundo Kitchenham (2007), esta é a

estrutura recomendada na área médica. Nesse sentido definiu-se a seguinte estrutura para a questão de pesquisa principal:

- **População (P):** Organizações de Software e Projetos de Software.
- **Intervenção (I):** Padrões para apoiar atividades de Rastreabilidade de Requisitos.
- **Resultados (O):** Modelos de processos, técnicas, metodologias, ferramentas e frameworks de Rastreabilidade de Requisitos.

### 3.4.3. Questões de Pesquisa Secundárias

Com base na questão principal e utilizando o conhecimento obtido na realização de uma revisão informal para Rastreabilidade de Requisitos, foi estabelecido um conjunto de questões de pesquisa secundárias. Tais questões têm o objetivo de esclarecer detalhes importantes que a revisão procura identificar para colaborar com o projeto onde está inserida. As questões de pesquisa secundárias são apresentadas a seguir:

- **(Q1A)** Qual o contexto de aplicação da abordagem encontrada?
- **(Q1B)** Quais as formas de rastreabilidade (para frente e para trás) são cobertas pela abordagem?
- **(Q1C)** Quais os tipos de rastreabilidade (vertical e horizontal) são cobertos pela abordagem?
- **(Q1D)** Quais os ativos (papeis, artefatos) envolvidos?
- **(Q1E)** Existem softwares de apoio para a abordagem proposta?
- **(Q1F)** Caso existam softwares de apoio, qual a sua licença de uso?

### 3.4.4. Escopo e Recursos da Pesquisa

Com o objetivo de assegurar a viabilidade da pesquisa, foi definido um escopo para a mesma, que compreende quais máquinas de busca foram consideradas na elaboração deste trabalho, o que pode ser descrito por meio da definição de critérios de seleção de fontes e algumas restrições.

Para a seleção das fontes de pesquisa, foram definidos os seguintes critérios:

- Disponibilidade para consultas web.

- Disponibilidade para busca de artigos através do domínio da UFPA.
- Disponibilidade de artigos na íntegra por meio do domínio da UFPA ou a partir da utilização da engine de busca Google e/ou Google Scholar.
- Disponibilidade de artigos em inglês ou português.
- Presença de mecanismo de busca que faça uso de palavras-chave.
- Relevância da fonte.
- Boa relação entre estudos retornados e estudos selecionáveis.

Como restrições para a pesquisa têm-se:

- A pesquisa não pode incorrer em ônus financeiro aos pesquisadores. Portanto, apenas foram selecionadas as fontes que possibilitam consultas de forma gratuita (também foram consideradas fontes que oferecem consultas de forma gratuita a partir do acesso pelo domínio da UFPA).
- Foram apenas considerados os estudos obtidos por meio das fontes selecionadas e em conformidade com os critérios de inclusão e exclusão.
- A pesquisa esteve restrita aos resultados publicados entre 01 de janeiro de 2003 até dezembro de 2013, contemplando, desta forma, um período de mais de 11 anos, dada a necessidade de identificar abordagens para apoio às atividades de Rastreabilidade de Requisitos mais atuais. Vale ressaltar que esse período pode ser adequado de acordo com as limitações das fontes.

Adicionalmente, para a execução da revisão sistemática, foram utilizados os seguintes recursos:

- Dois pesquisadores da UFPA (um aluno de mestrado em Ciência da Computação e um aluno da graduação de Ciência da Computação).
  - O aluno de mestrado, autor desta dissertação, foi o responsável pela coordenação de esforços da equipe. Atuou, também, nas etapas de seleção, avaliação da qualidade, análise e extração de resultados.
  - O aluno da graduação atuou na etapa de seleção de estudos primários e desenvolveu esforços referentes à organização das tabelas, conforme os dados iam sendo coletados e desenvolvidos pelos demais pesquisadores.



- Acesso às fontes de pesquisa por meio do domínio da Universidade Federal do Pará.
- Validações sobre documentos e procedimentos da realização da revisão sistemática através de reuniões com o coordenador do Projeto SPIDER e orientador deste trabalho.

### 3.4.5. Seleção de Fontes

Com base na relevância das fontes para o tópico de interesse deste trabalho foram elencadas nove possíveis fontes de interesse. Após, levando em consideração os critérios de seleção e nas restrições da pesquisa, foram selecionadas as fontes de pesquisa (Quadro 3.1) nas quais foram realizadas as buscas dos estudos primários, de acordo com disponibilidade de pesquisa e relevância para área de Qualidade de Software.

**Quadro 3.1 – Seleção de Fontes de Pesquisa**

Fonte	Selecionada?	Justificativa
ACM	Sim	-
El Compendex	Sim	-
IEEE Xplore Digital Library	Sim	-
ISI Web of Knowledge	Sim	-
Scopus	Sim	-
Jairo	Não	Máquina de busca ineficiente.
Anais do WAMPS - Anais do Workshop Anual do MPS.BR	Sim	-
Anais do SBQS – Simpósio Brasileiro de Qualidade de Software	Não	-
Anais do WER – Workshop de Engenharia de Requisitos	Sim	-

Fonte: Elaborado pelo autor.

### 3.4.6. Busca Primária



A etapa de busca primária é em que os pesquisadores utilizam métodos previamente definidos para encontrar os estudos primários nas fontes de pesquisas selecionadas para a pesquisa. Essa busca pode ser feita de duas formas:

- **Buscas Manuais:** são aquelas definidas para fontes que não possuem uma engine de busca, ou seja, não possui uma catalogação de estudos primários que permita o retorno de estudos por meio de uma string de busca. Geralmente essas fontes são as de anais de eventos ou periódicos que não têm seus trabalhos indexados em bases de dados.
- **Buscas Automáticas:** são aquelas definidas para fontes que possuem uma engine de busca, ou seja, as que possuem trabalhos indexados em repositórios de estudos online.

Para a busca manual, a estratégia deu-se da seguinte forma:

- Encontrar os Anais dos Eventos selecionados como fontes de pesquisas, levando em consideração as restrições da pesquisa.
- Analisar todos os estudos primários dispostos nos anais por meio da leitura de título e resumo dos mesmos.

Já os métodos que foram utilizados para as buscas automáticas seguiram a seguinte estratégia:

- Identificação de Palavras-chave e sinônimos.
- Geração de Strings de Busca.
- Realização das buscas nas fontes de pesquisas selecionadas (aplicação de strings de busca).

Nas subseções seguintes são detalhadas as estratégias que foram utilizadas para as buscas automáticas no contexto deste trabalho.

### **3.4.7. Identificação de Palavras-Chaves e Sinônimos**

As palavras-chaves foram identificadas a partir das questões de pesquisa e em conformidade com à estrutura População, Intervenção e Resultados. De acordo com as restrições da pesquisa, os idiomas em que as palavras-chaves foram definidas foram o Português e o Inglês. A seguir são apresentadas as palavras-chaves para a questão de pesquisa principal.

- **POPULAÇÃO**

- Inglês: Software Development, Software Project, Project, Development, Organization, Enterprise, Company, Industry, Institute, Research Group, Technology Center.
- Português: Desenvolvimento de Software, Projeto de Software, Projeto, Desenvolvimento, Organização, Empresa, Companhia, Indústria, Instituição, Grupo de Pesquisa, Centros de Tecnologia.

- **INTERVENÇÃO**

- Inglês: Requirements Traceability, Traceability, Requirements Tracing.
- Português: Rastreabilidade de Requisitos, Rastreabilidade; Rastreamento de requisitos.

- **RESULTADOS**

- Inglês: Model, Process, Framework, Method, Technique, Methodology, Knowledge. Activity, Task, Tool, Software, Program, System, Application, Environment, Workbench.
- Português: Modelo, Processo, Framework, Método, técnica, metodologia, conhecimento, Atividade, Tarefa, ferramenta, software, aplicativo, sistema, programa, ambiente, workbench.

#### 3.4.8. Geração de Strings de Busca

Uma string de busca é o agrupamento das palavras-chave, por meio dos operadores <OR> e <AND>. O operador <OR> é utilizado para o agrupamento das palavras-chave e sinônimos, por elemento (População, Intervenção e Resultados). O operador <AND> é utilizado para agrupar o conjunto de palavras-chave definidos para todos os elementos, de acordo com a estrutura PICO (ou PIO no caso desta pesquisa) conforme segue (SANTOS, 2010):

**P <and> I <and> C <and> O**

Vale ressaltar que o elemento Comparação não está no contexto desse trabalho, logo o conjunto de palavras-chave para esse elemento é vazio. Para a questão de pesquisa foram consideradas duas strings de busca (uma para busca em inglês e outra para o português) e com base nesta foi gerada uma string de busca para cada fonte de

pesquisa automática devido à diferença entre as máquinas de busca de cada fonte ser diferente.

- **Para Q1:**

- **Inglês:** ("Software" OR "Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center") AND ("*Requirements Traceability*" OR "*Traceability*" OR "*Requirements Tracing*") AND ("Model" OR "Process" OR "Framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench").
- **Português:** ("Software" OR "Projeto" OR "Desenvolvimento" OR "Organização" OR "Empresa" OR "Companhia" OR "Indústria" OR "Instituto" OR "Grupo de Pesquisa" OR "Centro de Tecnologia") AND ("Rastreabilidade de Requisitos" OR "Rastreabilidade" OR "Rastreamento de Requisitos") AND ("Modelo" OR "Processo" OR "Framework" OR "Método" OR "Técnica" OR "Metodologia" OR "Tarefa" OR "Ferramenta" OR "Sistema" OR "Programa" OR "Ambiente" OR "Workbench").

- **Para IEEE:**

- **Inglês:** ("Software\*" AND ("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center")) AND ("*Requirements Traceability*" OR "*Traceability*" OR "*Requirements Tracing*") AND ("Model" OR "Process" OR "framework" OR "Method" OR "*Technique*" OR "*Methodology*" OR "*Task*" OR "*Tool*" OR "*System*" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories"))).

- **Para El Compendex:**

- **Inglês:** Software AND (Project OR Development OR Organization OR Enterprise OR Company OR Industry OR Institute OR Research Group OR Technology Center) AND (Requirements Traceability) AND (Model OR Process OR framework OR Method OR Technique OR Methodology

OR Task OR Tool OR System OR Application OR Environment OR Workbench OR Problem OR Issue OR Challenge OR Success Stories).

- **Para ISI Web of Knowledge:**

- **Inglês:** (((("Software" AND ("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center")) AND ("*Requirements Traceability*" OR "*Traceability*" OR "*Requirements Tracing*") AND ("Model" OR "Process" OR "framework" OR "Method" OR "*Technique*" OR "*Methodology*" OR "*Task*" OR "*Tool*" OR "*System*" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories")))).

- **Para ACM:**

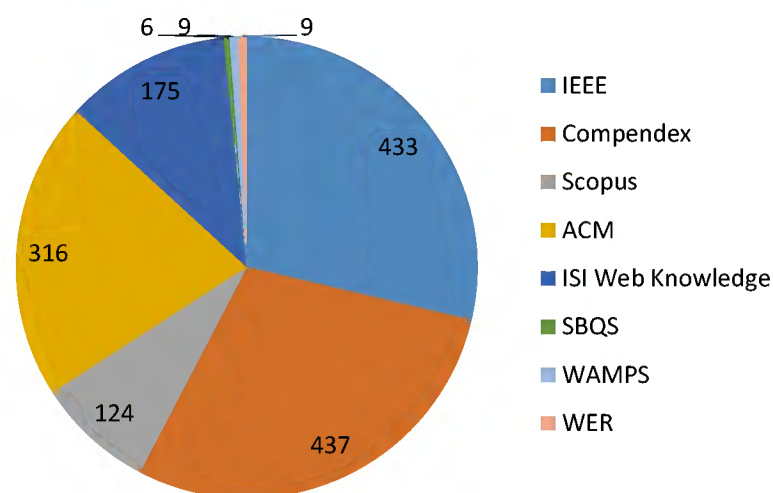
- **Inglês:** all of this text (and): "Software" "Requirements Traceability" any of this text (or): "Project" "Development" "Organization" "Enterprise" "Company" "Industry" "Institute" "Research Group" "Technology Center" "*Traceability*" "*Requirements Tracing*" "Model" "Process" "*framework*" "Method" "*Technique*" "*Methodology*" "*Task*" "*Tool*" "System" "Application" "Environment" "Workbench" "Problem" "Issue" "Challenge" "Success Stories".

- **Para Scopus:**

- **Inglês:** TITLE-ABS-KEY(("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center") AND ("Requirements Traceability" OR "Requirements Tracing") AND ("Model" OR "Process" OR "framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories")) AND (LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-

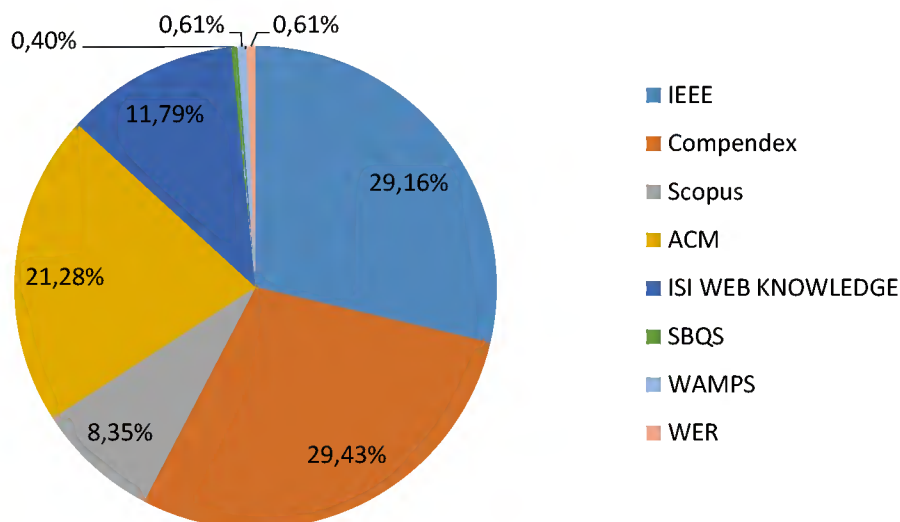
TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008)OR LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006) OR LIMIT-TO(PUBYEAR, 2005) OR LIMIT-TO(PUBYEAR, 2004) OR LIMIT-TO(PUBYEAR, 2003).

Conforme definido, foi realizada a pesquisa por trabalhos em cada base de dados por meio da aplicação das strings de busca apresentadas acima, para as fontes automáticas e a estratégia de leitura de título e resumo para fontes manuais. Obteve-se, então, um total de 1485 trabalhos, no qual 433 trabalhos foram identificados na IEEE, 437 na El Compedex, 124 na Scopus, 316 na ACM, 175 na ISI Web Knowledge, 6 no SBQS, 9 no WAMPS, e por fim, 9 trabalhos foram identificados no WER. As proporções podem ser visualizadas na Figura 3.2.



**Figura 3.2 – Número de Estudos retornados a partir da Busca Primária.**

O percentual de artigos retornados em cada fonte de pesquisa em relação ao total (1485 estudos) pode ser visualizado na Figura 3.3.



**Figura 3.3 – Percentual de estudos retornados a partir da Busca Primária.**

É importante frisar que a quantidade de estudos retornada por cada fonte pode ser justificada por diversos fatores, tais como, a quantidade total de estudos na fonte e a relevância da fonte para a questão de pesquisa.

Além disso, vale informar que durante a busca primária, ocorreu uma mudança na fonte de pesquisa ISI Web Knowledge (troca de portal de busca). Para lidar com este fato, para fim de organização, os estudos retornados desta base foram agrupados em duas categorias: “ISI Web Knowledge Dezembro de 2013” com 87 estudos retornados e “ISI Web Knowledge Fevereiro de 2014” com mais 88 estudos diferentes retornados.

Nas fontes de pesquisa SBQS, WAMPS e WER a busca foi feita de forma manual, e dessa forma os estudos retornados desta base já foram considerados estudos incluídos de acordo com os critérios de inclusão.

### **3.5. Condução da Revisão**

No contexto desse trabalho a fase de condução foi dividida em: Seleção de Estudos Primários, Avaliação de Qualidade dos Estudos e a Estratégia de Extração de Dados.

#### **3.5.1. Seleção dos Estudos Primários**

A etapa de seleção de estudos primários pode ser dividida em: Definição de Critérios de Inclusão e Exclusão de Estudos Primários e Definição do Processo de Seleção dos mesmos.

### 3.5.1.1. Critérios de Inclusão e Exclusão dos Estudos Primários

Os critérios de Inclusão e Exclusão dos estudos primários são os que vão nortear os pesquisadores na seleção dos estudos que foram coletados das fontes de pesquisas, além do que determina o rigor da pesquisa e impossibilita os vieses dos pesquisadores no momento da seleção.

Os critérios de exclusão dos artigos foram:

- CE.1)** Artigos que não estejam disponíveis livremente para consulta ou download (em versão completa) nas fontes de pesquisa ou por meio de busca manual (para artigos que não sejam fornecidos na íntegra) realizada nas ferramentas de busca Google (<http://www.google.com.br/>) e/ou Google Scholar (<http://scholar.google.com.br/>).
- CE.2)** Artigos que claramente não atendam as questões de pesquisa;
- CE.3)** Artigos repetidos (em mais de uma fonte de busca) tiveram apenas sua primeira ocorrência considerada.
- CE.4)** Artigos duplicados tiveram apenas sua versão mais recente ou a mais completa considerada, salvo casos em que haja informações complementares.
- CE.5)** Estudos enquadrados como resumos, keynote speeches, cursos, tutoriais, workshops e afins.
- CE.6)** Artigos que não mencionem as palavras-chave da pesquisa no título, resumo ou nas palavras-chave do artigo, salvo trabalhos que abordem melhoria do processo de software nos quais seja observada a possibilidade da Rastreabilidade de Requisitos ser tratada ao longo do trabalho.
- CE.7)** Excluir se o estudo não estiver inserido no contexto de Projetos de Software, Indústria de Software ou Engenharia de Software.
- CE.8)** Excluir se o estudo não estiver apresentado em uma das linguagens aceitas (Inglês e Português).

Os critérios de inclusão dos artigos basearam-se em:



- CI.1)** Estudos que apresentem primária ou secundariamente abordagens (padrões e CASEs) de apoio às atividades de Rastreabilidade de Requisitos.
- CI.2)** Estudos que apresentem relatos de experiência da indústria, ou pesquisas de caráter experimental ou teórico, contanto que apresentem exemplos de aplicação, descrição de experimentos ou casos reais de uso de abordagens (padrões e CASEs) para apoio às atividades de Rastreabilidade de Requisitos.

#### 3.5.1.2. Processo de Seleção dos Estudos Primários

Durante a condução desta revisão sistemática, os estudos primários foram identificados conforme o processo seguinte:

1. Realizou-se as buscas em todas as fontes selecionadas, por meio de strings de busca. Estudos claramente irrelevantes à pesquisa foram descartados. Os artigos foram catalogados na ferramenta Mendeley Desktop (MENDELEY Ltda, 2015), estabelecendo uma planilha com a lista, para cada pesquisador, de possíveis estudos primários.
2. A partir da leitura de resumo, introdução e conclusão, os artigos foram avaliados quanto aos critérios de inclusão e exclusão, e o resultado foi registrado.
3. Os dois pesquisadores responsáveis pela seleção dos artigos entraram em consenso, quando necessário. Isso ocorreu quando não houve uma unanimidade na inclusão de um artigo e pelo menos um pesquisador decidiu incluir.
4. Na fase de consenso, em caso de discordância sobre a inclusão de algum estudo, o mesmo foi incluído.
5. Posteriormente, o índice de concordância foi medido através do cálculo do valor Kappa, utilizado por Bjørnson & Dingsøyr (2008), para futura referência.
6. Os estudos primários identificados foram posteriormente lidos em totalidade e, então, foi aplicada a avaliação de qualidade e a estratégia de extração de dados, conforme descrito nas subseções seguintes.



Foram aplicados os critérios de exclusão e inclusão e o processo de seleção de estudos primários acima apresentados. As tabelas referentes à aplicação dos critérios de inclusão e exclusão nas bases de dados IEEE Xplore Digital Library, El Compedex, Scopus, ACM, ISI Web Knowledge, SBQS, WAMPS e WER podem ser visualizadas nos Apêndices A, B, C, D, E, F, G e H respectivamente. A Tabela 3.1 apresenta as estatísticas de quantos estudos foram excluídos ou incluídos antes e depois do consenso em cada base de dados.

**Tabela 3.1 – Seleção de Estudos Primários**

Seleção de Estudos Primários		
Fontes	Estudos Retornados	Estudos Incluídos
IEEE	433	164
El Compedex	437	110
Scopus	124	16
ACM	316	73
ISI Web Knowledge	175	24
SBQS		6
WAMPS		9
WER		9
Total	1485	411

**Fonte: Elaborado pelo autor.**

Após o processo de seleção dos estudos primários, pode-se verificar que embora a ISI Web of Knowledge apresente um número considerável de estudos retornados, com quase 12%, essa fonte teve uma representatividade pequena, quando comparada com as demais (com exceção da Scopus). Nos estudos primários selecionados esta fonte obteve apenas 6%. Isso pode ser explicado pela ordem que o processo de seleção ocorreu, já que muitos estudos dessas fontes foram enquadrados no critério de exclusão de estudos repetidos, devido estes já terem sido selecionados em outras fontes. A Tabela 3.2 apresenta a quantidade de ocorrências destes critérios em cada base de dados após a etapa de consenso em frequência absoluta dos resultados.

**Tabela 3.2. Estatísticas de Aplicação dos Critérios de Exclusão e Inclusão**

Critério	Quantidade					
	IEEE	Compendex	Scopus	ACM	ISI	Total
CE.1	4	0	1	14	7	26
CE.2	8	3	2	3	3	19

CE.3	0	57	85	61	112	315
CE.4	7	4	0	5	0	16
CE.5	8	12	0	16	0	36
CE.6	7	43	3	127	5	185
CE.7	33	14	2	0	9	58
CE.8	4	2	2	0	0	8
CI.1	196	100	11	18	14	339
CI.2	2	0	0	0	0	2

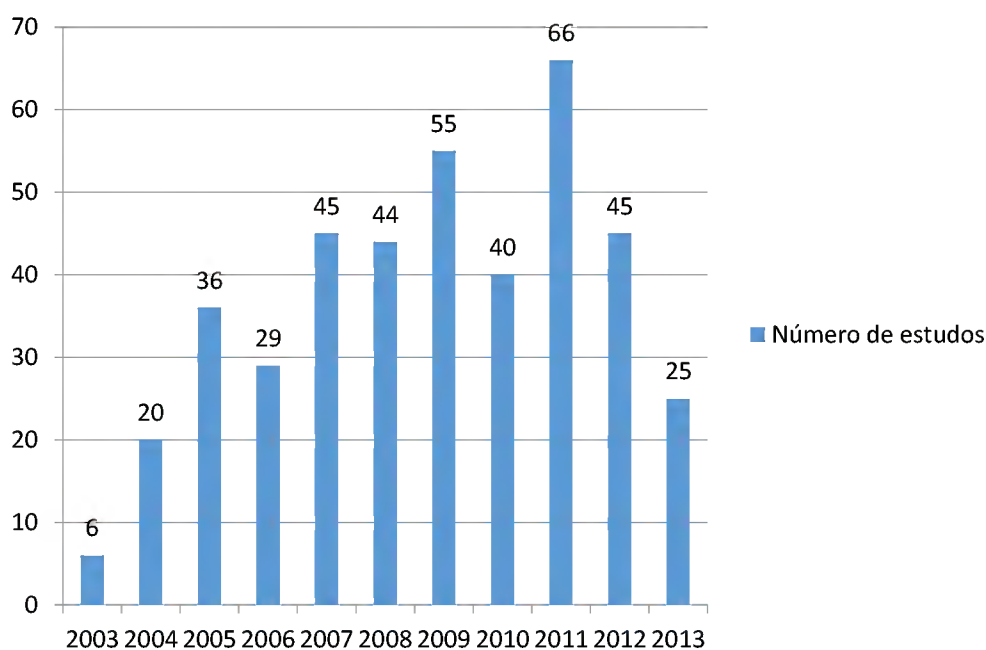
**Fonte: Elaborado pelo autor.**

Como se pode observar, houve uma grande quantidade de ocorrências dos critérios CE.3, CE.6 e CI.1. Com relação ao critério CE.3, verificou-se que como a base IEEE foi avaliada primeiro e por possuir uma quantidade relevante de estudos primários, as demais bases possuíam indexados estudos que já haviam sido selecionados na IEEE, o que ocasionou o grande número de estudos excluídos pelo referido critério. O critério CE.6 foi bastante aplicado para filtrar artigos que claramente não correspondiam ao escopo da pesquisa, por não mencionarem as palavras-chaves de busca em seu título ou resumo. O critério CI.1 restringe ainda mais o escopo da pesquisa à rastreabilidade de requisitos em relação ao mapeamento proposto.

Uma das restrições dessa pesquisa era o ano de publicação dos estudos, que deveria estar entre 01 de janeiro de 2003 até dezembro de 2013, neste caso, todos os estudos primários selecionados foram publicados no período desejado. A pesquisa contemplou, desta forma, um período de 11 anos, podendo, assim, identificar as informações acerca da literatura mundial atual para Rastreabilidade de Requisito.

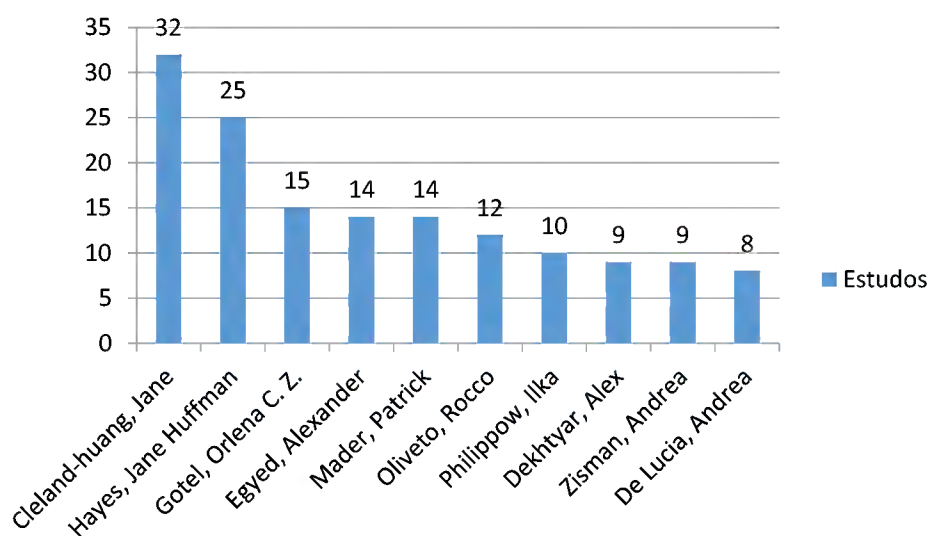
Ao analisar a quantidade de estudos publicados com relação ao ano, pode-se demonstrar e confirmar o que outros autores destacam no que diz respeito ao crescimento de pesquisas em Rastreabilidade de Requisitos no contexto de projetos de softwares na última década. De acordo com os resultados obtidos na busca nas fontes de pesquisa selecionadas, no ano inteiro de 2003 foram publicados apenas seis estudos referente à questão de pesquisa e a partir deste ano houve um grande crescimento. E mesmo que a partir de 2011 apresente uma queda nesta quantidade, o valor encontrado em 2013 ainda é superior ao ano de 2003, o que pode ter sido motivado pelo aumento

no reconhecimento da importância da área. A Figura 3.4 ilustra a distribuição dos estudos primários, identificados pelo processo de seleção, ao longo dos anos.



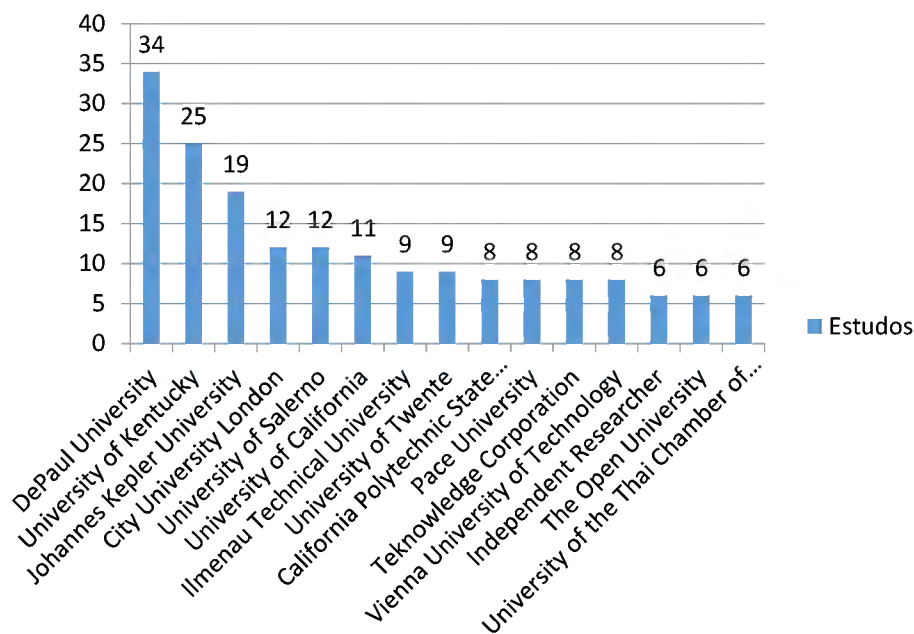
**Figura 3.4 – Número de Estudos ao Longo dos Anos**

Na Figura 3.5 pode-se observar o número de publicações por autores que entraram nesta pesquisa por meio de suas publicações dentro do espaço de tempo determinado. É válido informar que outros autores foram omitidos pelo número reduzido de artigos publicados com relação os que aparecem no gráfico. Por questão de organização, foram incluídos os dez autores que mais publicaram.



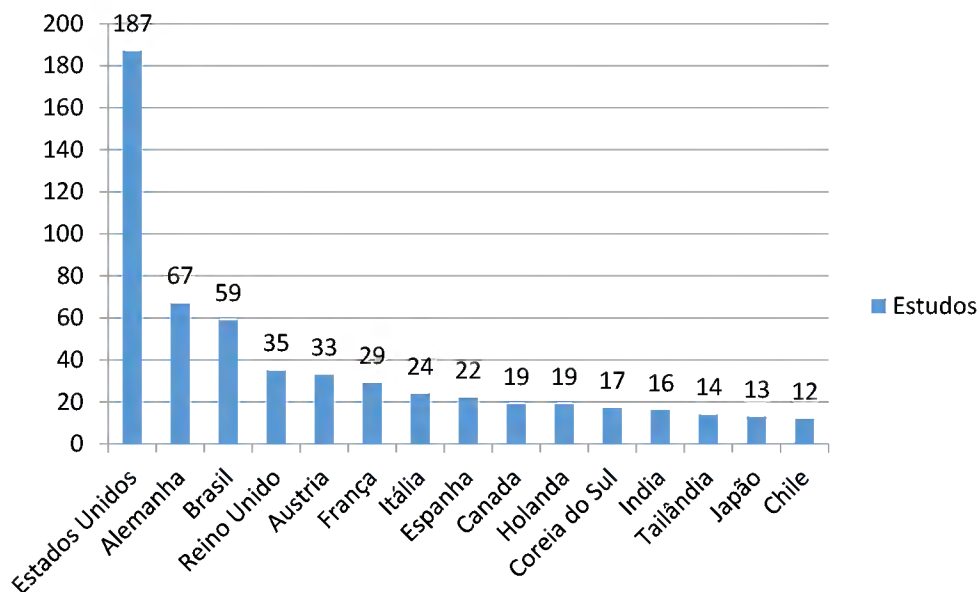
**Figura 3.5 – Número de Estudos por Autor**

A distribuição de estudos por instituição de filiação pode ser visualizada no Figura 3.6, onde o número de instituições com publicações é maior do que o que se visualiza no gráfico, porém algumas foram omitidas. O gráfico apresenta as principais instituições que publicam sobre Rastreabilidade de Requisitos.



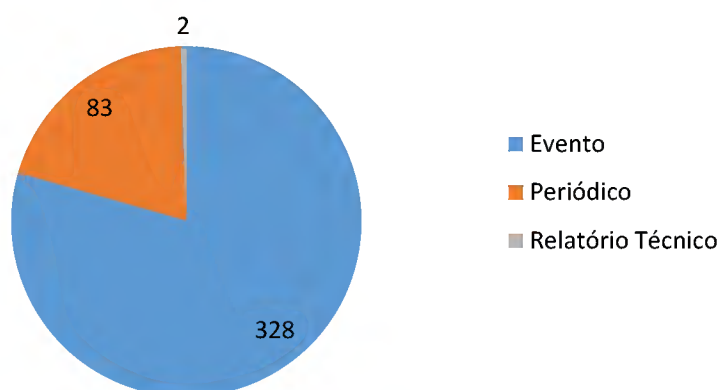
**Figura 3.6 – Número de Estudos por Instituição**

Na figura 3.7 pode-se observar quais os países que mais publicam sobre Rastreabilidade de Requisitos no contexto de projetos de software. Pode-se observar também a importância do Brasil, no que diz respeito a qualidade de software, porém essa expressividade pode se dá devido à escolha de fontes de pesquisas nacionais para compor esta pesquisa. Os Estados Unidos, como se pode observar, apresentou um número de publicações muito superior ao segundo lugar. Isso pode ser explicado devido ao modelo CMMI ter sido criado no país e possuir grande difusão pelo mesmo.



**Figura 3.7– Número de Estudos por País**

A distribuição dos trabalhos por tipo de publicação pode ser visualizada no Figura 3.8, em que se pode perceber que a maioria dos estudos primários, 79% (328 estudos), foi publicada em eventos, outra parcela, que corresponde a 20% (83 estudos) foram publicados em periódicos, e por último, com menos de 1% dos resultados (2 estudos) são relatórios técnicos. Pode-se observar que a grande maioria dos estudos incluídos, foram publicados por meio de eventos, e isso pode ser explicado devido ao fato de que a computação é uma ciência relativamente nova. Então ainda não existem tantos periódicos especializados na área de Engenharia de Software.



**Figura 3.8 – Trabalhos por Tipo de Publicação**

O Quadro 3.2 apresenta a distribuição dos estudos primários em relação ao local de publicação, de onde foi possível retirar a informação de qual(is) Evento(s),

Periódico(s) ou Relatório(s) Técnico(s) mais tiveram publicações nos últimos onze anos.

**Quadro 3.2 – Meios que mais Publicaram Estudos**

<b>Número de estudos por local de publicação</b>		
<b>Nome</b>	<b>Ocorrências</b>	<b>Tipo de publicação</b>
International workshop on Traceability in emerging forms of software engineering	26	Evento
International Conference on Software Engineering	22	Evento
International Requirements Engineering Conference	20	Evento
International Conference on Automated Software Engineering	16	Evento
Workshop on Traceability in Emerging Forms of Software Engineering	10	Evento
Workshop Anual do MPS	9	Evento
Workshop de Engenharia de Requisitos	9	Evento
European Conference on Software Maintenance and Reengineering	8	Evento
International Conference on Software Maintenance	8	Evento
Software & Systems Modeling	7	Periódico

**Fonte: Elaborado pelo autor.**

### **3.5.2. Avaliação da Qualidade dos Estudos Primários**

A qualidade de um artigo pode ser mensurada pela relevância e valor científico de seu conteúdo; também é considerado um critério de exclusão, a ser aplicado durante a condução da pesquisa. Desta forma, durante a análise dos estudos primários e coleta de resultados, foram aplicados os critérios de qualidade definidos abaixo, permitindo um processo adicional de validação dos estudos, de forma a identificar possíveis trabalhos que ainda devem ser desconsiderados da pesquisa e observar o grau de importância dos estudos individualmente para quaisquer comparações durante a síntese dos dados (KITCHENHAM, 2004).

Adicionalmente, a avaliação da qualidade pode servir como recomendação de estudos para futuras pesquisas, fornecendo informações a respeito da qualidade das informações de cada estudo avaliado (KITCHENHAM, 2004).

Os critérios de qualidade que foram aplicados aos estudos primários foram adaptados de Costa (2010), uma vez que descreviam critérios abrangentes o suficiente para cobrir o escopo dos estudos a serem considerados, com alterações para se adequarem aos objetivos e questões de pesquisa desta revisão sistemática. Segue a lista dos critérios de qualidade e suas categorias:

1. Introdução/Planejamento
  - a. Os objetivos ou questões do estudo são claramente definidos (incluindo justificativas para a realização do estudo)?
  - b. O tipo de estudo está definido claramente?
2. Desenvolvimento
  - a. Existe uma clara descrição do contexto no qual a pesquisa foi realizada?
  - b. O trabalho é bem/adequadamente referenciado (apresenta trabalhos relacionados ou semelhantes e se baseia em modelos e teorias da literatura)?
3. Conclusão
  - a. O estudo relata de forma clara e não ambígua os resultados?
  - b. Os objetivos ou questões do estudo são alcançados?
4. Critérios para a Questão de Investigação
  - a. O estudo lista primária ou secundariamente os Modelos, Processos, Métodos, Técnicas, Metodologia e afins, para apoiar atividades de Rastreabilidade de Requisitos?
  - b. O estudo apresenta Ferramentas de apoio às atividades de Rastreabilidade de Requisitos?
5. Critério Específico para estudos Experimentais
  - a. Existe um método ou um conjunto de métodos descrito para a realização do estudo?
6. Critério Específico para estudos Teóricos
  - a. Existe um processo não tendencioso na escolha dos estudos?



7. Critério Específico para Revisões Sistemáticas
  - a. Existe um protocolo rigoroso, descrito e seguido?
8. Critério Específico para Relato de Experiência Industrial
  - a. Existe uma descrição sobre a(s) organização(ões) onde foi conduzido o estudo?

Os critérios de qualidade (1) a (4) aplicam-se a todos os estudos primários avaliados, enquanto os critérios (5) a (8) aplicam-se especificamente aos respectivos tipos de trabalho mencionados. Estes tipos de estudos foram propostos por Easterbrooks (2007).

#### 3.5.2.1. Processo de Avaliação de Qualidade

Os estudos primários selecionados foram lidos em totalidade e, então, foram avaliados utilizando os critérios de qualidade. Para avaliar o grau de adequação aos critérios de qualidade, foi adotada a estratégia de avaliação proposta por Costa (2010), onde se utiliza a escala de Likert-5, permitindo respostas gradativas de 0 (discordo totalmente) a 4 (concordo totalmente), como apresentado no Quadro 3.3.

**Quadro 3.3 – Escala de Likert-5.**

<b>Escala de Likert-5</b>	
Concordo totalmente (4)	Deve ser concedido no caso em que o trabalho apresente no texto os critérios que atendam totalmente a questão.
Concordo parcialmente (3)	Deve ser concedido no caso em que o trabalho atenda parcialmente aos critérios da questão.
Neutro (2)	Deve ser concedido no caso em que o trabalho não deixe claro se atende ou não a questão;
Discordo parcialmente (1)	Deve ser concedido no caso em que os critérios contidos na questão não são atendidos pelo trabalho avaliado;
Discordo totalmente (0)	Deve ser concedido no caso em que não existe nada no trabalho que atenda aos critérios da questão.

**Fonte: Elaborado pelo autor.**

Para auxiliar a avaliação, a escala de Likert-5 foi adaptada para cada critério de qualidade, como se pode observar no Quadro 3.4.

Quadro 3.4. Escala para cada critério de qualidade.

Escala por Critério	
Critério	Escala
1a.	<p>4 - Define e justifica o estudo claramente.</p> <p>3 - Define claramente o estudo, porém a justificativa não é clara.</p> <p>2 - Define claramente o estudo, mas não justifica.</p> <p>1 - A definição dos objetivos do estudo não é clara.</p> <p>0 - Não define o estudo.</p>
1b.	<p>4 – Define o tipo de estudo, referenciando na literatura a metodologia.</p> <p>3 – Define o tipo de estudo, porém sem referenciar a metodologia.</p> <p>2 – Não define o tipo de estudo. É possível inferir facilmente.</p> <p>1 – Não define o tipo de estudo. É possível inferir com dificuldade.</p> <p>0 – Não é possível inferir o tipo de estudo.</p>
2a.	<p>4 – Define claramente uma seção com o contexto da pesquisa.</p> <p>3 – O contexto da pesquisa está incluído em uma seção não exclusiva.</p> <p>2 – O contexto da pesquisa está disperso ao longo do texto.</p> <p>1 – O contexto da pesquisa está disperso e é insubstancial.</p> <p>0 – O contexto da pesquisa não é abordado.</p>
2b.	<p>4 – O texto apresenta uma seção de trabalhos relacionados.</p> <p>3 – O texto apresenta trabalhos relacionados em uma seção não exclusiva.</p> <p>2 – O texto apresenta trabalhos relacionados dispersos ao longo do texto.</p> <p>1 – O texto não apresenta trabalhos relacionados, mas se apoia na literatura.</p> <p>0 – O texto não apresenta trabalhos relacionados nem se apoia na literatura.</p>
3a.	<p>4 – Resultados são claramente apresentados na seção de conclusão.</p> <p>3 – Resultados são claramente referenciados na seção de conclusão.</p> <p>2 – Resultados apresentados na conclusão não são claros.</p> <p>1 – Resultados referenciados na conclusão não são claros.</p> <p>0 – Não são apresentados resultados.</p>

Fonte: Elaborado pelo autor.

Quadro 3.4. Escala para cada critério de qualidade (cont.).

Escala por Critério	
Critério	Escala
3b.	<p>4 – Os resultados estão totalmente aderentes ao objetivo do estudo.</p> <p>3 – Os resultados estão aderentes ao objetivo do estudo, no entanto o autor faz ressalvas.</p> <p>2 – Os resultados são parcialmente aderentes ao objetivo do estudo.</p> <p>1 – Os resultados não estão aderentes ao objetivo do estudo.</p> <p>0 – Não é alcançado nenhum resultado.</p>
4a.	<p>4 – Algum dos elementos é claramente descrito.</p> <p>3 – Algum dos elementos é apresentado, mas não aprofundado.</p> <p>2 – Algum dos elementos é avaliado, porém não descrito.</p> <p>1 – Algum dos elementos é citado indiretamente.</p> <p>0 – Nenhum dos elementos é apresentado direta ou indiretamente.</p>
4b.	<p>4 – Apresenta uma ferramenta disponível para uso em versão funcional.</p> <p>3 – Apresenta ferramenta em versão funcional, porém não disponível para uso.</p> <p>2 – Apresenta um projeto ou proposta de ferramenta, incluindo a descrição de funcionalidades.</p> <p>1 – Apresenta apenas algoritmos de apoio ou análises de ferramentas de suporte à Rastreabilidade de Requisitos.</p> <p>0 – Não são apresentadas ferramentas de apoio.</p>
5a.	<p>4 – O método de experimento é definido e referenciado claramente.</p> <p>3 – O método de experimento é definido claramente.</p> <p>2 – O método de experimento é citado.</p> <p>1 – O método de experimento não é citado, porém é possível inferir.</p> <p>0 – Não é possível inferir o método de experimento.</p>
6a.	<p>4 – O texto descreve critérios para a escolha dos estudos.</p> <p>3 – O texto não descreve critérios para a escolha dos estudos, porém apresenta estudos que discordam do estudo apresentado.</p> <p>2 – O texto descreve apenas estudos aderentes ao estudo apresentado.</p> <p>1 – O texto descreve estudos insuficientes.</p> <p>0 – O texto não descreve estudos base.</p>

Fonte: Elaborado pelo autor.

Quadro 3.4. Escala para cada critério de qualidade (conclusão).

Escala por Critério	
Critério	Escala
7a.	<p>4 – O protocolo de revisão é apresentado, descrito e seguido.</p> <p>3 – O protocolo de revisão é apresentado e descrito, porém há evidências de que não foi seguido adequadamente.</p> <p>2 – O protocolo de revisão não foi suficientemente descrito.</p> <p>1 – O protocolo de revisão apenas foi citado ao longo do texto.</p> <p>0 – Não há um protocolo de revisão.</p>
8a.	<p>4 – A área de atuação, tamanho e origem da organização são informados.</p> <p>3 – Apenas duas das características do item 4 são informadas.</p> <p>2 – Apenas uma das características do item 4 é informada.</p> <p>1 – Nenhuma das características do item 4 é informada.</p> <p>0 – O estudo não foi conduzido em uma ou mais organizações.</p>

Fonte: Elaborado pelo autor.

Deve-se então, dada à soma das pontuações, enquadrá-los em um dos cinco níveis de qualidade classificados por Beecham (2007), tais como, apresentados na Tabela 3.3.

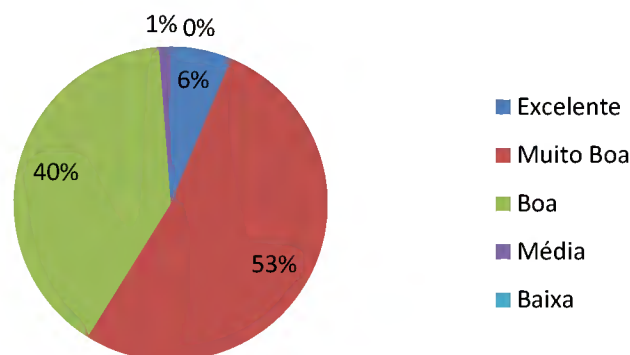
Tabela 3.3. Níveis de Qualidade

Faixa de Notas	Avaliação
Excelente	>86%
Muito Boa	66%-85%
Boa	46%-65%
Média	26%-45%
Baixa	< 26%

Fonte: Beecham (2007).

Com relação à qualidade dos estudos, foi possível notar prevalência nos estudos com qualidade “Muito Boa” e “Boa”, contando com 216 e 164 estudos, respectivamente. Em seguida, houve 26 ocorrências de trabalhos de qualidade “Excelente”. Houve apenas cinco estudos de qualidade “Média”, ao passo que não foi

identificado nenhum estudo com qualidade “Baixa”. A Figura 3.9 sintetiza estas informações.



**Figura 3.9. Percentual de Estudos por Categoria de Qualidade**

Com base na avaliação da qualidade realizada, pôde-se verificar que os estudos quanto à sua forma apresentavam em sua maioria uma boa qualidade, sendo assim, este poderia ser um indicativo da qualidade do conteúdo dos mesmos.

### 3.5.3. Estratégia de Extração de Informações

Para cada estudo primário analisado durante a etapa de extração de resultados, foi feito um resumo que identificou o tipo de abordagem proposta pelo estudo, o tipo de estudo e definições importantes sobre rastreabilidade de requisitos, a fim de responder as seguintes questões:

- **Q1A – Qual o contexto de aplicação da abordagem encontrada?** A abrangência da abordagem (referente à questão de pesquisa Q1) foi analisada em relação ao contexto de desenvolvimento onde foi encontrada a abordagem (acadêmico, industrial) e depois tipo de metodologia utilizada (ágil, tradicional).
- **Q1B – Quais as formas de rastreabilidade (para frente e para trás) são cobertas pela abordagem?** Para cada abordagem encontrada deveria ser identificada a forma de rastreabilidade que ela implementa. Pretendia-se identificar informações referentes a bidirecionalidade da rastreabilidade.
- **Q1C – Quais os tipos de rastreabilidade (vertical e horizontal) são cobertos pela abordagem?** Para cada abordagem identificada nos

estudos, foi analisado o tipo de rastreabilidade que a mesma implementa, a fim de verificar se a rastreabilidade oferecida refere-se ao relacionamento entre artefatos do mesmo tipo ou de tipos diferentes.

- **Q1D – Quais os ativos (papéis, artefatos) envolvidos?** Para cada abordagem encontrada nos estudos, verificou-se quais os ativos envolvidos em seu desenvolvimento.
- **Q1E – Quais técnicas são cobertas pelo apoio sistematizado?** Verificou-se entre as técnicas encontradas nos estudos, quais possuem apoio sistematizado.
- **Q1F – Quais Softwares foram utilizados?** Deveria registrar cada software mencionado no estudo, com uma breve descrição do mesmo (se houver);
- **Q1G – Qual a licença dos softwares utilizados?** Para cada software descrito em Q1F deveria ser identificado se este é Livre, Gratuito (porém não tem código aberto) ou Proprietário.

#### **3.5.4. Sumarização dos Resultados**

Como sumarização dos resultados desta revisão foi elaborado um catálogo de abordagens de apoio à rastreabilidade de requisitos, em que são dispostas informações importantes sobre técnicas e ferramentas aplicadas neste contexto, com o objetivo de reunir em um único instrumento essas abordagens de apoio dispersas na literatura, a fim de auxiliar organizações desenvolvedoras/mantenedoras de software ou qualquer interessado no assunto a ter uma visão ampla destas abordagens. O catálogo de abordagens será apresentado no próximo capítulo deste trabalho.

#### **3.6. Considerações Finais**

Este capítulo apresentou uma revisão sistemática da literatura aplicada à Rastreabilidade de Requisitos. Foram consideradas para fazer parte deste estudo nove fontes de pesquisa. Por meio da aplicação de critérios de seleção de fontes, foram selecionadas as bases de estudos ACM Digital Library, El Compendex, ISI Web of Knowledge, IEEEExplore Digital Library, Scopus, SBQS (Anais do Simpósio Brasileiro

de Qualidade de Software), WAMPS (Anais do Workshop Anual do MPS.BR) e WER (Anais do Workshop de Engenharia de Requisitos).

Uma estratégia de busca por estudos primários foi estabelecida e executada. Como resultado da busca, foram encontrados 1485 estudos. Cada estudo foi avaliado por dois pesquisadores, os quais decidiram selecionar 411 estudos relevantes para este trabalho. Posteriormente, a qualidade destes estudos foi analisada conforme critérios apresentados neste capítulo.

Como resultado da seleção e avaliação de qualidade dos estudos selecionados, foi possível coletar informações estatísticas relevantes às quais apontam alguns números identificados no contexto da RSL como, por exemplo: a quantidade de estudos selecionados com relação ao tempo; países; instituições; e autores que mais publicaram a respeito do tópico de interesse; e principalmente encontrar abordagens de apoio à rastreabilidade e classifica-las.

Os resultados da extração de dados dos estudos selecionados serão discutidos no Capítulo 4, o qual apresenta o catálogo de abordagens de apoio à rastreabilidade de requisitos no contexto de projetos de software desenvolvido no contexto desta dissertação.



## **4 UM CATÁLOGO DE ABORDAGENS DE APOIO A RASTREABILIDADE DE REQUISITOS**

Este capítulo discute a principal contribuição deste trabalho que é um Catálogo que visa apresentar abordagens de apoio à atividade de Rastreabilidade de Requisitos no que tange a projetos de software. Neste trabalho entende-se por catálogo um instrumento disponibilizado por meio de um documento virtual onde se encontram diversas abordagens de apoio a rastreabilidade de requisitos, que é de grande valia para a Engenharia de Software por reunir diversas abordagens que se encontram dispersas na literatura em um único banco de conhecimento, a fim de facilitar e ajudar organizações na escolha de qual abordagem utilizar quanto a implementação da atividade de rastreabilidade de requisitos.

No contexto deste trabalho, entende-se por abordagens de apoio: técnicas, processos, metodologias, frameworks, modelos e ferramentas. Devido à importância das técnicas no contexto da rastreabilidade de requisitos, sendo justificado pelo fato destas serem a principal abordagem na definição de como se dará o processo de rastreabilidade, além do que, é por meio delas que se consegue encontrar os elos de rastreabilidade, sendo assim então, partes integrantes das demais abordagens citadas. Outro fator a ser citado para a escolha de técnicas como a principal abordagem norteadora deste trabalho é o caráter de aplicação prática que este catálogo propõe-se. É possível então definir que a organização deste dar-se-á com base em técnicas, exemplos de processos e ferramentas que foram encontradas na literatura.

Vale ressaltar que a busca por abordagens de apoio à rastreabilidade de requisitos deu-se por meio da realização de uma Revisão Sistemática da Literatura apresentada no Capítulo 3, por ser um método que consiste em uma pesquisa organizada e metodológica na literatura, que possui como características: a abrangência, já que engloba todos ou, pelo menos, a grande maioria dos estudos relevantes à questão de pesquisa; não tendenciosa, pois possui um protocolo de revisão,

não sendo dirigida por interesses pessoais de seus pesquisadores; passível de replicação, já que existe um processo bem definido de sua execução; além de ter como principal meta a realização de uma pesquisa exaustiva na literatura, em busca de evidências que possam apoiar uma determinada hipótese, ou simplesmente a busca por conhecimento aprofundado acerca de certo fenômeno de interesse (MAFRA; TRAVASSOS, 2006).

#### **4.1. Catálogo de Abordagens de Apoio a Rastreabilidade de Requisitos**

Esta subseção apresenta o catálogo de abordagens, desde sua organização e estrutura, até sua apresentação por meio das técnicas de apoio a rastreabilidade.

##### **4.1.1. Organização do Catálogo**

Este catálogo está organizado pela ordenação alfabética das técnicas descobertas. Quanto a sua estrutura, cada técnica será analisada com base nos seguintes itens:

- **Tipo de Técnica:** Quanto à avaliação por tipo de técnica, pode ser encontrada na literatura a definição de técnicas de rastreabilidade divididas em: Manuais, que são aquelas onde a geração dos elos de rastreabilidade é realizada totalmente pela ação humana; Semiautomáticas, que são aquelas onde a geração dos elos de rastreabilidade é realizada por regras de associação automáticas, porém precisa da confirmação humana para ser efetivada; Automática, que são aquelas onde a geração dos elos de rastreabilidade é realizada por regras de associação automáticas, sem a necessidade de posterior confirmação humana para sua efetivação; e de Visualização, que são aquelas onde o objetivo não é a geração de elos de rastreabilidade e sim a forma de visualização dos mesmos. Porém, ao se analisar diversas técnicas, notou-se que algumas delas poderiam enquadrar-se em mais de um tipo especificado, de acordo com sua implementação, sendo assim, quanto ao tipo, cada técnica pode ser considerada possuidora de mais de um tipo. Ainda vale ressaltar que na literatura diversos autores reconhecem como técnicas àquelas que auxiliam na visualização dos elos de rastreabilidades já formados por algum dos tipos de técnicas já mencionados anteriormente.

Neste caso, para esse item têm-se as seguintes opções: Manual; Semiautomática; Automática; Híbrida; Visualização.

- **Fases Envolvidas:** Quanto à avaliação sobre as fases de um ciclo de vida de desenvolvimento de um software, este catálogo valer-se-á das fases de um processo de software, a fim de apresentar qual/quais da/das fase/fases a técnica em questão cobre. Sendo assim, uma técnica pode cobrir uma ou mais fases. Quando uma técnica permear no apoio à rastreabilidade em todas as fases, a fase será considerada como geral, caso não seja possível identificar uma ou mais fases específicas, a fase será então não especificada. Esse item possui as seguintes opções: Requisitos; Arquitetura; Implementação; Teste; Geral; Não Especificada;
- **Dimensão de Rastreabilidade:** Quanto à dimensão em que a rastreabilidade apresenta-se, este catálogo seguirá o conceito altamente difundido na literatura onde é definido que a rastreabilidade pode ser horizontal e vertical. Lindvall e Sandahl (1996) descrevem que a rastreabilidade vertical está relacionada com a capacidade de relacionar artefatos dependentes dentro de um modelo, enquanto que a rastreabilidade horizontal está relacionada à habilidade de relacionar artefatos entre diferentes modelos. Entre os artefatos que podem estar relacionados estão requisitos, artefatos de análise, de design, código-fonte, casos de teste, dentre outros. Para De Lucia, Fasano e Oliveto (2008), a rastreabilidade vertical fornece apenas uma visão limitada sobre os artefatos afetados por uma mudança. Sistemas complexos necessitam de modelos com vários níveis de abstração, como por exemplo, o código-fonte. Desta forma, muitas abordagens acabam estendendo a rastreabilidade vertical com a horizontal, permitindo gerenciar dependências entre requisitos e artefatos de design, entre requisitos e o código-fonte, entre requisitos e casos de teste, e entre artefatos de design e o código-fonte. Com base nisso, as seguintes opções serão consideradas: Horizontal; Vertical; Ambas; Não Especificado;
- **Estado de Rastreabilidade:** Quanto ao estado de rastreabilidade, pode-se dividi-la em Pré e Pós rastreabilidade. Gotel e Filkestein (1994) descrevem que pré-rastreabilidade refere-se a todos os aspectos de requisitos antes de documentá-los em uma especificação de requisitos, já pós-rastreabilidade refere-se a todos os aspectos de requisitos depois de documentá-los em uma

especificação de requisitos. Sendo assim, a análise seguirá da seguinte forma: Pré-rastreabilidade; Pós-rastreabilidade; Ambas;

- **Exemplo de Implementação:** Neste item, serão apresentados exemplos de implementação de cada uma das técnicas, caso existam;
- **Ferramentas Disponíveis:** Ferramentas que utilizam a técnica em questão serão apresentadas neste item;
- **Principais Referências:** Serão especificadas as principais referências, oriundas da Revisão Sistemática da Literatura, que tratam da técnica;
- **Notas:** Ocasionalmente, qualquer item dessa estrutura poderá precisar de uma nota com o objetivo de melhor explicar ou detalhar uma definição.

#### 4.1.2. Técnicas

Diversas técnicas foram encontradas na literatura por meio da RSL realizada, deste modo as técnicas encontradas são dispostas abaixo baseadas na estrutura definida anteriormente.

##### 4.1.2.1. Allocation Dirichlet Latent (ADL)

É um modelo probabilístico Bayesiano que foi introduzido por Blei et al. (2003). Esta técnica lida com estruturas latentes assim como a técnica denominada LSI (Latent Semantic Indexing). Ela representa todos os elementos como vetores de pesos latentes. A diferença com relação à outra técnica está na transformação do espaço de pesos de palavra-chave para o espaço de pesos tópicos latentes. ADL é implementada com um modelo Bayesiano hierárquico de três níveis. A ideia básica é que os documentos são representados como distribuições aleatórias sobre tópicos latentes, onde cada tópico é caracterizado por uma distribuição de palavras.

Em linhas gerais, cada tópico  $t$  é definido para ser uma distribuição de probabilidade  $\phi_t$  (sobre  $W$  palavras) extraídas de uma distribuição Dirichlet com o parâmetro  $\beta$ . Além disso, cada documento  $d$  está associado com uma distribuição de probabilidade  $\theta_d$  (sobre  $T$  tópicos), extraídos de uma Dirichlet com parâmetro  $\alpha$ . Vale

ressaltar, que uma amostra obtida a partir de uma distribuição de Dirichlet é precisamente em si uma distribuição discreta.

É importante notar que a ADL é uma estrutura de aprendizado de máquina não supervisionada, o que significa que não é necessário nenhum treinamento prévio de dados com rótulos de formação. A única entrada necessária para ADL é o conjunto de palavras (convertidos a um esparsa  $W \times$  matriz  $D$  após a remoção de palavras não relevantes) e o número desejado de tópicos  $T$  a ser aprendido.

No contexto da rastreabilidade esta técnica é utilizada sobre documentos de textos, Casos de Uso e códigos-fonte com a captura de palavras e tópicos e suas estruturas latentes que podem demonstrar o vínculo entre esses diferentes artefatos.

O Quadro 4.1 apresenta a catalogação desta técnica com base na estrutura definida na Seção 4.1.

**Quadro 4.3 – Informações gerais sobre a técnica ADL**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiautomática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical.	Pré-rastreabilidade e Pós-rastreabilidade.
<b>Nota</b>	
<p><b>Fases Envolvidas:</b> No processo de desenvolvimento de software essa técnica pode ser considerada como de uso geral, porém como explicado anteriormente é uma técnica da área de IR (Recuperação de Informação) que trabalha como a descoberta de informação por meio de palavras, com isso, alguns artefatos podem ser mais difíceis de serem rastreados, isso vai depender fortemente da forma que foram desenvolvidos.</p>	

**Fonte: Elaborado pelo autor.**

Como exemplo de implementação do ADL no trabalho de Dekhtyar et al. (2007) foi usada a implementação open source Java ADL, o ADL-J. A ADL é implementada como um modelo Bayesiano hierárquico de três níveis. Cada documento é modelado como uma mistura de  $k$  tópicos para a classificação. Assume-se o seguinte processo gerador para cada documento  $w$  em um corpus  $D$ :

- Escolha  $N \sim \text{Poisson}(\xi)$ .
- Escolha  $\theta \sim \text{Dir}(\alpha)$ .
- Para cada uma das palavras  $N$  WN:
- Escolha um tema  $z_n \sim \text{Multinomial}(\theta)$ .

- Escolha uma palavra  $w_n$  de  $p(w_n | z_n, \beta)$ , uma probabilidade multinomial condicionado sobre o tema  $z_n$ .

Links de candidatos entre os requisitos e elementos de design foram registrados, cuja distância euclidiana medida estava dentro de um determinado limite.

No Quadro 4.2 são apresentadas ferramentas que implementam a técnica ADL e as principais referências que tratam desta técnica na literatura.

**Quadro 4.2 – Ferramentas e Principais Referências sobre a técnica ADL**

Ferramentas Disponíveis	
Nome	Descrição
<b>Topically-Rich Artifact Search Engine (TRASE)</b>	Trase é um motor de pesquisa sobre os artefatos do projeto que aprende de forma dinâmica com um modelo LDA nos resultados de busca em tempo real. Trase foi implementado usando uma combinação das tecnologias Perl, AJAX e Lucene.  <b>Disponível em:</b> Não disponível para download.
Principais Referências	
<ul style="list-style-type: none"> <li>• ASUNCION, H. U.; ASUNCION, A. U.; TAYLOR, R. N. <b>Software traceability with topic modeling</b>. 2010 ACM/IEEE 32nd International Conference on Software Engineering, 1, 95 – 104. doi:10.1145/1806799.1806817. 2010.</li> <li>• BLEI, D.; NG, A.; JORDAN, M. <b>Latent Dirichlet Allocation</b>. Journal of Machine Learning Research, 3:993–1022, 2003.</li> <li>• DEKHTYAR, A.; HAYES, J. H.; SUNDARAM, S.; HOLBROOK, E. A.; DEKHTYAR, O. <b>Technique Integration for Requirements Assessment</b>. 15th IEEE International Requirements Engineering Conference (RE 2007), 141–150. doi:10.1109/RE.2007.17. 2007.</li> <li>• HEINRICH, G. <b>LDA-J Library</b>. Code available at <a href="http://www.arbylon.net/projects/">http://www.arbylon.net/projects/</a>.</li> </ul>	

Fonte: Elaborado pelo autor.

#### 4.1.2.2. Clustering

A ideia desta técnica é utilizar um o cluster para melhorar o desempenho de algoritmos de recuperação de informação (IR). Ela tem sido empregada para abordar várias questões, tais como a melhoria do desempenho de recuperação de informação, da navegação de documento, de descobertas de tópicos, da organização dos resultados de pesquisa, e do conceito de decomposição dos mesmos. Além disso, clustering tem sido utilizada para melhorar a qualidade de uma consulta de IR, bem como os resultados da organização.



Com relação à consulta de informação, clustering pode ser particularmente útil para abordar o conhecido problema "consulta curta", em que os usuários podem inserir informações insuficientes para produzir um rico clustering de resultados. Neste contexto, dados de consulta histórica são recolhidos, e as consultas são colocadas em clusterings. Quando uma nova consulta é recebida, é comparada com o clustering de consultas existentes e associada com ao que é mais semelhante a ela. Termos de clustering são então usados para expandir e melhorar a consulta atual.

Duan e Cleland-Huang (2007) demonstraram os benefícios de utilizar clustering como técnica de rastreabilidade automática. Eles examinaram três algoritmos: hierarchical agglomerative, k-means, e bisecting divisive. Eles relataram que a granularidade dos algoritmos de clustering foi razoável, 5-6 clusters ou mais, não foi observada diferença significativa entre os clusterings produzidos pelos três algoritmos. O Quadro 4.3 e apresenta a catalogação desta técnica.

**Quadro 4.3 – Informações gerais sobre a técnica Clustering**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical.	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Como exemplo de aplicação dessa técnica pode-se destacar o apresentado por Chen e Grundy (2011), afirmando que, em geral, cada documento tem uma estrutura hierárquica inerente. Os documentos são geralmente divididos em seções com títulos. Cada seção tem outra seção a que faz parte, ou alguma seção que a compõe ou então alguma seção que pertence a mesma seção a qual ela está inserida. Ou seja, existem relacionamentos entre essas seções. Por exemplo, no trabalho citado, a seção 3a faz parte da seção 3 e possui mais três seções irmãs, as seções 3b, 3c, e 3d. Deve-se aproveitar essas relações para reduzir o número de ligações falsas recuperadas usando clustering. Tomando ligações entre a classe “java.awt.dnd.DragSource” e seções em um documento como um exemplo para ilustrar um algoritmo de clustering.

A Figura 4.1 mostra um exemplo em que algumas seções estão relacionados com a classe “DragSource”. Cada linha representa um link, linhas de cor azul e em itálico referem-se a ligações verdadeiras. Antes de iniciar a etapa de inicialização,



todos os links recuperados são agrupados com base em classes; ou seja, as ligações relacionadas à mesma classe são agrupadas juntas.

Clustering é realizado então em cada grupo que representa seções relacionadas a uma mesma classe. Em seguida, o algoritmo seleciona  $k$  clusters de acordo com o número de ligações com valores de similaridade  $\geq s$ . Cada cluster contém uma dessas seções relacionadas. Quando o grupo contém ligações com um valor de similaridade igual a 1, o algoritmo usa  $s = 1$ . Caso contrário, o algoritmo usa  $s = 0,3$  para criar clusters. A partir da observação empírica encontram-se quatro razões para usar este último valor quando nenhum valor de similaridade das ligações no grupo é igual a 1.

Em primeiro lugar, a maioria dos links de falhas tem uma pontuação de similaridade  $\leq 0,3$ . Em segundo lugar, as ligações com similaridade  $\geq 0,3$  são mais propensas a ser verdade. Em terceiro lugar, se usa  $s \leq 0,3$ , a abordagem recupera muitas ligações de falhas e apenas links um pouco mais verdadeiros. Em quarto lugar, se  $s \geq 0,3$ , a abordagem diminui ligeiramente o número de ligações de falha, mas não obtém as ligações mais verdadeiras. Empiricamente, portanto, é encontrado o limite de 0,3 para ser a melhor escolha para os sistemas de destino utilizados. Eles afirmam, porém, que é necessário a realização de mais experimentos, para validar a sua adequação para outros sistemas. É possível notar na Figura 4.1 que 15 ligações têm uma pontuação de similaridade = 1. O algoritmo cria, assim, 15 grupos, cada um contendo uma destas seções.

**Figura 4.1. Relações relacionadas para a classe java.awt.dnd.dragsource**

1.0	---dnd1.pdf:2.1 Overview
1.0	---dnd1.pdf:2.2.1 DragGestureRecognizer
1.0	---dnd1.pdf:2.3 Drag Source
1.0	---dnd1.pdf:2.3.1 The DragSource definition
1.0	---dnd1.pdf:2.3.2 The DragSourceContext Definition
1.0	---dnd1.pdf:2.3.3 The DragSourceDragEvent Definition
1.0	---dnd1.pdf:2.3.6 The DragSourceDropEvent Definition
1.0	---dnd1.pdf:2.4.3 The DropTargetContext Definition
1.0	---dnd1.pdf:2.4.4 The DropTargetListener Definition
1.0	---dnd1.pdf:2.4.5 The DropTargetDragEvent and ...
1.0	---dnd1.pdf:2.5 Data Transfer Phase
1.0	---dnd1.pdf:2.5.1 FlavorMap and SystemFlavorMap
1.0	---dnd1.pdf:2.5.2 Transferring Data across the JVM ...
1.0	---dnd1.pdf:3.0.1 What are the implications of the ...
1.0	---dnd1.pdf:3.0.3 Lifetime of the Transferable(s)?
0.06234840	---dnd1.pdf:3.0.4 Implications of ACTION_...
0.05901812	---dnd1.pdf:2.3.3 The DragSourceListener Definition
0.05479498	---dnd1.pdf:2.5.3 Transferring lists of files across...
0.05061744	---dnd1.pdf:2.3.4 The DragSourceEvent Definition
0.04429026	---dnd1.pdf:3.0.2 Inter/latra VM transfers?
0.04183720	---dnd1.pdf:3.0.5 Semantics of ACTION_...
0.03083606	---dnd1.pdf:2.4.1 java.awt.Component additions...
0.02992645	---dnd1.pdf:2.5.4 Transferring java.rmi.Remote ...
0.02787049	---dnd1.pdf:3.0 Issues
0.02787049	---dnd1.pdf:2.0 API
0.02658593	---dnd1.pdf:2.4.2 The DropTarget Definition
0.02162598	---dnd1.pdf:1.1 Provision of a platform independent
0.01930924	---dnd1.pdf:2.2 Drag Gesture Recognition
0.01330427	---dnd1.pdf:1.2 Integration with platform ...
0.01330427	---dnd1.pdf:Appendix A : DropTargetPeer definition
0.01317056	---dnd1.pdf:1.0 Requirements
0.00965462	---dnd1.pdf:2.4.6 Autoscrolling support
0.00064342	---dnd1.pdf:Appendix B : DragSourceContextPeer definition
0.00064342	---dnd1.pdf:Appendix C : DropTargetContextPeer definition

Fonte: Chen e Grundy (2011).

No Quadro 4.4 são apresentadas ferramentas que implementam a técnica de Clustering e as principais referências que tratam desta técnica na literatura.

**Quadro 4.4 – Ferramentas e Principais Referências sobre a técnica Clustering**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>Poirot</b>	<p>Poirot utiliza um modelo de inferência probabilística para gerar dinamicamente links candidatos entre a consulta e documentos que estão sendo pesquisadas. Ele calcula a probabilidade valor <math>Pr(d   q)</math> para consulta <math>q</math> e documento <math>d</math>, onde <math>q</math> representa qualquer artefato utilizado para iniciar um rastro, e <math>d</math> representa um documento dentro de um conjunto específico de documentos pesquisáveis. Esta ferramenta é apresentada por Duan e Cleland (2007), habilitada para a utilização de Clustering com o algoritmo bisecting divisive. A fim de facilitar a compreensão humana, Poirot aplica-se à regra de ouro, <math>7 \pm 2</math>, para determinar o nível de granularidade do cluster desejado. Outro ponto chave de Poirot é a sua capacidade de superar a limitação do clustering monotético, ou seja, clustering com base apenas em uma única característica comum.</p> <p><b>Disponível em:</b> não disponível para download.</p>
<b>Tracter</b>	<p>Tracter ferramenta desenvolvida por Mahmoud (2011) para resolver vários problemas de apresentação de Poirot. A intenção do projeto era a aplicação de novas interfaces de usuário de pesquisa, de modo a aumentar a eficiência de navegação do analista humano.</p> <p><b>Disponível em:</b> não disponível para download.</p>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• LIN, J.; LIN, C.; CLELAND-HUANG, J.; SETTIMI, R.; AMAYA, J.; BEDFORD, G.; ZOU, X. <b>Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability</b>. 14th IEEE International Conference Requirements Engineering, 363 – 364. 2006</li> <li>• MAHMOUD, A.; NIU, M. <b>TraCter: a tool for candidate traceability link clustering</b>. in RE, 2011, pp. 335–336.</li> <li>• DUAN, C.; CLELAND-HUANG, J. <b>Clustering support for automated tracing</b>. Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering - ASE '07, 244. doi:10.1145/1321631.1321668. 2007.</li> <li>• CHEN, X.; GRUNDY, J. <b>Improving automated documentation to code traceability by combining retrieval techniques</b>. 26th IEEE/ACM International Conference on Automated Software Engineering (ASE), 223–232. doi:10.1109/ASE.2011.6100057. 2011.</li> </ul>	

**Fonte:** Elaborado pelo autor.

#### 4.1.2.3. Constraint Networks

Constraints Networks, segundo Bowen (1990), são redes de restrições, onde uma coleção de objetos e um conjunto de restrições especificam relacionamentos que devem ser satisfeitos pelos valores assumidos por esses objetos. O grande atrativo dessas redes, segundo o autor, é que as restrições podem suportar uma inferência não direcional. Isto significa que quando os valores são adquiridos por qualquer um dos objetos envolvidos em uma restrição, estes podem ser inferidos por outros objetos ligados à restrição. Assim, por exemplo, uma rede de restrições pode capturar o efeito de qual decisão de projeto tem sobre as demais etapas de construção de um produto (rastreadibilidade). Igualmente, a mesma rede pode limitar as opções do designer se as decisões de produção são definidas inicialmente. A rede de restrições é composta por um conjunto finito de variáveis  $X = \{X_1, \dots, X_n\}$ , cada um associado a um domínio de valores discretos,  $D_1, \dots, D_n$  e um conjunto de restrições,  $\{C_1, \dots, C_t\}$ . Cada uma das restrições é expressa como uma relação, definidas sobre um subconjunto de variáveis, cujas tuplas são todas atribuídas de valores simultâneos para os membros deste subconjunto variável. O Quadro 4.5 apresenta a catalogação desta técnica.

**Quadro 4.5 – Informações gerais sobre a técnica Constraint Networks**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreadibilidade</b>	<b>Estado de Rastreadibilidade</b>
Horizontal e Vertical.	Pós-rastreadibilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

Fonte: Elaborado pelo autor.

Para esta técnica não foram encontrados nenhum exemplo de implementação e nenhuma ferramenta que a implemente. Neste cenário, o Quadro 4.6 apresenta apenas as principais referências de trabalhos encontrados na literatura que tratam desta técnica.

**Quadro 4.6 – Principais Referências sobre a técnica Constraint Networks**

<b>Principais Referências</b>
<ul style="list-style-type: none"> <li>• J. Bowen, P. O'Grady, L. Smith. <b>A Constraint Programming Language for Life-cycle Engineering</b>. Artificial Intelligence in Engineering, vol. 5, no. 4, pp. 206- 220, 1990.</li> <li>• Dechter, R. <b>Constraint Networks (Survey)</b>. In Encyclopedia of Artificial Intelligence, 2nd edition, 1992, John Wiley &amp; Sons, Inc., pp. 276-285.</li> </ul>

Fonte: Elaborado pelo autor.

#### 4.1.2.4. Cross References

Cross References (Referências Cruzadas) são representadas como forma de relacionar artefatos e permitir a navegação com a fonte de elemento de informação por meio de hyperlinks, matrizes ou grafos. Geralmente é difícil ter uma visão global dos relacionamentos de rastreabilidade quando se utiliza referências cruzadas para representar os relacionamentos de rastreabilidade. Por este motivo, também é mais difícil de encontrar relações que não foram identificadas ou identificadas incorretamente.

A especificação de requisitos, por exemplo, é um documento com muitas referências cruzadas, bem como com referências em documentos diferentes. Nesse sentido, os links relevantes são, então, incorporados como ponteiros em um texto, que pode ser de linguagem natural informal ou uma especificação formal. Outra forma da utilização de referências cruzadas está nas ligações entre diagramas.

Segundo Wieringa (1995), o uso de referências cruzadas é simples de entender e pode ser implementada facilmente. Pacotes existentes em latex já contém instalações de referências cruzadas. Outro ponto a ser destacado é que referência cruzada é útil para as especificações escritas, mas não para uma representação concisa de links como pode ser feito com matrizes. Referências cruzadas são sempre ligações binárias, de modo que a maioria das ligações não podem ser facilmente representadas.

O Quadro 4.7 apresenta a catalogação desta técnica.

**Quadro 4.7 – Informações gerais sobre a técnica Cross References**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Manual	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical.	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Jackson (1991) apresenta um sistema em que os requisitos são escritos em linguagem natural, que estão todos disponíveis online e que contém referências cruzadas definidas pelo autor do documento. As referências cruzadas são usadas para extrair itens dependentes e para a produção de vários relatórios. O Quadro 4.8

apresenta ferramentas que implementam esta técnica e as principais referências de estudos encontrados na literatura que tratam da mesma.

**Quadro 4.8 – Ferramentas e Principais Referências sobre a técnica Cross References**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>RADIX</b>	<p>O sistema RADIX é um pacote onde é possível definir a referência cruzada entre requisitos e produzir diversos relatórios. A ferramenta RADIX fornece uma série de documentação de formatos macros nroff/troff e trabalhos em conjunto com os Macros de Compensação (MM) para facilitar a organização de requisitos em um documento. MM é um macro pacote de formatação de texto para utilização com o texto UNIX formatados em nroff e troff. MM pode ser usado para produzir cartas, relatórios, memorandos, documentos técnicos, manuais e livros.</p> <p><b>Disponível em:</b> <a href="http://lxr.free-electrons.com/source/include/linux/radix-tree.h?v=3.14">http://lxr.free-electrons.com/source/include/linux/radix-tree.h?v=3.14</a></p>
<b>C&amp;L - Cenários e Léxico</b>	<p>Esta ferramenta é destinada a apoiar o trabalho do engenheiro de software nas atividades do processo de desenvolvimento. Ela cria automaticamente elos de rastreabilidade entre símbolos léxicos e cenários da aplicação. Isto facilita a automação da criação dos elos, já que cenários fazem referência a símbolos registrados no léxico. A ferramenta percorre os cenários, identificado os símbolos presentes no léxico e criando os hiperlinks entre cenários e léxico. Como um termo do léxico pode estar presente em mais de um cenário, cria-se uma "teia" de referências cruzadas entre léxico e cenários.</p> <p><b>Disponível em:</b> <a href="http://transparencia.les.inf.puc-rio.br:8080/cel/visao/index.html">http://transparencia.les.inf.puc-rio.br:8080/cel/visao/index.html</a></p>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• WIERINGA, R. <b>An introduction to requirements traceability</b>. Tech. rep., Institute for Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands. 1995.</li> <li>• JACKSON, J. <b>A key frased based traceability scheme</b>. In <b>Colloquium on Tools and Techniques for Maintaining Traceability During Design</b>. SavoyPlace, London WC. R OBL, U.K., 2 december 1991.</li> <li>• YU, W. D. <b>Verifying software requirements: a requirements tracing methodology and tool RADIX</b>. IEEE Journal on Selected Areas in Communication, 1994.</li> <li>• WINKLER, S.; PILGRIM, J. <b>A survey of traceability in requirements engineering and model-driven development</b>. Software &amp; Systems Modeling, 9(4), 529–565. doi:10.1007/s10270-009-0145-0. 2009.</li> </ul>	

**Fonte:** Elaborado pelo autor.

#### 4.1.2.5. Event-Based Traceability (EBT)

Cleland-Huang et al. (2002) propuseram uma abordagem baseada em eventos para apoiar a geração de elos de rastreabilidade entre os requisitos e o modelo de desempenho, e entre os requisitos não-funcionais e artefatos de design e de código. Nessa abordagem, elos refinados de rastreabilidade são gerados dinamicamente durante a manutenção do sistema e o refinamento é baseado em links definidos pelo usuário. Esses links são especificados durante a criação, elaboração e construção do sistema. Cleland-Huang et al. (2003) utilizaram uma arquitetura publicação-assinatura para definir links de rastreabilidade que suportam análise de impacto automática sobre Requisitos não funcionais. Os links são estabelecidos entre requisitos de performance quantitativamente definidos, restrições na especificação dos requisitos, e variáveis localizadas em modelos de simulação de performance.

A técnica baseada em eventos suporta a geração dinâmica de rastreabilidade com base em regras invariantes de padrões de projeto, que são usadas para identificar os componentes críticos de classes. Segundo Cleland-Huang et al. (2003), ela pode ser usada para rastrear requisitos associados a modelos de desempenho, e também para rastrear a qualidade de requisitos implementados através de padrões de projetos já conhecidos, que podem ser ligados.

No contexto dessa técnica, um link de rastreabilidade não necessariamente é uma ligação, mas um registro de um evento. Em rastreabilidade baseada em eventos, requisitos e outros artefatos de engenharia de software estão ligados através de uma espécie de relacionamento publicação-assinatura, nos quais os requisitos assumem o papel de editores, enquanto artefatos dependentes agem como assinantes. Quando os requisitos forem alterados, os eventos são publicados em um servidor de eventos e notificações enviadas para todos os assinantes dependentes. As mensagens devem levar informação suficiente para fornecer semânticas significativas sobre cada caso, a fim de apoiar o processo de atualização. O Quadro 4.9 apresenta a catalogação desta técnica.

**Quadro 4.9 – Informações gerais sobre a técnica EBT**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiautomática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical.	Pós-rastreabilidade.



Nota
Não possui nenhuma nota.

**Fonte: Elaborado pelo autor.**

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.10 apresenta uma ferramenta que implementa esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.10 – Ferramentas e Principais Referências sobre a técnica EBT**

Ferramentas Disponíveis	
Nome	Descrição
<b>DOORS</b> + <b>outras</b> <b>implementações:</b>	<p>No trabalho de Cleland-Huang e Chang (2003), foi criado um protótipo que implementa a técnica EBT baseado em uma arquitetura de três níveis, constituído por três componentes principais, incluindo um disparador de evento, um gerenciador de serviço de eventos e um gerenciador de assinante. O disparador de evento foi implementado em cima da ferramenta DOORS, que é uma ferramenta para gestão de requisitos, para realizar a captura manual de eventos de mudanças e como ocorreram. O gerenciador de serviços de eventos foi desenvolvido como um servlet para apoiar o processo de inscrição pela Internet. O servidor de eventos recebe eventos publicados em um endereço atribuído, verifica os processos de arquivos de assinatura, a fim de identificar assinantes, e realizar as notificações de eventos para o assinante. O terceiro componente, o gerenciador de assinante, foi desenvolvido como um pequeno servidor que recebe notificações de eventos de um determinado tipo de artefato dependente.</p> <p><b>Disponível em:</b>  <a href="https://www.ibm.com/developerworks/downloads/r/ordng/">https://www.ibm.com/developerworks/downloads/r/ordng/</a></p>
Principais Referências	
<ul style="list-style-type: none"> <li>• CLELAND-HUANG J.; CHANG C.; WISE J. <b>Supporting Event Based Traceability through HighLevel Recognition of Change Events</b>. Proceedings of IEEE COMPSAC Conference, Oxford, England, August 2002.</li> <li>• CLELAND-HUANG, J.; CHRISTENSEN, C. <b>Event-based traceability for managing evolutionary change</b>. IEEE Transactions on Software Engineering, v. 29, n. 9, p. 796–810, 2003.</li> <li>• CLELAND-HUANG J.; SCHMELZER D. <b>Dynamic Tracing Non-Functional Requirements through Design patter Invariants</b>. Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003), Canada, October, 2003.</li> </ul>	

**Fonte: Elaborado pelo autor.**



## 4.1.2.6. ER Models

Ligações entre artefatos também podem ser representados por modelos de entidade relacionamento (ER). Os itens vinculados são entidades, as ligações são instâncias de relacionamento, ou seja, a rastreabilidade de requisitos pode ser representada por esse tipo de técnica. A representação ER tem a vantagem de que as ligações diversas podem ser representadas. Além disso, um modelo ER de ligações pode ser implementado utilizando qualquer tecnologia de banco de dados, o que tem a vantagem de consulta ad hoc e recursos de relatórios facilmente disponíveis. O Quadro 4.11 apresenta a catalogação desta técnica.

Quadro 4.11 – Informações gerais sobre a técnica ER-Model

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Manual	Arquitetura
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.12 apresenta a ferramenta que implementa esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.12 – Ferramentas e Principais Referências sobre a técnica ER-Models

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>ARTS</b>	Lockheed Missiles e Space Company, Inc. (LMSC) produziu um sistema automatizado de apoio à Rastreabilidade de Requisitos chamado Automated Requirements Tractability System (ARTS). ARTS teve seu uso limitado desde junho de 1980 e disponibilizada em outubro de 1980. ARTS é uma ferramenta de contabilidade que opera em uma base de dados que consiste em requisitos de sistema e seus atributos. Ela fornece rastreabilidade bidirecional, bem como gestão de base de dados e operações de saída em quaisquer atributos relacionadas a requisitos selecionados pelo usuário.  <b>Disponível em:</b> Não disponível para download.
<b>Principais Referências</b>	

- RAMAMOORTHY, C. V.; USUDA, Y.; PRAKASH, A.; TSAI, W. T. **The evolution support environment system**. IEEE Transactions on Software Engineering. November 1900.
- HOWARD, S. G. **Requirements and traceability management**. In Colloquium on Tools and Techniques for Maintaining Traceability During Design, Savoy Place, London WCR OBL U.K., 2 December 1991.
- DORFMAN, M.; FLYNN, R. F. **ARTS - an automated requirements traceability system**. Journal of Systems and Software, 4(1):63–74, 1984.

**Fonte: Elaborado pelo autor.**

#### 4.1.2.7. Feature Oriented Requirements Traceability (FORT)

A técnica denominada Rastreabilidade de Requisitos Orientada a Feature visa reduzir a dificuldade em gerir os links de rastreabilidade, identificando-os através de requisitos priorizados e constantemente considerando o custo e os esforços (AHN; CHONG, 2006). Ahn e Chong (2006) ainda definem que o processo FORT consiste em cinco fases:

1. Em primeiro lugar, definição de requisitos, que por sua vez, consiste em três atividades: análise da especificação de requisitos, Identificação de requisitos atômicos e atribuição de um identificador para cada requisito. O objetivo da fase de definição de requisitos é normalizar os requisitos do usuário para mapeá-los para vários artefatos. A fim de alcançar isso, a especificação de requisitos é analisada e requisitos atômicos são identificados. Um identificador exclusivo é então associado a cada requisito. O resultado desta fase é uma lista de requisitos com identificadores;
2. O próximo passo é a modelagem de features, que consiste em três atividades: Identificação de Categorias e features, organização de diagramas de features e atribuição de relacionamentos entre requisitos e features. Nesta fase, as relações entre as features também são levadas em consideração. Finalmente todos os requisitos são atribuídos a cada feature. O resultado desta fase é um diagrama de feature e uma lista de features;
3. Em terceiro lugar é a priorização das features, que consiste em duas atividades: estimar valores de requisitos e ordenar a lista de features. Na fase priorização de feature os stakeholders devem estimar os requisitos com base

no valor, risco e esforço para cada requisito. Em seguida, as features são priorizadas. FORT fornece uma escala para a priorização de features;

4. Em quarto lugar, é a fase da geração de links entre os requisitos. Consiste em três atividades: atribuição dos artefatos para a feature relacionada, quebrando elementos de implementação em diferentes níveis e estabelecendo links de rastreabilidade. Em seguida, os itens de implementação são divididos por níveis de granularidade e os links de rastreabilidade são estabelecidos. Esta fase resulta em uma lista de links de rastreabilidade;
5. Finalmente, tem-se a fase de avaliação de links de rastreabilidade, que consiste em duas atividades: a utilização real de links de rastreabilidade durante o desenvolvimento e aperfeiçoamento dos links de rastreabilidade. Com base nesta avaliação, os links de rastreabilidade podem ser mudados, removidos ou adicionados.

O quadro 4.13 apresenta a catalogação desta técnica.

**Quadro 4.13 – Informações gerais sobre a técnica FORT**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiamotômica	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Ahn e Chong (2006) fornecem um exemplo de implementação desta técnica apresentando uma evidência empírica relacionada à utilização de FORT. Neste contexto foi aplicado um estudo de caso de um sistema de aluguel de automóveis, contendo 9 componentes, 49 classes e 152 métodos. O resultado do estudo de caso indica que FORT reduz esforços para gerar links de rastreabilidade em 24-72%. Em suma, FORT fornece variabilidade de informações pelo uso de modelagem de features. Esta variabilidade é útil para estimar o impacto das mudanças de requisitos. FORT também reduz as esforço para criar links de rastreabilidade por priorizar os features e, além disso, fornece uma relação estreita entre os requisitos e os artefatos.

Não foi encontrada nenhuma ferramenta que implemente esta técnica na literatura, sendo assim, o Quadro 4.14 apresenta as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.14 – Principais Referências sobre a técnica FORT

<b>Principais Referências</b>
<ul style="list-style-type: none"> <li>• AHN, S.; CHONG, K. <b>A feature-oriented requirements tracing method: A study of cost-benefit analysis</b>. In Proc. of 2006 International Conference on Hybrid Information Technology, Washington, DC, USA, 2006, IEEE Computer Society, pp. 611-616. 2006.</li> <li>• TORKAR, R.; GORSCHKEK, T.; FELDT, R.; SVAHNBERG, M.; RAJA, U. A.; KAMRAN, K. <b>Requirements Traceability: a Systematic Review and Industry Case Study</b>. International Journal of Software Engineering and Knowledge Engineering, 22(03), 385–433. doi:10.1142/S021819401250009X. 2012.</li> <li>• ZIFTCI, C.; KRUEGER, I. <b>Tracing requirements to tests with high precision and recall</b>. 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), 472–475. doi:10.1109/ASE.2011.6100102.</li> </ul>

**Fonte: Elaborado pelo autor.**

#### 4.1.2.8. Goal-centric traceability (GCT)

Esta técnica foi definida por Cleland et al. (2005) para gerir o impacto da mudança sobre os requisitos não funcionais de um sistema de software, utilizando um Softgoal Interdependence Graphs (SIGs) - a partir de Framework NFR (pré-estabelecido) - para monitorar e rastrear o impacto das mudanças no modelo de ciclo de vida do software. Esta técnica estabelece links de rastreabilidade entre os requisitos funcionais ou não funcionais e os diagramas de classe UML com base em um modelo de rede probabilística. GCT centra-se em requisitos de qualidade especificados como requisitos não funcionais (NFR). Eles representam uma grande variedade de atributos como performance, confiabilidade, segurança, segurança e usabilidade, tudo o que pode ser um desafio para implementar e manter corretamente em um sistema de software.

GCT fornece um framework para ligar um conjunto de QAMs (Quality Assessment Models – Modelos de Avaliação de Qualidade) cuidadosamente colocadas em uma hierarquia de metas definidas para avaliar o sistema. Quando um evento de análise de impacto ocorre, GCT identifica um conjunto inicial de requisitos impactados e, em seguida, utilizando QAMs, reavalia cada requisito visitado a fim de avaliar tanto o impacto direto e indireto da mudança.

GCT é executado através das quatro fases distintas: a modelagem de metas, a detecção de impacto, a análise de meta e a tomada de decisão. A modelagem de metas ocorre principalmente durante as fases de elicitação, especificação e projeto de arquitetura do sistema, é nessa fase que as metas são modeladas no SIG. Durante a fase de detecção de impacto, links de rastreabilidade são estabelecidos entre o modelo funcional do sistema e um conjunto de elementos potencialmente impactados do SIG. Quando ocorre uma alteração em requisitos não-funcionais, um algoritmo de recuperação probabilística retorna dinamicamente ligações entre as classes impactadas e elementos no SIG. Na fase de análise de metas para cada elemento impactado, as mudanças são propagadas em todo o SIG para identificar as metas potencialmente impactadas. Na fase de tomada de decisão, os stakeholders examinam o relatório de impacto e avaliam o impacto da mudança proposta sobre metas de requisitos não-funcionais e gerenciam riscos. Em seguida, determinam se as mudanças devem ser implementadas ou não. Na pesquisa de Grechanik et al. (2007), é afirmado que GCT é projetado para trabalhar exclusivamente com requisitos não funcionais e diagramas de classe. O Quadro 4.15 apresenta a catalogação desta técnica.

**Quadro 4.15 – Informações gerais sobre a técnica GCT**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiautomática	Requisitos e Arquitetura
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Um exemplo da indústria de base foi realizado por Cleland-Huang (2008) para verificar esta técnica. Os resultados experimentais revelam que essa abordagem é bem sucedida para o gerenciamento de rastreabilidade em Requisitos não funcionais.

Esses resultados foram baseados em um estudo de caso em um Sistema de quebra de gelo. Esse sistema consistia em 180 requisitos funcionais que ocasionaram em 19 casos de uso e 75 classes agrupadas em 16 pacotes.

Para esta técnica não foi encontrada nenhuma ferramenta que a implemente. Neste cenário, o Quadro 4.16 apresenta as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.16 – Principais Referências sobre a técnica GCT

Principais Referências
<ul style="list-style-type: none"> <li>• CLELAND-HUANG, J.; MARRERO, W.; BERENBACH, B. <b>Goal-Centric Traceability: Using Virtual Plumblines to Maintain Critical Systemic Qualities</b>. Software Engineering, IEEE Transactions on, 34(5), 685–699. 2008.</li> <li>• DERMEVAL, D. et al. <b>On the Use of Metamodeling for Relating Requirements and Architectural Design Decisions</b>. 1278–1283. 2013.</li> <li>• RAJA, U. A.; KAMRAN, K. <b>Framework for Requirements Traceability</b>. April, 2008.</li> <li>• GALVAO, I.; GOKNIL, A. <b>Survey of Traceability Approaches in Model-Driven Engineering</b>. EDOC 2007. 11th IEEE International Enterprise Distributed Object Computing Conference, 313–313. doi:10.1109/EDOC.2007.42. 2007.</li> <li>• GRECHANIK, M.; MCKINLEY, K. S.; PERRY, D. E. <b>Recovering and using use-case-diagram-to-source-code traceability links</b>. Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering - ESEC-FSE '07, 95. doi:10.1145/1287624.1287640. 2007.</li> <li>• CLELAND-HUANG, J. et al. <b>Goal-centric traceability for managing non-functional requirements</b>. Proceedings of the 27th International Conference on Software Engineering - ICSE '05, 362. doi:10.1145/1062455.1062525. 2005.</li> </ul>

Fonte: Elaborado pelo autor.

#### 4.1.2.9. Graphs

A utilização de grafos é considerada uma técnica de visualização de rastreabilidade, pois a mesma utiliza-se de outras técnicas de geração de links de rastreabilidade para apresentar como a rastreabilidade dá-se para determinados artefatos. Esta técnica faz-se presente neste catálogo por ser considerada uma técnica em diversos trabalhos, tais como, os trabalhos de Voytek et al. (2011), Heim et al. (2008), Merten et al. (2011).

Para mostrar as relações de rastreabilidade, os artefatos podem ser representados como nós de um grafo, e os relacionamentos de rastreabilidade como suas arestas. A visualização de rastreabilidade por meio de grafos pode necessitar de menos espaço para representar informações globais sobre as relações de rastreabilidade. A principal desvantagem do uso de grafos é que eles são menos intuitivos do que referências cruzadas, matriz e exibição de árvore, por exemplo.

O Quadro 4.17 apresenta a catalogação desta técnica.



Quadro 4.17 – Informações gerais sobre a técnica Graphs

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Visualização	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.18 apresenta ferramentas que implementam esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.18 – Ferramentas e Principais Referências sobre a técnica Graphs

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>TOOR</b>	Nesta ferramenta as relações de rastreabilidade são definidas e derivadas em termos de axiomas. Com base nesses axiomas, a ferramenta permite a identificação automática de relações de rastreabilidade entre os requisitos, design e as especificações do código. TOOR apresenta a rastreabilidade por meio de grafos.  <b>Disponível em:</b> <a href="https://cseweb.ucsd.edu/~goguen/sys/toor.html">https://cseweb.ucsd.edu/~goguen/sys/toor.html</a>
<b>TraceLab</b>	TraceLab fornece um ambiente experimental no qual os pesquisadores podem compor experimentos a partir de uma combinação de componentes existentes e definidos pelo usuário, utilizar conjuntos de dados disponíveis publicamente, trocar de componentes com colaboradores, e comparativamente avaliar os resultados em função de parâmetros anteriores.  <b>Disponível em:</b> <a href="http://www.coest.org/index.php/tracelab">http://www.coest.org/index.php/tracelab</a>
<b>OSRMT</b>	Sigla para “Open Source Requirements Management Tool” (Ferramenta de código aberto para gerência de requisitos), licenciada sob os termos da GPL (General Public License), é uma ferramenta desenvolvida na linguagem Java, projetada para apoiar o processo de gerência de requisitos, e fornece apoio à rastreabilidade de requisitos por meio da geração de grafos.  <b>Disponível em:</b> <a href="http://sourceforge.net/projects/osrmt/">http://sourceforge.net/projects/osrmt/</a>
<b>Principais Referências</b>	



- VOYTEK, J.; NÚÑEZ, J. **Visualizing Non-Functional Traces in Student Projects in Information System and Service Design**. In proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011.
- SCHWARZ, H.; EBERT, J.; WINTER, A. **Graph-based traceability: a comprehensive approach**. *Software & Systems Modeling*. 9(4), 473–492. doi:10.1007/s10270-009-0141-4. 2009.
- POHL, K. **Process-Centered Requirements Engineering**. Wiley, New York 1996. ISBN 978-0-863-80193-8. PROTART, 1996.
- HEIM, P.; LOHMANN, S.; LAUENROTH, K.; ZIEGLER, J. **Graph based visualization of requirements relationships**. In Proceedings of Requirements Engineering Visualization, REV '08, pages 51-55, Washington, DC, USA, 2008.
- MERTEN, T; JUEPPNER, D; DELATER, A. **Improved Representation of Traceability Links in Requirements Engineering Knowledge using Sunburst and Netmap Visualizations**. In Proceedings of the 4th International Workshop on Managing Requirements Knowledge, Italy, August 30, 2011.
- PINHEIRO, F.; GOGUEN, J. **An object-oriented tool for tracing requirements**. *IEEE Software*, v. 13, n. 2, p. 796–810, 1996.

Fonte: Elaborado pelo autor.

#### 4.1.2.10. Hyperlink

Munson e Nguyen (2005) afirmam que a utilização de Hyperlink favorece a rastreabilidade, à medida que permitem o acesso rápido às informações envolvidas e rastreadas. Sendo assim, o usuário pode clicar sobre os requisitos apresentados e serão exibidos os detalhes desse requisito. Essa tecnologia de documentos hipermídia é tida como uma boa base de apoio à rastreabilidade em ambientes de desenvolvimento.

O Quadro 4.19 apresenta a catalogação desta técnica.

**Quadro 4.19 – Informações gerais sobre a técnica Hyperlinks**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Visualização	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

Fonte: Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.20 apresenta a ferramenta que implementa esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.20 – Ferramentas e Principais Referências sobre a técnica Hyperlinks**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>COCAR</b>	<p>O ambiente COCAR visa abranger diversas etapas do processo de desenvolvimento de software, dando suporte à realização de algumas atividades, tendo como ponto em comum o modelo de casos de uso. Resumidamente, inserem-se os requisitos do sistema para o qual se desejam realizar as atividades do processo de desenvolvimento e, com base neles, o ambiente permite construir o modelo de casos de uso, os pontos de casos de uso, controles de gerenciamento de requisitos, casos de teste baseados em casos de uso, etc. É interessante notar que, quando a rastreabilidade é apresentada no ambiente COCAR, tanto para os Requisitos Funcionais como para os Casos de Uso, é possível o acesso aos detalhes de ambos por meio de hyperlinks. O uso de hyperlinks e hiperdocumentos para as ferramentas de rastreabilidade de requisitos é sugerido por Zisman e Spanoudakis (2015) e Munson e Nguyen (2005).</p> <p><b>Disponível em:</b> <a href="http://lapes.dc.ufscar.br/tools/cocar">http://lapes.dc.ufscar.br/tools/cocar</a></p>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• THOMMAZO, A.; MARTINS, M. D.; FABBRI, S. <b>O Gerenciamento de Requisitos no Ambiente COCAR</b>. Workshop de Engenharia de Requisitos. 2007.</li> <li>• ZISMAN, A.; SPANOUDAKIS, G. <b>Software Traceability: Past, Present, and Future</b>. The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society – URL : <a href="http://www.resg.org.uk/archive/rq33.pdf">http://www.resg.org.uk/archive/rq33.pdf</a>. Acesso em: 05/03/2015</li> <li>• EBNER, G.; KAINDL, H. <b>Tracing All Around in Reengineering</b>. IEEE Softw., 19(3):70–77, 2002.</li> <li>• MUNSON, E.; NGUYEN, T. <b>Concordance, conformance, versions, and Traceability</b>. Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering, Long Beach, California, 2005.</li> </ul>	

**Fonte:** Elaborado pelo autor.

#### 4.1.2.11.Key Phrases (KP)

A utilização de palavras-chave pode ser considerada uma técnica de apoio à rastreabilidade de requisitos. KP pode extrair frases-chave de comentários de código estreitamente relacionados às classes. Ela pode ajudar a encontrar palavras alternativas para o nome da classe ou palavras que indicam quais as tarefas que a classe cumpre.

Quando os comentários nas classes estão bem documentados, KP pode extrair todas possíveis frases-chave que resumem o propósito de cada uma delas.

Key phrases ainda fornece um breve resumo do conteúdo de um documento. Em grandes coleções de documentos, tais como, bibliotecas digitais, sua utilização tem se tornado comum. Palavras-chave e frases-chave são particularmente úteis porque podem ser interpretadas individualmente e independentemente uma da outra. Elas podem ser usadas tanto em sistemas de recuperação de informação, quanto em descrições dos documentos devolvidos pela consulta. Além disso, pode ser utilizada como a base para os índices de pesquisa, como uma forma de alinhamento de uma coleção, e como uma técnica de agrupamento de documentos.

Key phrases podem ajudar os usuários a ter uma ideia sensata para o conteúdo de uma coleção, pode fornecer pontos de entrada para isso, mostrar como as consultas podem ser estendidas. Elas são geralmente escolhidas manualmente. O uso de uma lista de palavras-chave pode melhorar o recall das relações geradas (menos relações perdidas), mas diminui a sua precisão (ou seja, gera relações mais irrelevantes), quando comparado às técnicas vetoriais clássicas de IR. No entanto, a utilização de um dicionário de sinônimos supera em termos de recall e, por vezes, também em termos de precisão. O Quadro 4.21 apresenta a catalogação desta técnica.

**Quadro 4.21 – Informações gerais sobre a técnica KP**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pré-rastreabilidade e Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação e nenhuma ferramenta que a implemente. Neste cenário, o Quadro 4.22 apresenta as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.22 – Ferramentas e Principais Referências sobre a técnica KP**

<b>Principais Referências</b>
<ul style="list-style-type: none"> <li>• JACKSON, J. <b>A Keyphrase Based Traceability Scheme</b>. In: Proceedings of the IEE Colloquium on Tools and Techniques for Maintaining Traceability During Design, pp. 2/1–2/4, 1991.</li> </ul>

- HAYES, J.H.; DEKHTYAR, A.; OSBORNE, J. **Improving Requirements Tracing via Information Retrieval**. In proceedings of the 11th IEEE International Requirements Engineering Conference, Monterey Bay, 2003.

Fonte: Elaborado pelo autor.

#### 4.1.2.12. Latent Semantic Indexing (LSI)

A Indexação Semântica Latente (do inglês Latent Semantic Indexing - LSI) consiste em uma técnica automática que analisa as co-ocorrências de termos em bases textuais, buscando encontrar relacionamentos latentes entre termos em documentos distintos, definida inicialmente por Deerwester et al. (1990). Esse método assume que existe uma estrutura semântica oculta (latente), subjacente aos dados, na utilização dos termos nos documentos. Tal estrutura é parcialmente obscurecida pela aleatoriedade da escolha de termos. Ou seja, o fato de se escolher uma palavra individual afeta parcialmente na semântica da recuperação. Essa escolha ignora a correlação entre as palavras no documento e a LSI busca preservar essa semântica em sua recuperação e indexação.

A pesquisa de De Lucia et al. (2004) aplica LSI para recuperar links de rastreabilidade entre diferentes tipos de artefatos. Posteriormente, desenvolve a ferramenta de manutenção de rastreabilidade chamada ADAMS e incorpora a técnica na ferramenta desenvolvida (DE LUCIA et al., 2005).

A abordagem proposta por Malec e Marcus (2003) baseia-se no uso de indexação semântica latente (LSI) para apoiar a geração de links de rastreabilidade entre a documentação do sistema (por exemplo, manual, requisitos, design, conjuntos de testes) e o código-fonte. Esta abordagem não depende da linguagem de especificação usada para produzir a documentação de um sistema e da linguagem de programação utilizada no código-fonte. Ela leva em consideração termos sinônimos, uma vez que utiliza combinações lineares de termos como dimensões do espaço de representação. Um link de rastreabilidade entre dois documentos é estabelecido quando a medida de similaridade semântica destes documentos é superior a um determinado limiar.

LSI começa com uma matriz de termos por documentos. Posteriormente, ela usa decomposição singular do valor (SVD) para derivar um modelo particular de

estrutura semântica latente a partir da matriz termo-a-documentos. Uma vez que todos os documentos foram representados no subespaço, pode-se calcular as semelhanças entre os documentos. Toma-se o cosseno entre as suas representações de vetores correspondentes para calcular essa semelhança métrica. Esta métrica tem um valor entre [-1, 1]. Um valor de 1 indica que dois documentos são (quase) idênticos. Estas medidas podem ser usados para agrupar documentos semelhantes, ou para a identificação de vínculos de rastreabilidade entre os documentos.

Segundo Eyal-Salman et al. (2012), os passos para a implementação da técnica LSI são os seguinte:

- A construção de uma matriz th termo-documento  $[i, j]$ , onde cada elemento refere-se à associação entre o termo  $i$ th e  $j$ th documento. Termos representam todas as palavras extraídas das classes e apresentam sua descrição, enquanto documentos representam nomes das classes.
- Em seguida, deve-se dividir a matriz termo-documento em um subespaço LSI, aplicando SVD. SVD é realizada na matriz para determinar os padrões nas relações entre os termos;
- Logo depois é realizado o cálculo da similaridade do cosseno no subespaço LSI;
- Os resultados serão de acordo com um determinado limite, e, em seguida, os vínculos de rastreabilidade entre os artefatos e código-fonte serão estabelecidos.

O Quadro 4.23 apresenta a catalogação desta técnica.

**Quadro 4.23 – Informações gerais sobre a técnica LSI**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pré-rastreabilidade e Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.24 apresenta ferramentas que implementam esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.24 – Ferramentas e Principais Referências sobre a técnica LSI

Ferramentas Disponíveis	
Nome	Descrição
<b>ADAMS</b>	<p>ADAMS - ADvanced Artefact Management System é uma ferramenta web que integra características de gerenciamento de projetos, alocação de recursos, gerenciamento de artefatos, coordenação e cooperação dos colaboradores, versionamento de artefatos e rastreabilidade. A ferramenta ADAMS conta ainda com um módulo de recuperação de rastreabilidade, implementada por meio de uma técnica de recuperação de informação (IR) chamada de Latent Semantic Indexing (LSI), que permite identificar links de rastreabilidade horizontal por meio da similaridade entre termos. Existem plugins para a ferramenta ADAMS que recuperam a informação a partir de requisitos textuais, diagramas UML, e arquivos contendo código-fonte (este último dentro do ambiente eclipse).</p> <p><b>Disponível em:</b> não disponível para download.</p>
<b>TraceViz</b>	<p>Esta ferramenta foi implementada para a visualização de rastreabilidade. Porém técnicas de IR também foram utilizadas para melhorar a qualidade dos requisitos estabelecidos. Park et al. (2000) utilizam medidas de similaridade calculadas para melhorar a qualidade das especificações de requisitos com o auxílio de LSI.</p> <p><b>Disponível em:</b> <a href="http://www.filewatcher.com/m/traceviz.rb.1276-0.html">http://www.filewatcher.com/m/traceviz.rb.1276-0.html</a></p>
<b>RETRO</b>	<p>Ferramenta desenvolvida por Huffman Hayes et al. (2007), que apoia a rastreabilidade de artefatos textuais da engenharia de software. A ferramenta gera Matrizes de Rastreabilidade, utilizando técnicas da área de Recuperação de Informação (IR), como é o caso de LSI.</p> <p><b>Disponível em:</b> <a href="http://opensource.gsfc.nasa.gov/projects/RETRO/">http://opensource.gsfc.nasa.gov/projects/RETRO/</a></p>
Principais Referências	
<ul style="list-style-type: none"> <li>• EYAL-SALMAN, H.; SERIAI, A.-D.; DONY, C.; AL-MSIE'DEEN, R. <b>Recovering traceability links between feature models and source code of product variants.</b> Proceedings of the VARIability for You Workshop on Variability Modeling Made Useful for Everyone - VARY '12, 21–25. doi:10.1145/2425415.2425420. 2012.</li> <li>• LORMANS, M.; DEURSEN, A. <b>Can LSI help reconstructing requirements traceability in design and test?</b> Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 10 pp.–56. doi:10.1109/CSMR.2006.13. 2006.</li> <li>• DEERWESTER, S. et al. <b>Indexing by latent semantic analysis.</b> Journal of the American Society for Information Science, 41(6):391–407, 1990.</li> <li>• DE LUCIA, A.; FASANO, F.; OLIVETO, R.; TORTORA, G. <b>Enhancing an artefact management system with traceability recovery features.</b> In 20th IEEE Int. Conf. on Software Maintenance, pages 306 – 315. IEEE, 2004.</li> <li>• DE LUCIA, A.; FASANO, F.; OLIVETO, R.; TORTORA, G. <b>Adams re-trace: A traceability recovery tool.</b> In Proc. of the 9th European Conf. on</li> </ul>	



Software Maintenance and Reengineering, pages 32–41. IEEE Computer Society, March 2005.

- MALETIC, J. et al. **Using a hypertext model for traceability link conformance analysis**. In 2nd Workshop on Traceability in Emerging Forms of Software Engineering, pages 47–54, Montreal, Canada, 2003. TEFSE 2003.
- MARCUS A., MALETIC J.I. **Recovering Documentation-to-Source-Code Traceability Links using Latent Semantic Indexing**. ICSE, 2003.
- PARK, S.; KIM, H.; KO, Y.; SEO, J. **Implementation of an efficient requirements-analysis supporting system using similarity measure techniques**. Inf Softw Technol 42(6):429–438. 2000.

Fonte: Elaborado pelo autor.

#### 4.1.2.13. Natural Language Processing (NLP)

Segundo Deeptimahanti e Sanyal (2008), o uso de NLP na engenharia de requisitos não tem o objetivo de compreensão do texto em si, mas sim a função de extrair conceitos contidos no desenvolvimento dos requisitos. O trabalho de Leal et al. (2008) apresenta que o NLP pode ser feito em vários níveis. A figura 4.2 apresenta estes níveis para processamento de textos com informações não estruturadas, bem como com seus respectivos objetos de análise e o resultado do processamento. Tendo em vista que a maioria das especificações de requisitos de software é escrita em linguagem natural, e a partir destes documentos de especificação são gerados os demais artefatos para o desenvolvimento do software, vê-se de grande valia a utilização desta técnica para o apoio da atividade de rastreabilidade dos requisitos.

Figura 4.2 – Níveis de análise de NLP em artefatos textuais (LEAL et al., 2008)

Tipo	Objeto de Análise	Resultado
Morfológica	Palavras e variantes	Termos nos documentos
Pragmática	Sentidos de Contexto	Intenção das unidades textuais (Contexto de unidades)
Semântica	Relacionamento das palavras com o universo de conhecimento	Conceitos e relacionamentos
Estatística	Recorrência de termos	Nível de relacionamento entre os termos
Sintática	Ordem das palavras	Relacionamento entre os termos

O processo para o uso de NLP, segundo Silva e Martins (2013), é descrito da seguinte forma: os textos de entrada devem estar no padrão ASCII, para que o etiquetador de texto possa gerar um arquivo similar com as marcações necessárias nas



palavras; os textos em linguagem natural devem ser ajustados no formato exigido pelo etiquetador (esse formato requer que cada palavra, símbolo e pontuação do texto estejam separados por um espaço em branco); o etiquetador de texto realiza a etiquetagem morfosintática no texto de entrada já formatado; essa etiquetagem é feita com base na definição de um conjunto finito de etiquetas (tags) e essas etiquetas devem ter significados linguísticos associados a elas; o artefato de saída do etiquetador é um arquivo no padrão ASCII, contendo o texto de entrada devidamente etiquetado. O Quadro 4.25 apresenta a catalogação desta técnica.

**Quadro 4.25 – Informações gerais sobre a técnica NLP**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pré-rastreabilidade e Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.26 apresenta ferramentas que implementam esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.26 – Ferramentas e Principais Referências sobre a técnica NLP**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>EA-Tracer</b>	Em EA-Tracer, usa-se uma ferramenta de mineração de dados chamada EA-Miner. A ferramenta usa um processador de linguagem natural para identificar as principais características do texto por meio da técnica de NLP.  <b>Disponível em:</b> <a href="http://oro.open.ac.uk/33836/">http://oro.open.ac.uk/33836/</a>
<b>MUPRET</b>	A MUltiPerspective REquirements Traceability (MUPRET) é uma ferramenta para gerar automaticamente relacionamentos de rastreabilidade. MUPRET usa ontologia para ajudar a classificar conceitos e obter relações de rastreabilidade. Ela combina a técnica de processamento de linguagem natural (NLP) e abordagens baseadas em regras para analisar artefatos de requisitos e inferir relações de rastreabilidade, respectivamente.  <b>Disponível em:</b> não disponível para download.
<b>PARADIGMA</b>	Esta ferramenta utiliza recursos do NLP em conjunto com padrões linguísticos identificados nos textos de uma especificação de

	requisitos, permitindo a geração automatizada de modelos conceituais de classes utilizando a notação da UML.  <b>Disponível em:</b> não disponível para download.
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• SILVA, W. C.; MARTINS, L. E. G. <b>Paradigma: uma ferramenta de apoio à elicitação e modelagem de requisitos baseada em processamento de linguagem natural</b>. In: Workshop on Requiriments Engineering, 11th, 2008, Barcelona. Proceedings... Barcelona: [S.n.], 2008.</li> <li>• DEVA KUMAR, D.; SANYAL, R. <b>Static UML Model Generator from Analysis of Requirements (SUGAR)</b>. Advanced Software Engineering and Its Applications, 2008. ASEA 2008 , vol., no., pp.77,84, 13-15 Dec. 2008</li> <li>• LEAL, M., FIGUEIREDO, M. C.; DE SOUZA, C. R. B. <b>Uma abordagem semi-automática para a manutenção de links de rastreabilidade</b>. In: 11th Workshop on Requirements Engineering, 2008, Barcelona, ES. Proceedings of the 11th Workshop on Requirements Engineering, p. 47-58. 2008.</li> <li>• SARDINHA, A.; NIU, N.; YU, Y.; RASHID, A. <b>EA-Tracer : Identifying Traceability Links between Code Aspects and Early Aspects</b>. Proc. of the 27th ACM Symposium on Applied Computing (ACM SAC'12), Riva Del Garda, Italy, 1035–1042. 2012.</li> <li>• ASSAWAMEKIN, N.; SUNETNANTA, T.; PLUEMPITIWIRIYAWAJ, C. <b>MUPRET: An Ontology-Driven Traceability Tool for Multiperspective Requirements Artifacts</b>. 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on Computer and Information Science, 943–948. doi:10.1109/ICIS.2009.55. 2009.</li> </ul>	

**Fonte:** Elaborado pelo autor.

#### 4.1.2.14. Probabilistic Network Model (PNM)

O Modelo de Rastreabilidade probabilístico define uma relação entre artefatos. Esta relação pode ser representada por um modelo de recuperação de informação booleano, ou seja, um modelo que utiliza como técnica de descoberta de informações de rastreabilidade modelos de redes probabilísticas. Com este modelo, é possível identificar se um artefato é relevante para a busca fazendo uso de álgebra booleana. Como consequência, todos os artefatos que estiverem vinculados a um conceito serão recuperados por esta busca. O modelo booleano não fornece os meios para quantificar a influência de determinado grupo de conceitos, ou suas propriedades, no código da aplicação.

Os modelos de redes probabilísticas utilizam redes bayesianas de crença ou modelos semelhantes para calcular a probabilidade de uma alteração que afete outras

entidades do programa. Eles são combinados com algoritmos gulosos ou sistemas lineares ou não-lineares de equações para melhorar o cálculo de probabilidades.

Nesse tipo de técnica, cada artefato é representado por uma distribuição probabilística, onde pode ser formada uma matriz com termos normalizados. Uma vez que os artefatos são representados como distribuição de probabilidade, é calculada a distância entre distribuição de probabilidade de dois artefatos e uma lista de classificação dos links é retornada. A partir de então, os documentos de destino são classificados por meio da "distância" de suas distribuições de probabilidade com relação aos documentos de origem. O Quadro 4.27 apresenta a catalogação desta técnica.

**Quadro 4.27 – Informações gerais sobre a técnica PNM**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.28 apresenta ferramentas que implementam esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.28 – Ferramentas e Principais Referências sobre a técnica PNM**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>RETRO</b>	RETRO - REquirements TRacing On-target é uma ferramenta desenvolvida por Huffman Hayes et al. (2007), que apoia a rastreabilidade de artefatos textuais da engenharia de software. A ferramenta gera Matrizes de Rastreabilidade utilizando técnicas da área de Recuperação de Informação (IR), como é o caso dos modelos de redes probabilísticos.  <b>Disponível em:</b> <a href="http://opensource.gsfc.nasa.gov/projects/RETRO/">http://opensource.gsfc.nasa.gov/projects/RETRO/</a>

<b>Poirot</b>	<p>Poirot foi projetado para dar suporte ao usuário a tarefas relacionadas à rastreabilidade da forma mais eficiente e eficaz possível. Ela utiliza um modelo de rede probabilística para gerar dinamicamente links candidatos de rastreabilidade entre artefatos.</p> <p><b>Disponível em:</b> não disponível para download.</p>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• GIBIEC, M.; CZAUDERNA, A.; Cleland-Huang, J. <b>Towards mining replacement queries for hard-to-retrieve traces</b>. Proceedings of the IEEE/ACM International Conference on Automated Software Engineering - ASE '10, 245. doi:10.1145/1858996.1859046. 2010.</li> <li>• LIN, J. et al. <b>Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability</b>. 14th IEEE International Conference Requirements Engineering, 363 – 364.2006</li> <li>• HAYES, J. et al. <b>REquirements TRacing on target (RETRO): improving software maintenance through traceability recovery</b>. Innovations in Systems and Software Engineering, 3:193–202, 2007.</li> </ul>	

**Fonte: Elaborado pelo autor.**

#### 4.1.2.15. Rules-based Traceability (RBT)

Spanoudakis et al. (2004) propõem um método para criar automaticamente links de rastreabilidade utilizando regras. Eles usam duas regras de rastreabilidade básicas, a primeira definida como uma regra requisitos-para-objeto-modelo de rastreabilidade (regra RTOM) e de regras de rastreabilidade inter-requisitos (regra IREQ).

As regras são implantadas em três tipos de documentos específicos, os documentos de instrução, ou seja, requisitos, casos de uso e os modelos de análise de objeto. Regras RTOM são utilizadas para rastrear requisitos e casos de uso a uma modelo, enquanto regras IREQ são usados para rastrear entre requisitos e casos de uso. O método parte do princípio de que todos os tipos de documentos estão em formato baseado em XML.

As regras de rastreabilidade também estão representadas em uma linguagem de marcação baseada em XML. O método consiste em quatro fases: (i) marcação gramatical dos artefatos; (ii) a conversão dos artefatos marcados em representações XML; (iii) geração das relações de rastreabilidade entre os artefatos; e (iv) geração das relações de rastreabilidade entre diferentes partes dos artefatos. A técnica baseada em

regras abrange documentos de requisitos, documentos de casos de uso e modelos de análise de objetos.

Os documentos a serem rastreados devem estar representados em XML e as relações geradas podem ser representadas como hiperlinks. A abordagem foi avaliada em estudos de caso para uma família de sistemas de software de televisão e por um sistema de gerenciamento de cursos universitários. Os níveis de recall e precisão alcançados por esta abordagem nos estudos de casos relatados são promissores: medidas de recall e precisão variam entre 50% e 95%. Estes resultados fornecem evidência da capacidade da abordagem para apoiar a geração automática das relações de rastreabilidade. O Quadro 4.29 apresenta a catalogação desta técnica.

**Quadro 4.29 – Informações gerais sobre a técnica RBT**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Automática	Requisitos e Arquitetura
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

Fonte: Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.30 apresenta a ferramenta que implementa esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.30 – Ferramentas e Principais Referências sobre a técnica RBT**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>xTraque</b>	Esta ferramenta permite a geração das relações de rastreabilidade através da interpretação das regras de rastreabilidade. Ele também oferece suporte para a criação de novas regras de rastreabilidade e tradução de documentos para o formato XML.  <b>Disponível em:</b> não disponível para download.
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• SPANOUDAKIS, G.; ZISMAN, A.; PÉREZ-MIÑANA, E.; KRAUSE, P.; <b>Rule-based generation of requirements traceability relations</b>. Journal of Systems and Software, 72(2), 105–127. doi:10.1016/S0164-1212(03)00242-5. 2004</li> <li>• MÄDER, P.; GOTEL, O. C. Z.; PHILIPPOW, I. <b>Rule-Based Maintenance of Post-Requirements Traceability Relations</b>. International Requirements Engineering Conference, 2008. RE '08. 16th IEEE, (c), 23–32. doi:10.1109/RE.2008.24. 2008.</li> </ul>	

- JURETA, I. J.; FAULKNER, S. **Tracing the Rationale Behind UML Model Change Through Argumentation**. In Proceedings of the 26th International Conference on Conceptual Modeling (pp. 454–469). Berlin, Heidelberg. 2007.
- JIRAPANTHONG, W.; ZISMAN, A. **XTraQue: traceability for product line systems**. *Software & Systems Modeling*, 8(1), 117–144. doi:10.1007/s10270-007-0066-8. 2007.
- RAMESH, B.; JARKE, M. **Towards Reference Models for Requirements Traceability**, *IEEE Transactions in Software Engineering*, 27(1), 58-93, 2001.

**Fonte: Elaborado pelo autor.**

#### 4.1.2.16.Scenarios Based

Cenários geralmente são utilizados como um meio alternativo para expressar o comportamento de um sistema pelas diversas etapas de um processo de desenvolvimento de software. Cada cenário pode assumir diferentes representações em cada uma dessas etapas. No trabalho de Naslavsky et al. (2005) é apresentado o uso de cenário como técnica de apoio à atividade de rastreabilidade de requisitos. Sendo assim, para estabelecer links de rastreabilidade entre esses cenários e outros artefatos desenvolvidos durante o processo de desenvolvimento de software é necessário compreender o propósito de cada um deles e qual a relação entre eles.

A primeira atividade a ser realizada para a utilização dessa técnica é a de realizar inicialmente uma hipótese, onde alguns links devem ser identificados manualmente a partir de documentação. A segunda atividade é a de atomização, onde os cenários são monitorados por uma ferramenta de cobertura de código e um conjunto de ligações é construído, uma espécie de gráfico. A terceira atividade é a generalização, em que o gráfico é totalmente percorrido. A quarta atividade é o refinamento, em que o gráfico novamente é percorrido em sua totalidade. Estas atividades produzem um conjunto final de links de rastreabilidade.

Cenários normalmente são utilizados para modelar funcionalidades do sistema e para gerar casos de teste funcionais. Cenários baseados em casos de teste criam um mapeamento entre os requisitos e outros artefatos como design e código. Os cenários são criados para rastrear apenas os casos interessantes, portanto, eles não podem fornecer cobertura completa. No entanto, de acordo com Cleland (2005), os cenários são frequentemente utilizados por vários métodos de avaliação de arquitetura como



método de avaliação de trade-off de arquitetura (ATAM), e método de avaliação de arquitetura de software (SAAM).

O Quadro 4.31 apresenta a catalogação desta técnica.

**Quadro 4.31 – Informações gerais sobre a técnica Scenarios Based**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiautomática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.32 apresenta ferramentas que implementam esta técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.32 – Ferramentas e Principais Referências sobre a técnica Scenarios Based**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>STRADA</b>	É uma ferramenta para rastreabilidade baseada em Cenários de detecção e análise. STRADA possui duas grandes capacidades: (1) Captura da rastreabilidade baseada em cenário; (2) Análise do Rastro.  <b>Disponível em:</b> não disponível para download.
<b>TraceAnalyzer</b>	É uma ferramenta que faz uso de casos de teste para gerar informações de rastreamento durante a execução do programa. Utiliza informações geradas a partir de Cenários (ou seja, casos de teste) para gerar links de rastreabilidade entre: (a) cenários e código-fonte; (b) modelo (diagramas de classe, por exemplo) e os elementos de código-fonte; (c) cenários e elementos do modelo; e (d) entre os elementos do modelo.  <b>Disponível em:</b> <a href="http://dominicgiles.com/downloads.html">http://dominicgiles.com/downloads.html</a>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• TANG, Y. J.; HAN, J. A <b>Rationale-based Architecture Model for Design Traceability and Reasoning</b>. Journal of Systems and Software, 80(6):918-934, June 2007.</li> <li>• VIANNA FERREIRA, P. J. A.; BARROS, M. D. O. <b>Traceability between</b></li> </ul>	



**function point and source code.** Proceeding of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering - TEFSE '11, 10. doi:10.1145/1987856.1987860. 2011.

- NASLAVSKY, L.; ALSPAUGH, T. A.; RICHARDSON, D. J.; ZIV, H. **Using scenarios to support traceability.** Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering - TEFSE '05, 25. doi:10.1145/1107656.1107663. 2005.
- EGYED, A.; GRÜNBACHER, P., REY, M.; LINZ, A. **STRADA : A Tool for Scenario-based Feature-to-Code Trace Detection and Analysis.** 2007.
- CLELAND-HUANG, J. **Toward Improved Traceability of Non-Functional Requirements.** Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering TEFSE'05, ACM, 2005, pp. 14-19.

Fonte: Elaborado pelo autor.

#### 4.1.2.17. Templates

A utilização de templates possibilita melhores resultados na análise de informações não estruturadas, já que eles estruturam as informações em áreas pré-definidas, possibilitando uma busca mais eficiente pelos itens e links de rastreabilidade. Eles objetivam fornecer uma estrutura inicial aos documentos, e assim permitir a padronização dos artefatos textuais. Um template é um arcabouço para um determinado tipo de artefato de software e tem o objetivo de padronizar, guiar e facilitar sua construção, e são utilizados em quase todas as atividades do processo de software, desde o planejamento do projeto, passando pela fase de requisitos, até a implementação.

O Quadro 4.33 apresenta a catalogação desta técnica.

**Quadro 4.33 – Informações gerais sobre a técnica Templates**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Manual	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pré-rastreabilidade e Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

Fonte: Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação e nenhuma ferramenta que a implemente. Neste cenário, o Quadro 4.34 apresenta as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.34 – Principais Referências sobre a técnica Template**

<b>Principais Referências</b>
<ul style="list-style-type: none"> <li>• CHEN, X.; GRUNDY, J. <b>Improving automated documentation to code traceability by combining retrieval techniques</b>. 26th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2011.</li> <li>• WITTEN, H. et al. <b>Kea: practical automatic keyphrase extraction</b>. 4th ACM DL, Berkeley, pp. 254-255. 1999.</li> </ul>

**Fonte: Elaborado pelo autor.**

#### 4.1.2.18. Traceability Matrix

No início da utilização de matrizes para a rastreabilidade de requisitos, esta técnica de visualização de links de rastreabilidade deu apoio apenas na representação de pares de artefatos, o que dificultava uma visão mais ampla quando era necessário realizar uma análise de mudança. Depois disso, diversas modificações foram realizadas para melhorar seu desempenho, como cores e símbolos diferentes começaram a ser usados para representar diferentes tipos de relacionamentos. Além disso, informações sobre as relações e artefatos podem ser obtidas, por exemplo, pelo clique do mouse.

Em linhas gerais a implementação de matriz de rastreabilidade para a visualização de rastreabilidade dá-se da seguinte forma: os conjuntos de itens a serem relacionados devem ficar cada um, em uma das dimensões da matriz, ou seja, dimensão horizontal e vertical; e cada relacionamento existente entre eles deve ser assinalado na matriz. As vantagens da utilização de matrizes é que eles são fáceis de entender. No entanto, para projetos em que o número real de relacionamentos é grande, torna-se mais difícil de visualizar relações específicas. Além disso, a matriz não mostra a hierarquia das relações e é difícil para navegar através das relações entre artefatos de forma recursiva.

A representação da matriz é equivalente a uma representação gráfica, mas é menos organizada como uma técnica de apresentação visual, e possui a limitação de apresentar apenas relações binárias entre os itens. O Quadro 4.35 apresenta a catalogação desta técnica.

Quadro 4.35 – Informações gerais sobre a técnica Traceability Matrix

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Visualização	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte:** Elaborado pelo autor.

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.36 apresenta ferramentas que implementam a técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

Quadro 4.36 – Ferramentas e Principais Referências sobre a técnica Traceability Matrix

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>ARTS</b>	É um sistema de banco de dados para a manutenção de links de rastreabilidade. Ele pode produzir tabelas de referência cruzada em vários formatos, incluindo o formato de Matriz de Rastreabilidade.  <b>Disponível em:</b> não disponível para download.
<b>RequisitePro</b>	É uma ferramenta de gerenciamento de requisitos da Rational-IBM. Provê funcionalidades oferecidas relativas ao controle da rastreabilidade.  <b>Disponível em:</b> <a href="http://www.ibm.com/developerworks/br/downloads/r/rrp/">http://www.ibm.com/developerworks/br/downloads/r/rrp/</a>
<b>AnalistPro</b>	É uma ferramenta para gerência de requisitos, rastreabilidade e análise de impacto, que possui um editor para a criação de requisitos e também possibilita a criação de diagramas. No contexto da rastreabilidade, a ferramenta possibilita a ligação entre os requisitos e entre os requisitos e diagramas, gerando então uma matriz de rastreabilidade.  <b>Disponível em:</b> <a href="http://www.analysttool.com/">http://www.analysttool.com/</a>
<b>Enterprise Architect</b>	É uma ferramenta CASE, voltada para a modelagem UML e a gerência de requisitos. Requisitos são armazenados em pacotes. Os relacionamentos entre requisitos ou entre pacotes são visualizados por meio de uma matriz de rastreabilidade.  <b>Disponível em:</b> <a href="http://www.sparxsystems.com.au/products/ea/trial.html">http://www.sparxsystems.com.au/products/ea/trial.html</a>
<b>OSRMT</b>	Sigla para “Open Source Requirements Management Tool”

	<p>(Ferramenta de código aberto para gerência de requisitos), licenciada sob os termos da GPL (General Public License), é uma ferramenta desenvolvida na linguagem Java, projetada para apoiar o processo de gerência de requisitos, e fornece apoio à rastreabilidade de requisitos por meio da geração de matrizes de rastreabilidade.</p> <p><b>Disponível em:</b> <a href="http://sourceforge.net/projects/osrmt/">http://sourceforge.net/projects/osrmt/</a></p>
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• SILVA, W. C.; MARTINS, L. E. G. <b>Paradigma: uma ferramenta de apoio à elicitação e modelagem de requisitos baseada em processamento de linguagem natural.</b> In: Workshop on Requiriments Engineering, 11th, 2008, Barcelona. Proceedings... Barcelona: [S.n.], 2008.</li> <li>• XIE, M.; POSHYVANYK, D. <b>When and How to Visualize Traceability Links.</b> In the Proceedings of the 3rd ACM International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2005) , Long Beach, CA, November 8th, 2005 pp. 56-61.</li> <li>• WINKLER, S.; PILGRIM, J. <b>A survey of traceability in requirements engineering and model-driven development.</b> Software &amp; Systems Modeling, 9(4), 529–565. doi:10.1007/s10270-009-0145-0. 2009.</li> <li>• FILHO, G.; LENCASTRE, M. <b>Towards a Traceability Visualisation Tool.</b> Proceedings - 2012 8th International Conference on the Quality of Information and Communications Technology, 221–223. doi:10.1109/QUATIC.2012.60. 2012.</li> <li>• SHAHID, M.; IBRAHIM, S.; MAHRIN, M. N. <b>An Evaluation of Requirements Management and Traceability Tools.</b> World Academy of Science, Engineering and Technology., 596–601. 2011.</li> </ul>	

**Fonte: Elaborado pelo autor.**

#### 4.1.2.19. Value-Based Requirements Tracing (VBRT)

O objetivo do processo de rastreamento de requisitos baseados em valores é identificar traços com base em requisitos priorizados e, assim, identificar quais são os traços mais importantes e valiosos que os outros. O processo VBRT representa um ciclo de desenvolvimento e aperfeiçoamento do sistema de rastreabilidade baseada em valor. Ele consiste em cinco etapas distintas: (1) a definição dos requisitos; (2) a priorização dos requisitos; (3) o empacotamento dos requisitos; (4) a ligação de artefatos; e (5) a avaliação.

A fundamentação desta técnica está no paradigma da engenharia de software baseada em valor, que defende que o valor agregado por cada requisito, caso de uso e defeito deve ser levado em consideração no processo de desenvolvimento (EGYED et al., 2005). Avalia a relação custo/benefício de se aumentar a qualidade das ligações de

rastreabilidade de acordo com três aspectos-chave: (a) granularidade dos artefatos; (b) valor dos artefatos; e (c) momento de realização da rastreabilidade (no início ou após o projeto). Ao final, sugere que a qualidade das informações de rastreabilidade aumenta em uma proporção inferior ao esforço despendido para prover as informações de entrada. O Quadro 4.37 e apresenta a catalogação desta técnica.

**Quadro 4.37 – Informações gerais sobre a técnica VBRT**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiautomática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Para esta técnica não foi encontrado nenhum exemplo de implementação e nenhuma ferramenta que a implemente. Neste cenário, o Quadro 4.38 apresenta as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.38 – Principais Referências sobre a técnica VBRT**

<b>Principais Referências</b>
<ul style="list-style-type: none"> <li>• HEINDL, M.; BIFFL, S. <b>A case study on value-based requirements tracing</b>. Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering - ESEC/FSE-13, 60. doi:10.1145/1081706.1081717. 2005.</li> <li>• EGYED, A., BIFFL, S., HEINDL, M. et al. <b>Determining the Cost-Quality TradeOff for Automated Software Traceability</b>. In: Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, pp. 360-363, Long Beach, CA, EUA, Nov. 2005.</li> </ul>

**Fonte: Elaborado pelo autor.**

#### 4.1.2.20. Vector Space Model (VSM)

É um modelo algébrico para representar documentos de texto (e quaisquer objetos, em geral) como vetores. Ele é usado na filtragem de informação, recuperação de informação, indexação e rankings de relevância. Esta técnica pode ser dividida em três etapas. A primeira etapa é a indexação de documentos, onde os termos relevantes são extraídos do documento. A segunda etapa é a ponderação dos termos indexados para melhorar a recuperação de informação relevante para o usuário. A última etapa é a

da classificação do documento com respeito à consulta de acordo com uma medida de similaridade.

Sendo assim, a premissa básica de adotar o modelo de espaço vetorial é que os vários objetos de recuperação de informação devem ser modelados como elementos de um espaço vetorial. Especificamente termos, documentos, consultas, conceitos, e assim por diante são todos os vetores em um espaço vetorial.

Em linhas gerais VSM é uma abordagem padrão que calcula a semelhança entre uma consulta e um documento por meio do cálculo do cosseno, onde cada uma das consultas está representada como um vetor de termos ponderados. Esta abordagem tenta reduzir o número de relações de rastreabilidade perdidas e irrelevantes usando uma técnica vetorial clássica de IR e uma técnica vetorial estendida de IR com o uso de listas de palavras-chave ou o uso de enciclopédias.

O Quadro 4.39 apresenta a catalogação desta técnica.

**Quadro 4.39 – Informações gerais sobre a técnica VSM**

<b>Tipo da Técnica</b>	<b>Fases Envolvidas</b>
Semiamatemática	Geral
<b>Dimensão de Rastreabilidade</b>	<b>Estado de Rastreabilidade</b>
Horizontal e Vertical	Pós-rastreabilidade.
<b>Nota</b>	
Não possui nenhuma nota.	

**Fonte: Elaborado pelo autor.**

Para esta técnica não foi encontrado nenhum exemplo de implementação. Neste cenário, o Quadro 4.40 apresenta ferramentas que implementam a técnica e as principais referências de trabalhos encontrados na literatura que tratam da mesma.

**Quadro 4.40 – Ferramentas e Principais Referências sobre a técnica VSM**

<b>Ferramentas Disponíveis</b>	
<b>Nome</b>	<b>Descrição</b>
<b>RETRO</b>	RETRO - REquirements TRacing On-target é uma ferramenta desenvolvida para rastreabilidade de requisitos que pode empregar várias técnicas de IR, incluindo VSM.  <b>Disponível em:</b> <a href="http://opensource.gsfc.nasa.gov/projects/RETRO/">http://opensource.gsfc.nasa.gov/projects/RETRO/</a>
<b>TraceME</b>	É um plugin do Eclipse destinado a fornecer assistência completa para capturar e gerenciar efetivamente links de rastreabilidade durante o desenvolvimento do software. TraceMe suporta a recuperação de rastreabilidade baseados em IR, usando o IR motor Lucene, uma



	implementação VSM.  <b>Disponível em:</b> não disponível para download.
<b>Principais Referências</b>	
<ul style="list-style-type: none"> <li>• ANTONIOL, G.; CANFORA, G.; DE LUCIA, A; CASAZZA, G. <b>Information retrieval models for recovering traceability links between code and documentation.</b> Software Maintenance, IEEE. 2011.</li> <li>• HAYES, J.; DEKHTYAR, A; OSBORNE, J. <b>Improving requirements tracing via information retrieval.</b> In in Proceedings of the International Conference on Requirements Engineering, pages 151–161, 2003.</li> <li>• PANDANABOYANA, S.; SRIDHARAN, S.; YANNELLI, J.; HAYES, J. H. <b>REquirements TRacing On target (RETRO) Enhanced with an Automated Thesaurus Builder.</b> International Workshop on Traceability in Emerging Forms of Software Engineering, 61–67. 2013.</li> <li>• BAVOTA, G. et al. <b>TraceME: Traceability Management in Eclipse.</b> (ICSM), 2012 28th IEEE International Conference on Software Maintenance, 642–645. doi:10.1109/ICSM.2012.6405343. 2012.</li> </ul>	

**Fonte: Elaborado pelo autor.**

#### 4.1.3. Aplicação do Catálogo

Este catálogo apresentou diversas técnicas que auxiliam a atividade de rastreabilidade de requisitos no contexto de projetos de software, podendo ser um instrumento de grande valia tanto para a área acadêmica quanto para a indústria. Para a área acadêmica pode servir como um guia para pesquisas futuras sobre rastreabilidade de requisitos, por exemplo, para o estudo na elaboração de uma nova técnica, na otimização de uma já existente, ou mesmo na combinação de duas ou mais técnicas a fim de melhorar a eficiência e eficácia das mesmas. Já para a indústria pode ser um norteador que auxilie no momento da escolha de qual abordagem de apoio para implementação da rastreabilidade de requisitos que mais se enquadra em diversos contextos.

Nesse sentido, para sua melhor aplicação é necessário que seus leitores tenham certo conhecimento do contexto em que sua organização encontra-se, como por exemplo, forma de trabalho da equipe, processo de desenvolvimento adotado, entre outros fatores. Além do que, qual sua expectativa e/ou limitações para a implantação da rastreabilidade, levando em consideração custos, cronograma, conhecimento técnico, equipe, infraestrutura, cultura e política organizacional.



Sendo assim, escolher uma técnica de apoio, ou até mesmo uma ferramenta em detrimento de outra, é uma tarefa que tem mais a ver com a forma que se deseja trabalhar na organização e a infraestrutura que se tem, já que existem técnicas consideradas simples de serem implementadas, ditas manuais, e com um grau de precisão maior, porém geralmente mais custosas em relação a tempo. E outras consideradas mais complexas, que realizam a rastreabilidade de forma automática ou semiautomática em um tempo menor, porém com baixa precisão na geração dos elos de rastreabilidade quanto comparadas com a primeira.

Para melhor utilização deste catálogo e conseqüentemente a escolha da abordagem que mais se enquadra a uma determinada organização, o leitor deve tomar como referência inicialmente as características de rastreabilidade que deseja controlar durante seu processo de desenvolvimento de software, ou seja, a dimensão de rastreabilidade (horizontal, vertical ou ambas) e o estado de rastreabilidade (pré-rastreabilidade, pós-rastreabilidade ou ambas), em segundo lugar, vale levar em consideração as fases do ciclo de vida de desenvolvimento de software que deseja acompanhar, neste caso, deve verificar se é apenas umas das fases apresentadas neste catálogo ou todas. Em seguida é importante saber o tipo de técnica que deseja implementar para em sua organização, se é uma técnica considerada manual, semiautomática ou automática, levando em consideração as vantagens e desvantagens apontadas para cada uma destas neste catálogo. Por fim, deve se visualizar quais as técnicas que atendem a estes requisitos e dentre estas verificar exemplos de implementação, se possuem ou não ferramenta que as implementem e mesmo aprofundar o conhecimento por meio das principais referências que tratam das mesmas, podendo assim, escolher a técnica mais adequada.

A forma de manuseio do catálogo deve ser pautada inicialmente na busca das definições estabelecidas por seu usuário, caso o mesmo não tenha um conhecimento prévio a respeito de rastreabilidade de requisitos, deve-se buscar um estudo mais aprofundado sobre conceitos que permeiam esta área a fim de definir o que se procura atingir com a rastreabilidade de requisitos.

## **4.2. Avaliação do Catálogo**

Para a avaliação deste catálogo foi utilizado o método de avaliação por especialista, esta técnica faz uso de um ou mais especialistas da área e com afinidade com o tema em questão. Sua escolha justifica-se pelo fato de se ter inicialmente a avaliação de alguém que de fato entenda da área a respeito das informações dispostas neste catálogo, para que se possa realizar o refinamento do mesmo e futuramente colocá-lo em avaliação na indústria.

Neste contexto, um especialista em qualidade de software, doutor em Engenharia de Software, implementador e avaliador do MR-MPS-SW e com experiência na implementação do CMMI-DEV, com ampla pesquisa e atuação na área, foi consultado para evidenciar a avaliação do catálogo. Para nortear que aspectos deveriam ser considerados na avaliação, foram estabelecidos os seguintes critérios objetivos:

- a) As abordagens elencadas no catálogo são coerentes e refletem as práticas relacionadas às atividades de Rastreabilidade de Requisitos;
- b) A descrição das técnicas, tipo, fases, dimensão, estado, contexto de aplicação, exemplo de implantação, ferramentas e suas principais referências são pertinentes.

O questionário da avaliação foi composto por 18 questões objetivas, divididas em 2 grupos: o primeiro diz respeito ao Perfil do Revisor, onde as questões tem o objetivo de descobrir o nível de conhecimento do entrevistado a respeito de gerência de requisitos, rastreabilidade de requisitos, implementação de modelos para melhoria de processo, e métodos de avaliação constantes nos modelos. O segundo grupo trata da Apresentação da Proposta, que tem como intuito verificar o entendimento do avaliador em relação ao trabalho em avaliação, tendo com quesitos o grau de corretude e completude do catálogo e se o mesmo pode ser utilizado como referência no auxílio da implementação da rastreabilidade de requisitos.

Como anexo ao questionário, foi solicitada uma avaliação subjetiva para revisão do material enviado, baseado em (MELLO et al., 2012) e (BRITO NETO, 2014), no qual era permitido o registro de comentários através de uma tabela preenchida pelo avaliador, contendo a identificação do comentário, sua categoria, o item a qual se corresponde (podendo ser relativo a uma fase, tarefa ou em geral da metodologia), o

texto do comentário em si, e uma sugestão com a proposta do revisor para contornar o problema. As categorias definidas para os comentários foram:

- Técnico Alto (TA), indicando que foi encontrado um problema em um item que, se não for alterado, comprometerá as considerações;
- Técnico Baixo (TB), indicando que foi encontrado um problema em um item que seria conveniente alterar;
- Editorial (E), indicando que foi encontrado um erro de português ou que o texto pode ser melhorado;
- Questionamento (Q), indicando que houve dúvidas quanto ao conteúdo das considerações;
- Geral (G), indicando que o comentário é geral em relação às considerações;
- Boas Práticas (BP), indicando que o comentário está relacionado à lista de boas práticas.

O material de avaliação e a metodologia definida foram enviados ao revisor selecionado através de contato por e-mail e após conferência realizada para explicar a metodologia de avaliação, também descrita no documento, foi aguardado o retorno da avaliação realizada.

Os feedbacks recebidos após a avaliação foram bastante proveitosos e favoráveis para o aprimoramento da metodologia e são descritos a seguir, junto ao perfil do avaliador.

Com relação ao perfil do especialista que enviou a avaliação realizada no dia 16 de março de 2015, o próprio revisor considera que possui conhecimento alto em rastreabilidade de requisitos e gerência de requisitos, inclusive implantando esta área do conhecimento em diversas organizações em um tempo de mais de cinco anos, com mais 16 implantações. Além disso, ele considerou que possui alta experiência com métodos de avaliação constantes em modelos para melhoria de processo de software, possuindo certificação como implementador e avaliador MPS.BR e implementador CMMI-DEV, com tempo de experiência de mais de cinco anos tanto em implementação quanto em avaliação de processos de gerência de requisitos.

Relacionado às questões objetivas, o avaliador considerou que o catálogo, à época da revisão, necessitava de alguns ajustes sobre a contextualização das atividades de rastreabilidade de requisitos e solicitou a inclusão de exemplos mais práticos e

detalhados, porém, que o mesmo poderia ser usado como um referencial de apoio para a implantação das atividades de rastreabilidade de requisitos em uma organização.

As observações da avaliação subjetiva renderam três itens relacionados à proposta do catálogo em geral, descritos a seguir.

O comentário considerado mais grave pelo revisor, adequado à categoria TA (Técnico Alto), foi a falta da categorização com relação ao tipo de técnica de rastreabilidade de requisitos, foi então sugerido o acréscimo do tipo de técnica de visualização. Após a análise na literatura, onde diversos autores definem estas formas de visualização como um tipo de técnica de rastreabilidade, a sugestão foi aceita.

Outro comentário está relacionado à generalidade da proposta e com avaliação considerada também TA, que, segundo o revisor, pode dificultar sua aplicação no dia-a-dia da empresa para quem não possui conhecimento aprofundado da área, foi à falta de uma seção que apresentasse a forma de aplicação do catálogo. Para atender à solicitação considerada como pertinente, foi acrescido ao catálogo uma seção para apresentar uma possível aplicação deste a partir de cenários.

Além disso, ainda no contexto geral, foi sugerida a contextualização das técnicas, incluindo mais exemplos práticos e detalhados, o que facilitaria a utilização do catálogo e seu entendimento mesmo para aqueles que não possuíssem um conhecimento aprofundado a respeito de rastreabilidade de requisitos. A solução encontrada para aperfeiçoar o catálogo foi detalhar mais os exemplos práticos encontrados para as técnicas.

Por fim, foi solicitado o acréscimo em cada ferramenta encontrada de um link com o endereço para download para facilitar ainda mais a efetividade na utilização do catálogo. Neste sentido, para cada ferramenta encontrada foi associado o endereço para download, quando encontrado.

### **4.3. Considerações Finais**

Este capítulo apresentou um catálogo de abordagens de apoio à atividade de rastreabilidade de requisitos, tendo com abordagens principais e norteadoras do mesmo, técnicas de rastreabilidade de requisitos e sua aplicação por meio de apoio ferramental. A base para a definição deste catálogo foi a Revisão Sistemática da

Literatura realizada no contexto deste trabalho com o objetivo de encontrar abordagens de apoio para a atividade de rastreabilidade de requisitos que foi apresentada no Capítulo 3. Além disso, foi apresentada a forma de aplicação deste catálogo, ou seja, qual a valia deste instrumento para organizações ou quaisquer interessados na área. Por fim, foi apresentada a avaliação do catálogo feita por um especialista da área e após as sugestões fornecidas e aprimoramento da proposta, este trabalho pode ser utilizado em organizações.

## 5 CONCLUSÕES

Neste capítulo é abordada uma sumarização do trabalho apresentado por meio de suas principais conclusões. Também são apresentadas as principais contribuições à área de gerência de requisitos, mais especificamente à atividade de rastreabilidade de requisitos, algumas oportunidades de melhorias identificadas, assim como trabalhos futuros a serem executados a partir do estudo realizado.

### 5.1. Sumarização dos Resultados

O trabalho realizado focou na investigação de abordagens de apoio à atividade de rastreabilidade de requisitos no contexto de projetos de softwares, atividade esta que está inserida na área de gerenciamento de requisitos em diversos modelos e normas de qualidade. A análise destas abordagens para rastreabilidade de requisitos resultaram na elaboração de um catálogo, apresentado no Capítulo 4.

O catálogo mencionado foi fundamentado por meio da realização de uma revisão sistemática da literatura, que ocorreu a partir da aplicação de um protocolo de revisão sistemática, o qual incluía critérios de inclusão e exclusão de estudos, assim como mais informações pertinentes à condução de uma revisão sistemática. Foram analisados 1485 trabalhos, dos quais 411 foram selecionados.

Dentre os estudos selecionados foram retiradas informações importantes sobre abordagens de apoio à atividade de rastreabilidade de requisitos. Dentre as abordagens podem ser consideradas técnicas, metodologias, processos, frameworks, ferramentas e afins. Nesse sentido o catálogo foi estruturado com base nas técnicas, por ser uma abordagem que é essencial e presente nas demais.

Além das informações obtidas no catálogo, por meio deste trabalho foi possível também uma visão geral sobre a evolução, tendências e o estado atual da rastreabilidade de requisitos em projetos de softwares.

Com base no exposto acima, este trabalho almejou fornecer contribuições para a área de gerenciamento de requisitos no que tange a rastreabilidade de requisitos, apresentando evidências encontradas na literatura, a fim de reunir em um único banco de conhecimento grande parte de abordagens disponíveis na literatura para implementar esta atividade. Isso pode servir como um referencial para organizações e pessoas interessadas no tópico estudado.

## 5.2. Contribuições

A seguir são apresentadas algumas contribuições obtidas durante o desenvolvimento deste trabalho:

- **Revisão Sistemática da Literatura** – a revisão sistemática da literatura avaliou 1485 artigos, selecionando 411 destes por meio da aplicação de um protocolo de revisão sistemática. O protocolo criado, incluindo os seus critérios de inclusão e método de avaliação de qualidade, é flexível o suficiente para ser utilizado em outros contextos, que não somente à rastreabilidade de requisitos;
- **Trabalho de Conclusão de Curso** – o planejamento e a execução da Revisão Sistemática da Literatura resultaram em um trabalho de conclusão de curso, que apresentou a categorização de partes dos resultados da mesma;
- **Catálogo de Abordagens** – o catálogo de abordagens foi a principal contribuição do trabalho, já que foi constituído por informações obtidas por meio de um método rigoroso, confiável e auditável, que é o de Revisão Sistemática da Literatura. É esperado que o catálogo de abordagens auxilie na implantação da atividade de rastreabilidade de requisitos em projetos de software dentro de organizações, reduzindo custos e aumentando a qualidade das mesmas;
- **Produção de Trabalho Científico** – houve a publicação de dois trabalhos no contexto desta dissertação de mestrado: um no Workshop Anual do MPS (WAMPS), o qual apresentou indicações de implementação de rastreabilidade de requisitos, baseadas na revisão sistemática da literatura realizada, no contexto do MR-MPS.BR; outro na Revista FSMA, que apresentou os resultados obtidos com a Revisão Sistemática da Literatura.



### **5.3. Limitações**

Uma das principais limitações deste trabalho está no fato da quantidade de pesquisadores envolvidos na execução da revisão sistemática da literatura, já que apenas dois pesquisadores participaram do processo, sendo esta a quantidade mínima para a realização deste método.

Outra limitação deu-se pela quantidade de fontes de pesquisas presentes no trabalho, pois, mesmo com um número relevante de trabalhos encontrados, outras fontes poderiam ter contribuído ainda mais para a realização do mesmo, porém, estas não satisfizeram os critérios de seleção de fontes.

Quanto ao catálogo, uma limitação que pode ser apresentada é a pequena frequência na presença de exemplos de implementações de técnicas e a disponibilização de endereços para downloads de ferramentas nos estudos avaliados. Além disso, outra limitação que pode ser apontada é quanto à avaliação do catálogo, pois devido ao tempo de execução desta pesquisa e a disponibilidade de demais especialistas na área, o catálogo foi avaliado apenas por um especialista usando a técnica de Revisão por Pares.

### **5.4. Trabalhos Futuros**

Este trabalho pode ser expandido em várias direções, devido à sua alta abrangência e relevância. Nesse sentido, espera-se que esta dissertação oriente novos trabalhos a respeito do mesmo tema. Assim, esta seção identifica sugestões de prosseguimentos do trabalho aqui apresentado, indicando possíveis evoluções que podem torná-lo mais completo e adequado para a rastreabilidade de requisitos.

#### **5.4.1. Evolução da Revisão Sistemática**

A revisão sistemática da literatura identificou diversas abordagens disponíveis na literatura com o foco em rastreabilidade de requisitos, porém, outras técnicas podem não terem sido identificadas nos estudos, devido ao número de fontes pesquisadas, ou até mesmo ao período de busca, já que existem eventos especializados na área possibilitando então que novas técnicas sejam descobertas. Sendo assim, uma possível evolução no protocolo de revisão pode ajudar a encontrar técnicas que possivelmente não tenham sido identificadas no estudo.

### **5.4.2. Aplicação Prática do Catálogo**

A aplicação deste catálogo na indústria, com sua utilização em um projeto de desenvolvimento de software pode ser de grande valia para o amadurecimento deste, isto porque oportunidades de melhoria em relação a sua estrutura, forma de apresentação ou mesmo conteúdo podem ser encontradas, além do que, sua efetividade quanto ao que se propõe pode ser avaliada.

### **5.4.3. Expansão do Catálogo**

Com a possibilidade do encontro ou surgimento de outras técnicas de apoio à rastreabilidade de requisitos, o catálogo também pode ser expandido neste contexto futuramente, além do que, pode também ser acrescido outras informações sobre as técnicas, caso sejam identificadas como importantes.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABNT – Associação Brasileira de Normas e Técnicas. **NBR ISO/IEC 12207:2009 – Engenharia de Sistemas de Software – Processos de Ciclo de Vida de Software**. Rio de Janeiro, Brasil, 2009.

AHN, S.; CHONG, K. **A feature-oriented requirements tracing method: A study of cost-benefit analysis**. In Proc. of 2006 International Conference on Hybrid Information Technology, Washington, DC, USA, 2006, IEEE Computer Society, pp. 611?616. 2006.

ALVES, C. F. **Seleção de Produtos de Software Utilizando uma Abordagem Baseada em Engenharia de Requisitos**. MSc. Thesis, Departamento de Informática, Universidade Federal de Pernambuco, Brasil, Março 2001.

ANTONIOL, G., CANFORA, G., DE LUCIA, A. et al. **Recovering Code to Documentation Links in OO Systems**. In: Proceedings of the 6th Working Conference on Reverse Engineering, pp. 136-144, Atlanta, GA, EUA, Out. 1999.

ANTONIOL, G.; CANFORA, G.; DE LUCIA, A; CASAZZA, G. **Information retrieval models for recovering traceability links between code and documentation**. Software Maintenance, IEEE. 2011.

ASSAWAMEKIN, N.; SUNETNANTA, T.; PLUEMPITIWIRIYAWAJ, C. **MUPRET: An Ontology-Driven Traceability Tool for Multiperspective Requirements Artifacts**. 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on Computer and Information Science, 943–948. doi:10.1109/ICIS.2009.55. 2009.

ASUNCION, H. U.; ASUNCION, A. U.; TAYLOR, R. N. **Software traceability with topic modeling**. 2010 ACM/IEEE 32nd International Conference on Software Engineering, 1, 95 – 104. doi:10.1145/1806799.1806817. 2010.

BAVOTA, G. et al. **TraceME: Traceability Management in Eclipse**. (ICSM), 2012 28th IEEE International Conference on Software Maintenance, 642–645. doi:10.1109/ICSM.2012.6405343. 2012.

BERTOLLO, G., FALBO, R. A. **Apoio Automatizado à Definição de Processos de Software em Níveis**. In: II Simpósio Brasileiro de Qualidade de Software, SBQS 2003, Fortaleza, Brasil. Anais. 2003, p. 1-6.

BIOLCHINI, J., MIAN, P.G., NATALI, A.C., TRAVASSOS, G.H. **Systematic Review in Software Enginering: Relevance and Utility**. Relatório Técnico ES-679/05, PESC – COPPE/UFRJ, 2005.

BJØRNSON, F.O. e DINGSØYR, T. **Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts**. Findings and Research Methods Used. Information and Software Technology, Elsevier, vol. 50, pp. 1055 – 1068. 2008.

BLEI, D.; NG, A.; JORDAN, M. **Latent Dirichlet Allocation**. Journal of Machine Learning Research, 3:993–1022, 2003.

BOEHM, B. W. **Software engineering economics**. Software Engineering, IEEE Transactions on, n. 1, p. 4-21, 1981.

CLARKE, M; OXMAN, A. D.; **Cochrane Reviewers' Handbook 4.1**. In: Review Manager (RevMan). Version 4.1. Oxford, England: The Cochrane Collaboration, 2000.

CLELAND-HUANG J.; CHANG C.; WISE J. **Supporting Event Based Traceability through HighLevel Recognition of Change Events**. Proceedings of IEEE COMPSAC Conference, Oxford, England, August 2002.

CLELAND-HUANG J.; SCHMELZER D. **Dynamic Tracing Non-Functional Requirements through Design patter Invariants**. Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003), Canada, October, 2003.

CLELAND-HUANG, J.; CHRISTENSEN, C. **Event-based traceability for managing evolutionary change**. IEEE Transactions on Software Engineering, v. 29, n. 9, p. 796–810, 2003.

CLELAND-HUANG, J. et al. **Goal-centric traceability for managing non-functional requirements**. Proceedings of the 27th International Conference on Software Engineering - ICSE '05, 362. doi:10.1145/1062455.1062525. 2005.

CLELAND-HUANG, J. **Toward Improved Traceability of Non-Functional Requirements**. Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering TEFSE'05, ACM, 2005.

CLELAND-HUANG, J. et al. **Center of Excellence for Traceability: Problem Statements and Grand Challenges (v0.1)**. Center of Excellence for Traceability Technical Report COET-GCT-06-01-0.9. 2006.

CLELAND-HUANG, J. **Requirements traceability - when and how does it deliver more than it costs?** Proceedings of the 14th IEEE International Requirements Engineering Conference, p. 323, Washington, USA. IEEE Computer Society. 2006.

CLELAND-HUANG, J.; MARRERO, W.; BERENBACH, B. **Goal-Centric Traceability: Using Virtual Plumblines to Maintain Critical Systemic Qualities**. Software Engineering, IEEE Transactions on, 34(5), 685–699. 2008.

CHEN, X.: GRUNDY, J. **Improving automated documentation to code traceability by combining retrieval techniques**. 26th IEEE/ACM International Conference on Automated Software Engineering (ASE), 223–232. doi:10.1109/ASE.2011.6100057. 2011.

CYSNEIROS, L. M. **Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual**. Tese de Doutorado. PUC-Rio, Fevereiro de 2001.

COCHRANE, A. Effective care in pregnancy and childbirth. In Chalmers I, Enkin M, Keirse MJNC, eds. Oxford University Press, Oxford, 1989.

COSTA, C. S. **Uma abordagem baseada em evidências para o gerenciamento de projetos no desenvolvimento distribuído de software**. Dissertação de Mestrado – Programa de Pós-Graduação em Ciência da Computação - Universidade Federal de Pernambuco, Recife, 2010.

DE LUCIA, A.; FASANO, F.; OLIVETO, R.; TORTORA, G. **Enhancing an artefact management system with traceability recovery features**. In 20th IEEE Int. Conf. on Software Maintenance, pages 306 – 315. IEEE, 2004.

DE LUCIA, A.; FASANO, F.; OLIVETO, R.; TORTORA, G. **Adams re-trace: A traceability recovery tool**. In Proc. of the 9th European Conf. on Software Maintenance and Reengineering, pages 32–41. IEEE Computer Society, March 2005.

DE LUCIA, A.; FASANO, F.; OLIVETO, R. **Traceability Management for Impact Analysis**. *Frontiers of Software Maintenance*. Beijing, China, IEEE, 2008.

DEKHTYAR, A.; HAYES, J. H.; SUNDARAM, S.; HOLBROOK, E. A.; DEKHTYAR, O. **Technique Integration for Requirements Assessment**. 15th IEEE International Requirements Engineering Conference (RE 2007), 141–150. doi:10.1109/RE.2007.17. 2007.

DEERWESTER, S. et al. **Indexing by latent semantic analysis**. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

DERMEVAL, D. et al. **On the Use of Metamodeling for Relating Requirements and Architectural Design Decisions**. 1278–1283. 2013.

DEVA KUMAR, D.; SANYAL, R. **Static UML Model Generator from Analysis of Requirements (SUGAR)**. *Advanced Software Engineering and Its Applications*, 2008. ASEA 2008 , vol., no., pp.77,84, 13-15 Dec. 2008

DORFMAN, M.; FLYNN, R. F. **ARTS - an automated requirements traceability system**. *Journal of Systems and Software*, 4(1):63–74, 1984.

DUAN, C.; CLELAND-HUANG, J. **Clustering support for automated tracing**. *Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering - ASE '07*, 244. doi:10.1145/1321631.1321668. 2007.

DYBA, T.; DINGSOYR, T.; HANSSSEN, G.K. **Applying Systematic Reviews to Diverse Study Types: An Experience Report**. in *Proceedings of the 1st Empirical Software Engineering and Measurement (ESEM-07)*. Madri, Spain: IEEE Computer Society, pp. 225-234, 2007.

EASTERBROOKS, S. et. al. **Selecting Empirical Methods for Software Engineering Research.** International Conference on Automated Software Engineering. Atlanta, Georgia, EUA, 2007.

EBNER, G.; KAINDL, H. **Tracing All Around in Reengineering.** IEEE Software, 19(3):70–77, 2002.

EGYED, A.; GRÜNBACHER, P. **Automating Requirements Traceability: Beyond the Record & Replay Paradigm.** In: Proceedings of the 17th International Conference on Automated Software Engineering, pp. 163-171, Edinburgh, Escócia, Set. 2002.

EGYED, A.; GRÜNBACHER, P., REY, M.; LINZ, A. **STRADA: A Tool for Scenario-based Feature-to-Code Trace Detection and Analysis.** In Proceedings of 29th International Conference on Software Engineering, 2007.

EGYED, A., BIFFL, S., HEINDL, M. et al. **Determining the Cost-Quality TradeOff for Automated Software Traceability.** In: Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering, pp. 360-363, Long Beach, CA, EUA, Nov. 2005.

EYAL-SALMAN, H.; SERIAI, A.-D.; DONY, C.; AL-MSIE'DEEN, R. **Recovering traceability links between feature models and source code of product variants.** Proceedings of the VARIability for You Workshop on Variability Modeling Made Useful for Everyone - VARY '12, 21–25. doi:10.1145/2425415.2425420. 2012.

FALBO, R. A. **Integração de Conhecimento em um Ambiente de Desenvolvimento de Software.** Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, Dezembro de 1998.

FERREIRA, P. J. A. V. **Rastreabilidade de Ponto de Função.** Dissertação de Mestrado - Programa de Pós-Graduação em Informática, UNIRIO, Rio de Janeiro, 2010.

FERREIRA, D. A. L. **Um Estudo de Mapeamento Sistemático da Literatura para Rastreabilidade de Requisitos no Contexto de Projetos de Software.** Belém-PA. Trabalho de Conclusão de Curso – Instituto de Ciências Exatas e Naturais, Universidade Federal do Pará, 2012.

FILHO, G.; LENCASTRE, M. **Towards a Traceability Visualisation Tool.** Proceedings - 2012 8th International Conference on the Quality of Information and Communications Technology, 221–223. doi:10.1109/QUATIC.2012.60. 2012.

FUGGETA, A. **Software Process: A Roadmap.** In: International Conference On Software Engineering, 22., 2000, New York. Proceedings. New York: ACM Press. 2000.

GALVAO, I.; GOKNIL, A. **Survey of Traceability Approaches in Model-Driven Engineering.** EDOC 2007. 11th IEEE International Enterprise Distributed Object Computing Conference, 313–313. doi:10.1109/EDOC.2007.42. 2007.

GENVIGIR, E.C. **Um modelo para rastreabilidade de requisitos de software baseado em eneralização de elos e atributos.** Dissertação (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009.



GIBIEC, M.; CZAUDERNA, A.; CLELAND-HUANG, J. **Towards mining replacement queries for hard-to-retrieve traces.** Proceedings of the IEEE/ACM International Conference on Automated Software Engineering - ASE '10, 245. doi:10.1145/1858996.1859046. 2010.

GOTEL, O. C. Z. e FINKELSTEIN, A. C. W. **An Analysis of the Requirements Traceability Problem.** In: Proceedings of the International Conference on Requirements Engineering, pp. 94-101, Colorado Springs, CO, EUA, Abr 1994.

GOTEL, O.; FINKELSTEIN, A. **Contribution Structures.** Proceedings of 2nd International Symposium on Requirements Engineering. IEEE Computer Society Press, pp. 100-107. 1995.

GRECHANIK, M.; MCKINLEY, K. S.; PERRY, D. E. **Recovering And Using Use-Case-Diagram-To-Source-Code Traceability Links.** In: Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGMOD Symposium on the Foundations of Software Engineering, pp. 95-104, Dubrovnik, Croácia, Set.2007.

GRUHN, V. **Process-Centered Software Engineering Environments: A Brief History and Future Challenges.** In: Annals of Software Engineering 14, p. 363-382. 2012.

HAYES, J. H.; DEKHTYAR, A.; OSBORNE, J. **Improving Requirements Tracing via Information Retrieval.** In proceedings of the 11th IEEE International Requirements Engineering Conference, Monterey Bay, 2003.

HAYES, J. et al. **REquirements TRacing on target (RETRO): improving software maintenance through traceability recovery.** Innovations in Systems and Software Engineering, 3:193–202, 2007.

HEIM, P.; LOHMANN, S.; LAUENROTH, K.; ZIEGLER, J. **Graph based visualization of requirements relationships.** In Proceedings of Requirements Engineering Visualization, REV '08, pages 51-55, Washington, DC, USA, 2008.

HEINDL, M.; BIFFL, S. **A case study on value-based requirements tracing.** Proceedings of the 10th European Software Engineering Conference Held Jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering - ESEC/FSE-13, 60. doi:10.1145/1081706.1081717. 2005.

HEINRICH, G. **LDA-J Library.** Code available at <http://www.arbylon.net/projects/>.

HOWARD, S. G. **Requirements and traceability management.** In Colloquium on Tools and Techniques for Maintaining Traceability During Design, Savoy Place, London WCR OBL U.K., 2 December 1991.

HUMPHREY, W. **Managing the software process.** Massachussets: Addison-Wesley, 1989. 512 p.

ISO/IEC 12207: **Systems and Software Engineering – Software Life Cycle Processes**, 2nd edition, 2008.



JACKSON, J. **A key frased based traceability scheme.** In Colloquium on Tools and Techniques for Maintaining Traceability During Design. SavoyPlace, London WC. R OBL, U.K., 2 december 1991.

JIRAPANTHONG, W.; ZISMAN, A. **XTraQue: traceability for product line systems.** *Software & Systems Modeling*, 8(1), 117–144. doi:10.1007/s10270-007-0066-8. 2007.

JURETA, I. J.; FAULKNER, S. **Tracing the Rationale Behind UML Model Change Through Argumentation.** In Proceedings of the 26th International Conference on Conceptual Modeling (pp. 454–469). Berlin, Heidelberg, 2007.

KANNENBERG, A. e SAIEDIAN, H. **Why Software Requirements Traceability Remains a Challenge.** *Crosstalk – The Journal of Defense Software Engineering*, v. 22, n. 5 (Jul/Ago), pp. 14-19. 2009.

KELLEHER, J.; SIMONSSON, M. **Utilizing use case classes for requirement and traceability modeling.** In MS'06: Proceedings of the 17th IASTED international conference on Modelling and simulation, pages 617–625, Anaheim, CA, USA. ACTA Press. 2006.

KITCHENHAM, B. **Procedures for Performing Systematic Reviews.** Joint Technical Report, Software Engineering Group, Keele University, and Empirical Software Eng., Nat'l ICT Australia, 2004.

KITCHENHAM, B. **Guidelines for performing Systematic Literature Reviews.** In Software Engineering, Technical Report EBSE-2007-01, Departament of Computer Science Keele University, Keele, 2007.

KOTONYA, G., SOMMERVILLE, I. **Requirements Engineering: Processes and Techniques.** Wiley, John & Sons Inc. 1998.

LEAL, M., FIGUEIREDO, M. C.; DE SOUZA, C. R. B. **Uma abordagem semi-automática para a manutenção de links de rastreabilidade.** In: 11th Workshop on Requirements Engineering, 2008, Barcelona, ES. Proceedings of the 11th Workshop on Requirements Engineering, p. 47-58. 2008.

LEAL, M. D. **Uma abordagem semiautomática de geração de informações de rastreabilidade de software.** Dissertação de Mestrado. Universidade Federal do Pará. Belém, 2011.

LEE, R.; TEPFENHART, W. **UML e C++ - Guia Prático de Desenvolvimento Orientado a Objeto.** Editora Makron Books, 2001.

LIN, J.; LIN, C.; CLELAND-HUANG, J.; SETTIMI, R.; AMAYA, J.; BEDFORD, G.; ZOU, X. **Poirot: A Distributed Tool Supporting Enterprise-Wide Automated Traceability.** 14th IEEE International Conference Requirements Engineering, 363 – 364. 2006.

LINDVAL, M.; SANDAHL, K. **Practical Implications of Traceability.** In Software Practice and Experience, vol. 26, no. 10, p. 1161-1180. 1996.

- LORMANS, M.; DEURSEN, A. **Can LSI help reconstructing requirements traceability in design and test?** In Proceedings of the 10th European Conference on Software Maintenance and Reengineering, 10 pp.-56. doi:10.1109/CSMR.2006.13. 2006.
- LOUCOPOULOS, P; KAVAKLI, V. **Enterprise Modelling and Teleological Approach to Requirements Engineering.** In International Journal of Intelligent and Cooperative Information Systems, 1995.
- MÄDER, P.; GOTEL, O. C. Z.; PHILIPPOW, I. **Rule-Based Maintenance of Post-Requirements Traceability Relations.** International Requirements Engineering Conference, 2008. RE '08. 16th IEEE, (c), 23–32. doi:10.1109/RE.2008.24. 2008.
- MAFRA, S.; BARCELOS, R. TRAVASSOS, G. **Aplicando uma metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software.** In Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006), v. 1, pp. 239-254. 2006.
- MAFRA, S.; TRAVASSOS, G. **Estudos Primários e Secundários apoiando a busca por Evidencia em Engenharia de Software - Relatório Técnico: RT-ES-687/06 – Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ – Rio de Janeiro, 2006.**
- MAHMOUD, A.; NIU, M. **TraCter: a tool for candidate traceability link clustering.** In RE, 2011, pp. 335–336.
- MALETIC, J. et al. **Using a hypertext model for traceability link conformance analysis.** In 2nd Workshop on Traceability in Emerging Forms of Software Engineering, pages 47–54, Montreal, Canada, 2003. TEFSE 2003.
- MARCUS A., MALETIC J.I. **Recovering Documentation-to-Source-Code Traceability Links using Latent Semantic Indexing.** In International Conference Software Engineering, 2003.
- MENDELEY. **Mendeley Reference Manager.** 2015. Disponível em: <<http://www.mendeley.com/>>. Acessado em 24/04/2015.
- MERTEN, T; JUEPPNER, D; DELATER, A. Improved Representation of Traceability Links in Requirements Engineering Knowledge using Sunburst and Netmap Visualizations. In Proceedings of the 4th International Workshop on Managing Requirements Knowledge, Italy, August 30, 2011.
- MONTEIRO, C. V. F. **Impacto do uso de ferramentas de software n as fases iniciais do processo de inovação.** Dissertação de Mestrado. Universidade Federal de Pernambuco. Recife, PE, Brasil, 2010.
- MULROW, C.D. **Systematic reviews: Rationale for sistematic reviews.** British Medical Journal v.309, pp.597-599, 1994.

MUNSON, E.; NGUYEN, T. Concordance, conformance, versions, and Traceability. Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering, Long Beach, California, 2005.

NAIR, S.; DE LA VARA, J. L.; SEN, S. **A review of traceability research at the requirements engineering conference re@21**. In: Requirements Engineering Conference (RE), 2013 21st IEEE International. IEEE. p. 222-229. 2013.

NASLAVSKY, L.; ALSPAUGH, T. A.; RICHARDSON, D. J.; ZIV, H. **Using scenarios to support traceability**. Proceedings of the 3rd International Workshop on Traceability in Emerging Forms of Software Engineering - TEFSE '05, 25. doi:10.1145/1107656.1107663. 2005.

NHSCRD - NHS Centre for Reviews and Dissemination. **Database of Abstracts of Reviews of Effectiveness**. In: The Cochrane Library, Issue 1. Oxford: Updated quarterly. 2003.

NOGUEIRA, M.O. **Qualidade no Setor de Software Brasileiro**. Tese de Doutorado, COPPE/UFRJ, abril 2006.

NUSEIBEH, B.; EASTERBROOK, S. **Requirements Engineering: A Roadmap**. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland. Jun, 2000.

PANDANABOYANA, S.; SRIDHARAN, S.; YANNELLI, J.; HAYES, J. H. **REquirements TRacing On target (RETRO) Enhanced with an Automated Thesaurus Builder**. International Workshop on Traceability in Emerging Forms of Software Engineering, 61–67. 2013.

PARK, S.; KIM, H.; KO, Y.; SEO, J. **Implementation of an efficient requirements-analysis supporting system using similarity measure techniques**. Inf Softw Technol 42(6):429–438. 2000.

PINHEIRO, F.; GOGUEN, J. **An object-oriented tool for tracing requirements**. IEEE Software, v. 13, n. 2, p. 796–810, 1996.

PMI – Project Management Institute. **PMBOK Guide – Um guia do Conhecimento em gerenciamento de projetos**. Quinta Edição, 2014.

POHL, K. **Process-Centered Requirements Engineering**. Wiley, New York 1996. ISBN 978-0-863-80193-8. PROTART, 1996.

PRESSMAN, R. S. **Engenharia de Software - uma abordagem profissional**. 7a. ed. Porto Alegre: AMGH Bookman, v. I, 2011.

RAJA, U. A.; KAMRAN, K. **Framework for Requirements Traceability**. April, 2008.

RAMAMOORTHY, C. V.; USUDA, Y.; PRAKASH, A.; TSAI, W. T. **The evolution support environment system**. IEEE Transactions on Software Engineering. November 1900.

- RAMESH, B. e JARKE, M. **Toward reference models for requirements traceability.** IEEE Transactions on Software Engineering, Vol. 27, Issue 1, Janeiro, p. 58 – 93. 2001.
- SANTOS, G. **Revisão Sistemática**, Minicurso. Simpósio Brasileiro de Qualidade de Software – SBQS 2010, Belém – PA, 2010.
- SARDINHA, A.; NIU, N.; YU, Y.; RASHID, A. **EA-Tracer: Identifying Traceability Links between Code Aspects and Early Aspects.** Proc. of the 27th ACM Symposium on Applied Computing (ACM SAC'12), Riva Del Garda, Italy, 1035–1042. 2012.
- SCHWARZ, H.; EBERT, J.; WINTER, A. **Graph-based traceability: a comprehensive approach.** In Software & Systems Modeling. 9(4), 473–492. doi:10.1007/s10270-009-0141-4. 2009.
- SEI – Software Engineering Institute. **Capability Maturity Model Integration for Development – CMMI-Dev.** Versão 1.3, 2010.
- SEI. **Capability Maturity Model Integration (CMMI) for Development**, Version 1.3. Carnegie Mellon, USA, 2010.
- SHAHID, M.; IBRAHIM, S.; MAHRIN, M. N. **An Evaluation of Requirements Management and Traceability Tools.** World Academy of Science, Engineering and Technology., 596–601. 2011.
- SILVA, E. L.; MENEZES, E. M. **Metodologia da Pesquisa e Elaboração da Dissertação.** 3. ed. rev. Atual – Laboratório de Ensino a Distância da UFSC, 2001.
- SILVA, W. C.; MARTINS, L. E. G. **Paradigma: uma ferramenta de apoio à elicitação e modelagem de requisitos baseada em processamento de linguagem natural.** In: Workshop on Requirements Engineering, 11th, 2008, Barcelona. Proceedings... Barcelona: [S.n.], 2008.
- SOFTEX, **Melhoria do Processo de Software Brasileiro (MPS.BR) - Guia Geral 2012.** Disponível em: <[http://www.softex.br/wp-content/uploads/2013/07/MPS.BR\\_Guia\\_Geral\\_Software\\_20121.pdf](http://www.softex.br/wp-content/uploads/2013/07/MPS.BR_Guia_Geral_Software_20121.pdf)>. Acesso em: 24 janeiro 2014.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. **MPS.BR - Guia Geral de Software: 2012.** 2012.
- SOFTEX – Associação para Promoção da Excelência do Software Brasileiro. **Guia de Implementação - Parte 1: 2013.** 2013.
- SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering: A Good Practice Guide.** Wiley, John & Sons Inc. 1997.
- SOMMERVILLE, I. **Engenharia de Software.** 9a edição, Pearson Addison-Wesley, 2011.

SPANOUDAKIS, G.; ZISMAN, A.; PÉREZ-MIÑANA, E.; KRAUSE, P.; Rule-based generation of requirements traceability relations. *Journal of Systems and Software*, 72(2), 105–127. doi:10.1016/S0164-1212(03)00242-5. 2004

SPENCE, I.; PROBASCO, L. **Traceability Strategies for Managing Requirements with Use Cases**. Rational Software White Paper. 2000.

TANG, Y. J.; HAN, J. **A Rationale-based Architecture Model for Design Traceability and Reasoning**. *Journal of Systems and Software*, 80(6):918-934, June 2007.

THOMMAZO, A.; MARTINS, M. D.; FABBRI, S. **O Gerenciamento de Requisitos no Ambiente COCAR**. Workshop de Engenharia de Requisitos. 2007.

TORANZO, M.; CASTRO, J. **A Comprehensive Traceability Model to Support the Design of Interactive Systems**. International Workshop on Interactive System Development and Object Models - WISDOM 99. Lisboa, Portugal. Jun. 1999.

TORKAR, R.; GORSCHKEK, T.; FELDT, R.; SVAHNBERG, M.; RAJA, U. A.; KAMRAN, K. **Requirements Traceability: a Systematic Review and Industry Case Study**. *International Journal of Software Engineering and Knowledge Engineering*, 22(03), 385–433. doi:10.1142/S021819401250009X. 2012.

TRAVASSOS, G. H. **O Modelo de Integração de Ferramentas da Estação TABA**. Tese de Doutorado. COPPE/UFRJ, Rio de Janeiro, Brasil, 1994.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. D. **Introdução à Engenharia de Software Experimental**. Relatório Técnico RT-ES-590/02 do Programa de Engenharia de Sistemas e Computação. COPPE/UFRJ, Rio de Janeiro, 2002.

TRAVASSOS, G., BIOLCHINI J. **Revisões Sistemáticas Aplicadas a Engenharia de Software**. In: XXI SBES - Brazilian Symposium on Software Engineering, João Pessoa, PB, Brasil, 2007.

VAN LAMSWEERDE, A. **Requirements Engineering in the year 00: A Research Perspective**. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE), Limerick, Ireland. Jun, 2000.

VIANNA FERREIRA, P. J. A.; BARROS, M. D. O. **Traceability between function point and source code**. Proceeding of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering - TEFSE '11, 10. doi:10.1145/1987856.1987860. 2011.

VOYTEK, J.; NÚÑEZ, J. **Visualizing Non-Functional Traces in Student Projects in Information System and Service Design**. In proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011.

WATKINS, R.; NEAL, M. **Why and How of Requirements Tracing**. *IEEE Software*, v. 11, n. 4 pp. 104-106. Jul, 1994.

WIERINGA, R. **An introduction to requirements traceability**. Tech. rep., Institute for Mathematics and Computer Science, Vrije Universiteit, Amsterdam, The Netherlands. 1995.

WINKLER, S.; PILGRIM, J. **A survey of traceability in requirements engineering and model-driven development**. *Software & Systems Modeling*, 9(4), 529–565. doi:10.1007/s10270-009-0145-0. 2009.

WITTEN, H. et al. **Kea: practical automatic keyphrase extraction**. 4th ACM DL, Berkeley, pp. 254-255. 1999.

XIE, M.; POSHYVANYK, D. **When and How to Visualize Traceability Links**. In the Proceedings of the 3rd ACM International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2005) , Long Beach, CA, November 8th, 2005 pp. 56-61.

YU, W. D. **Verifying software requirements: a requirements tracing methodology and tool RADIX**. *IEEE Journal on Selected Areas in Communication*, 1994.

ZAHRAN, S. **Software Process Improvement**. Addison-Wesley. 1998.

ZAVE, P. **Classification of Research Efforts in Requirements Engineering**. *ACM Computer Surveys*, Vol. 29, No. 4. 1997.

ZIFTCI, C.; KRUEGER, I. **Tracing requirements to tests with high precision and recall**. 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011), 472–475. doi:10.1109/ASE.2011.6100102.

ZISMAN, A.; SPANOUDAKIS, G. **Software Traceability: Past, Present, and Future**. The Newsletter of the Requirements Engineering Specialist Group of the British Computer Society – URL : <http://www.resg.org.uk/archive/rq33.pdf>. Accessed: 05/03/2015.



**APÊNDICE A – PROTOCOLO DE REVISÃO****Histórico de Revisões**

<b>Data</b>	<b>Ver são</b>	<b>Descrição</b>	<b>Autor</b>
<b>07/10/2013</b>	<b>0.1</b>	<b>Início da Concepção do Protocolo de Revisão Sistemática (objetivos, referências e questões de pesquisa).</b>	<b>Paulo Malcher</b>
<b>26/10/2013</b>	<b>0.2</b>	<b>Definição da fontes de buscas, critérios de inclusão e exclusão, critérios de qualidade e definição do processo de avaliação da qualidade</b>	<b>Paulo Malcher</b>
<b>07/11/2013</b>	<b>0.3</b>	<b>Definição da Estratégia de Extração de resultados e Refinamento das questões de pesquisa secundárias</b>	<b>Paulo Malcher</b>
<b>07/04/2014</b>	<b>0.4</b>	<b>Atualização do Protocolo de Revisão</b>	<b>Paulo Malcher</b>
<b>03/02/2015</b>	<b>1.0</b>	<b>Revisão do Protocolo de Revisão</b>	<b>Paulo Malcher</b>



## 1. Contexto

O projeto SPIDER (acrônimo para Software Process Improvement – Development and Research) surgiu em 2009 com o objetivo de propor abordagens sistematizadas para apoiar a implementação de programas de melhoria de processo de software. Com relação à abordagem para sistematizar a implementação de processos, o Projeto SPIDER tem como possibilidades: Utilização de ferramentas já existentes no mercado; A adequação de ferramentas já existente no mercado; A criação de novas ferramentas. O Projeto SPIDER foca, ainda, na utilização de software livre, o que proporciona a liberdade: de executar o programa; de estudar como o programa funciona, e adaptá-lo para as suas necessidades; de redistribuir cópias; de modificar o programa; e de liberar estas modificações (GNU, 2011). Utilizar software livre para a implementação de modelos de qualidade pode ocasionar em diminuição de custos e tempo de implementação (OLIVEIRA, 2010).

Neste contexto, esta revisão sistemática é parte de um projeto de dissertação de mestrado vinculado ao Projeto SPIDER com o objetivo de reunir conhecimentos a respeito da implementação da Rastreabilidade de Requisitos, criando assim um catálogo das principais técnicas utilizadas para sua realização, bem como a definição de em qual contexto estas técnicas são melhores utilizadas.

A rastreabilidade de requisitos é usada como meio de manter a relação entre requisitos e outros artefatos criados durante o processo de desenvolvimento de software. Ela também visa possibilitar a compreensão desses relacionamentos, ou seja, qual tipo de relação existe entre os artefatos envolvidos.

A rastreabilidade de requisitos tem sido identificada na literatura como fator de qualidade, característica que um sistema pode possuir e incluir como requisito não funcional (RAMESH, 2001). Em vista disso, esse tópico de pesquisa tem sido apresentado com certa frequência na literatura especializada, principalmente em relação a técnicas, métodos, modelos, ferramentas e outras abordagens para sua implementação. .

Neste contexto, esta revisão sistemática será um instrumento metodológico para apoiar a criação de um catálogo contento abordagens utilizadas na literatura para a implementação da Rastreabilidade de Requisitos, atendendo a um dos objetivos específicos do trabalho.

## 2. Objetivos

Esta revisão sistemática terá objetivo de identificar abordagens para a Rastreabilidade de Requisitos, no contexto de projetos de software, no que tange a modelos, metodologias, técnicas, ferramentas e afins. Desta forma, têm-se a seguinte estrutura, conforme proposto em Santos (2010):

**Analisar:** relatos de experiência e publicações científicas através de um estudo baseado em revisão sistemática.

**Com o propósito de:** identificar abordagens para apoiar a Rastreabilidade de Requisitos.

**Com relação:** a definição e uso de modelos, metodologias, ferramentas e demais instrumentos empregados para a implantação e execução da Rastreabilidade de Requisitos em organizações de desenvolvimento de software.

**Do ponto de vista:** de pesquisadores e organizações desenvolvedoras/mantenedoras de software.

**No contexto:** acadêmico e industrial.

## 3. Referências

KITCHENHAM, B. (2005) “Guidelines for performing Systematic Literature Reviews in Software Engineering”. Vol 2.3 EBSE Technical Report, EBSE-2007-01, 2007.

MAFRA, S., TRAVASSOS, G. (2005) “Técnicas de Leitura de Software: Uma Revisão Sistemática”. XIX Simpósio Brasileiro de Engenharia de Software (SBES 2005).

SANTOS, G. S. (2010) “Revisão Sistemática, Minicurso”. Simpósio Brasileiro de Qualidade de Software – SBQS 2010, Belém – PA.

## 4. Formulação da Pergunta

Questões de pesquisa são à base de uma revisão sistemática da literatura. Esta pesquisa objetiva investigar as propostas da literatura no que tange a seguinte indagação:

**Quais as abordagens existentes para apoiar a Rastreabilidade de Requisitos no contexto de Projetos de Software?**

Por “abordagens” o que se pretende investigar são os padrões (para fins desta revisão, por padrões entende-se modelos de processos, frameworks de processo, metodologias, técnicas, ferramentas e afins). Para tal, foi definida a seguinte questão de pesquisa a qual guiará esta revisão sistemática da literatura:

**Q1. Quais as abordagens existentes para apoiar a Rastreabilidade de Requisitos?**

**4.1. Estrutura da Questão Principal**

A questão levantada foi organizada conforme a estrutura Population, Intervention, Context, Outcomes, Comparison (PICOC), recomendada por Kitchenham (2007). Entretanto, apenas os itens População, Intervenção e Resultados foram considerados relevantes para a pesquisa. Tal restrição, segundo Santos (2010), caracteriza esta pesquisa como uma Revisão QUASI Sistemática da Literatura.

**4.1.1. Para Q1**

Tem-se o objetivo de identificar quais padrões para apoiar a Rastreabilidade de Requisitos (Intervenção), aplicáveis no contexto de organizações (empresas, instituições, centros e grupos) que atuam em projetos de software (População), sob forma de modelos de processos, técnicas, metodologias, ferramentas para Rastreabilidade de Requisitos. Logo, definiu-se a seguinte estrutura:

- **População (P):** Organizações de Software e Projetos de Software;
- **Intervenção (I):** Abordagens para apoiar atividades de Rastreabilidade de Requisitos;
- **Resultados (O):** Modelos de processos, técnicas, metodologias, ferramentas e frameworks de Rastreabilidade de Requisitos.

**4.2. Questões Secundárias**

Um conjunto de questões secundárias referentes à questão principal foi estabelecido, questões essas para serem respondidas durante a fase de extração de informações (para maiores detalhes, consultar **seção 15**). Tais questões têm objetivo de

esclarecer detalhes importantes que esta revisão procura identificar, para colaborar com o projeto onde está inserida.

**Q1A.** Qual o contexto de aplicação da abordagem encontrada?

**Q1B.** Quais as formas de rastreabilidade (para frente e para trás) são cobertas pela abordagem?

**Q1C.** Quais os tipos de rastreabilidade (vertical e horizontal) são cobertos pela abordagem?

**Q1D.** Quais os ativos (papéis, artefatos) envolvidos?

**Q1E.** Existem softwares de apoio para a abordagem proposta?

**Q1F.** Caso existam softwares de apoio, qual a sua licença de uso?

## **5. Escopo da Pesquisa**

Visando estabelecer limites viáveis para a execução desta pesquisa, foram definidos critérios para seleção das fontes de pesquisa e restrições para garantir a viabilidade da mesma.

### **5.1. Critérios de Seleção de Fontes**

Para seleção das fontes de pesquisa, foram definidos os seguintes critérios:

- Disponibilidade para consultas web.
- Disponibilidade para busca de artigos através do domínio da UFPA.
- Disponibilidade de artigos na íntegra através do domínio da UFPA ou a partir da utilização da engine de busca Google e/ou Google Scholar.
- Disponibilidade de artigos em inglês ou português.
- Relevância da fonte.

A princípio, serão consideradas fontes de dados que disponibilizem artigos na íntegra, de modo a diminuir o retrabalho por parte dos pesquisadores com relação a novas busca necessárias para encontrar as versões completas de artigos apresentando somente resumo/abstract. Adicionalmente, caso os artigos não estejam disponíveis na íntegra, serão utilizadas as máquinas de busca disponíveis nos sites do Google

(<http://www.google.com.br/>) e Google Scholar (<http://scholar.google.com.br/>) com o objetivo de encontrar a versão completa do artigo. Caso o artigo, ainda assim, não seja encontrado através da busca manual, a versão incompleta do mesmo será descartada da pesquisa.

## **5.2. Restrições**

Devido ao caráter da pesquisa, esta não deverá incorrer em ônus financeiro aos pesquisadores. Portanto apenas serão selecionadas as fontes que possibilitem consultas de forma gratuita ou a partir do acesso pelo domínio da UFPA. Serão apenas considerados os estudos obtidos através das fontes selecionadas e em conformidade com os critérios de inclusão e exclusão. A pesquisa estará restrita aos resultados publicados entre 01 de janeiro de 2003 até a data de realização das buscas, contemplando, desta forma, um período de 11 anos, dada a necessidade de identificar abordagens para apoio às atividades de Rastreabilidade de Requisitos mais atuais.

Serão considerados apenas artigos em inglês ou português devido à proficiência dos autores nestes idiomas.

## **5.3. Métodos de busca nas Fontes**

Os métodos de busca nas fontes podem ser definidos como: método de busca manual e método de busca automática.

### **5.3.1. Método de Busca Manual**

A busca manual utiliza as fontes onde os estudos estão inseridos em anais de eventos e não disponíveis em um repositório digital e sim em um documento digital. Na busca manual os anais das conferências são consultados para um determinado período específico. Todos os resultados apresentados são conferidos um por um permitindo o pesquisador a determinar se o artigo pertence ao conjunto que ele propôs.

### **5.3.2. Método de Busca Automática**

A pesquisa será realizada através de mecanismos de busca web por palavras-chave. Poderão ser necessárias eventuais buscas manuais por artigos que não sejam disponibilizados na íntegra, ou seja, artigos que possuam versão contendo apenas resumo/abstract. Para estes casos, o título do artigo, incluindo o nome dos autores, será

utilizado com uma “string de busca” a ser inserido nos mecanismos de busca web Google (<http://www.google.com.br/>) e Google Scholar (<http://scholar.google.com.br/>).

## **Palavras-Chave e Strings de Busca**

### **5.4. Palavras-Chave e Sinônimos**

A partir das questões de pesquisa, palavras-chave foram identificadas em acordo com a estrutura População, Intervenção e Resultados. Por considerar estudos nos idiomas Português e Inglês, as palavras-chave foram definidas para ambos os idiomas, para a posterior formulação das strings de busca. Para cada questão, segue a listagem de palavras-chave definidas:

- **POPULAÇÃO**
  - Software Development, Software Project, Project, Development, Organization, Enterprise, Company, Industry, Institute, Research Group, Technology Center;
- **INTERVENÇÃO**
  - Requirements Traceability, Traceability, Requirements Tracing;
- **RESULTADOS**
  - Model, Process, Framework, Method, Technique, Methodology, Knowledge. Activity, Task, Tool, Software, Program, System, application, environment, workbench;

### **5.5. Strings de Busca**

Para elaboração da string de busca, as palavras-chave e sinônimos relacionados são agrupadas por meio de operador <OR> e os conjuntos de termos são agrupados com o operador <AND>, de acordo com a estrutura PICO (ou PIO no caso desta pesquisa) conforme segue (Santos, 2010):

#### **P <and> I <and> C <and> O**

Conforme já especificado anteriormente, a Comparação (ou Controle) não se aplica ao contexto desta revisão. Tem-se, então, que o conjunto de palavras-chave referentes a Comparação (C) é vazio ( $C=\emptyset$ ). Adicionalmente, as palavras chave da Intervenção são relevantes apenas no contexto da Rastreabilidade de Requisitos, portanto as palavras chave.

Para a questão de pesquisa serão consideradas 2 strings de busca (uma para busca em inglês e outra para o português).

#### **5.5.1. Para Q1**

- ("Software" OR "Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center") AND ("Requirements Traceability" OR "Traceability" OR "Requirements Tracing") AND ("Model" OR "Process" OR "Framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench").

Algumas adaptações são necessárias de acordo com as necessidades específicas de cada fonte de pesquisa. A seguir são apresentadas as Strings de busca de cada fonte.

#### **Para IEEE**

- (("Software\*" AND ("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center")) AND ("Requirements Traceability" OR "Traceability" OR "Requirements Tracing") AND ("Model" OR "Process" OR "framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories")))

#### **Para El Compendex**

- Software AND (Project OR Development OR Organization OR Enterprise OR Company OR Industry OR Institute OR Research Group OR Technology Center) AND (Requirements Traceability) AND (Model OR Process OR framework OR Method OR Technique OR Methodology OR Task OR Tool OR System OR Application OR Environment OR Workbench OR Problem OR Issue OR Challenge OR Success Stories)



### Para ISI Web of Knowledge

- (((("Software" AND ("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center")) AND ("Requirements Traceability" OR "Traceability" OR "Requirements Tracing") AND ("Model" OR "Process" OR "framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories"))))

### Para ACM

- all of this text (and): "Software" "Requirements Traceability" any of this text (or): "Project" "Development" "Organization" "Enterprise" "Company" "Industry" "Institute" "Research Group" "Technology Center" "Traceability" "Requirements Tracing" "Model" "Process" "framework" "Method" "Technique" "Methodology" "Task" "Tool" "System" "Application" "Environment" "Workbench" "Problem" "Issue" "Challenge" "Success Stories"

### Para Scopus

- TITLE-ABS-KEY(("Project" OR "Development" OR "Organization" OR "Enterprise" OR "Company" OR "Industry" OR "Institute" OR "Research Group" OR "Technology Center") AND ("Requirements Traceability" OR "Requirements Tracing") AND ("Model" OR "Process" OR "framework" OR "Method" OR "Technique" OR "Methodology" OR "Task" OR "Tool" OR "System" OR "Application" OR "Environment" OR "Workbench" OR "Problem" OR "Issue" OR "Challenge" OR "Success Stories")) AND (LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009) OR LIMIT-TO(PUBYEAR, 2008) OR LIMIT-TO(PUBYEAR, 2007) OR LIMIT-TO(PUBYEAR, 2006) OR

LIMIT-TO(PUBYEAR, 2005) OR LIMIT-TO(PUBYEAR, 2004) OR  
LIMIT-TO(PUBYEAR, 2003).

Com base nos critérios de seleção de fontes e nas restrições definidas na seção 5, tem-se a seguinte listagem de fontes:

- IEEEXplore Digital Library
- El Compendex
- ISI Web of Knowledge
- ACM
- Scopus
- Anais do Wamps
- Anais do WER
- Anais do SBQS

## **6. Tipo dos Artigos**

Serão consideradas pesquisas do tipo experimental, teórica, relato de experiência industrial, revisões da literatura e relatórios técnicos na forma de artigos completos de periódicos ou conferências.

## **7. Idioma dos Artigos**

Serão considerados artigos escritos em português e inglês, a primeira por ser importante considerar pesquisas de âmbito nacional, dada a relevância do Programa MPS.BR para o estudo, e a segunda para expandir a abrangência da pesquisa, uma vez que é a língua definida como padrão na grande maioria dos periódicos e conferências internacionais.

## **8. Critérios de Inclusão e Exclusão dos Artigos**

Os critérios de exclusão dos artigos:

- CE.1)** Artigos que não estejam disponíveis livremente para consulta ou download (em versão completa) através das fontes de pesquisa ou através de busca manual (para artigos que não sejam fornecidos

na íntegra) realizada nas ferramentas de busca Google (<http://www.google.com.br/>) e/ou Google Scholar (<http://scholar.google.com.br/>);

- CE.2)** Artigos que claramente não atendam as questões de pesquisa;
- CE.3)** Artigos repetidos (em mais de uma fonte de busca) terão apenas sua primeira ocorrência considerada;
- CE.4)** Artigos duplicados terão apenas sua versão mais recente ou a mais completa considerada, salvo casos em que haja informações complementares.
- CE.5)** Estudos enquadrados como resumos, keynote speeches, cursos, tutoriais, workshops e afins;
- CE.6)** Artigos que não mencionem as palavras-chave da pesquisa no título, resumo ou nas palavras-chave do artigo, salvo trabalhos que abordem melhoria do processo de software nos quais seja observada possibilidade da Rastreabilidade de Requisitos ser tratada ao longo do trabalho.
- CE.7)** Excluir se o estudo não estiver inserido no contexto de Projetos de Software, Indústria de Software ou Engenharia de Software.
- CE.8)** Excluir se o estudo não estiver apresentado em uma das linguagens aceitas (Inglês e Português);

Os critérios de inclusão dos artigos se baseiam em:

- CI.1)** Estudos que apresentem primária ou secundariamente abordagens (padrões e CASEs) de apoio às atividades de Rastreabilidade de Requisitos;
- CI.2)** Estudos que apresentem relatos de experiência da indústria, ou pesquisas de caráter experimental ou teórico, contanto que apresentem exemplos de aplicação, descrição de experimentos ou casos reais de uso de abordagens (padrões e CASEs) para apoio às atividades de Rastreabilidade de Requisitos.

## **9. Critérios de Qualidade dos Estudos Primários**

Em adição aos critérios gerais de inclusão e exclusão, é considerado importante avaliar a qualidade dos estudos primários (KITCHENHAM, 2004). Os critérios de

qualidade que serão aplicados aos estudos primários foram adaptados de Catarina (2010) uma vez que descreviam critérios abrangentes o suficiente para cobrir o escopo dos estudos a serem considerados, com alterações para adequarem-se aos objetivos e questões de pesquisa desta revisão sistemática.

- 1. Introdução/Planejamento**
  - a. Os objetivos ou questões do estudo são claramente definidos (incluindo justificativas para a realização do estudo)?
  - b. O tipo de estudo está definido claramente?
- 2. Desenvolvimento**
  - a. Existe uma clara descrição do contexto no qual a pesquisa foi realizada?
  - b. O trabalho é bem/adequadamente referenciado (apresenta trabalhos relacionados ou semelhantes e baseia-se em modelos e teorias da literatura)?
- 3. Conclusão**
  - a. O estudo relata de forma clara e não ambígua os resultados?
  - b. Os objetivos ou questões do estudo são alcançados?
- 4. Critérios para a Questão de Investigação**
  - a. O estudo lista primária ou secundariamente ou Modelos, Processos, Métodos, Técnicas, Metodologia e afins, para apoiar atividades de Rastreabilidade de Requisitos?
  - b. O estudo apresenta Ferramentas de apoio às atividades de Rastreabilidade de Requisitos?
- 5. Critério Específico para estudos Experimentais**
  - a. Existe um método ou um conjunto de métodos descrito para a realização do estudo?
- 6. Critério Específico para estudos Teóricos**
  - a. Existe um processo não tendencioso na escolha dos estudos?
- 7. Critério Específico para Revisões Sistemáticas**
  - a. Existe um protocolo rigoroso, descrito e seguido?
- 8. Critério Específico para Relato de Experiência Industrial**
  - a. Existe uma descrição sobre a(s) organização(ões) onde foi conduzido o estudo?

## **10. Processo de Buscas nas Fontes**

Para a execução desta revisão sistemática cada método de busca definido anteriormente seguirá os seguintes processo:

### **10.1. Processo de Busca Manual**

Durante a condução desta revisão sistemática, os estudos primários serão identificados conforme o processo seguinte:

1. Realiza-se as buscas em todos os anais das conferências selecionadas, por meio da leitura do título e abstract, e se necessário, introdução e conclusão de todos os estudos, estudos claramente irrelevantes a pesquisa são descartados.

### **10.2. Processo de Busca Automática**

1. Realiza-se as buscas em todas as fontes selecionadas, por meio de Strings de busca, estudos claramente irrelevantes a pesquisa são descartados. Os artigos são catalogados na ferramenta JabRef, estabelecendo uma planilha com a lista, para cada pesquisador, de possíveis estudos primários;

## **11. Processo de Seleção dos Estudos Primários**

Para a execução desta revisão sistemática serão utilizados os seguintes recursos:

- Dois pesquisadores (um aluno de mestrado e um graduando);
- Acesso às fontes de pesquisa por meio do domínio da Universidade Federal do Pará;
- Validações sobre documentos e procedimentos da realização da revisão sistemática através de reuniões com o coordenador do Projeto SPIDER e orientador do projeto de dissertação, onde esta pesquisa está inserida, o Prof. Dr. Sandro Ronaldo Bezerra Oliveira;

Durante a condução desta revisão sistemática, os estudos primários serão identificados conforme o processo seguinte:

1. A partir da leitura de resumo, introdução e conclusão, os artigos deverão ser avaliados quanto aos critérios de inclusão e exclusão, e o resultado deve ser registrado.
2. Os dois pesquisadores responsáveis pela seleção dos artigos devem entrar em consenso, quando necessário. Isso ocorre quando não há uma unanimidade na inclusão de um artigo e pelo menos um pesquisador decidir incluir.
3. Na fase de consenso, em caso de discordância sobre a inclusão de algum estudo, o mesmo deverá ser incluído.
4. Os estudos primários identificados serão posteriormente lidos em totalidade e então será aplicada a avaliação de qualidade e a estratégia de extração de dados, conforme descrito nas subseções seguintes.

## **12. Avaliação da Qualidade dos Estudos Primários**

A qualidade de um artigo pode ser mensurada pela relevância e valor científico de seu conteúdo; também é considerado um critério de exclusão, a ser aplicado durante a condução da pesquisa. Desta forma, durante a análise dos estudos primários e coleta de resultados, serão aplicados os critérios de qualidade (definidos na seção 12), permitindo um processo adicional de validação dos estudos, de forma a identificar possíveis trabalhos que ainda devem ser desconsiderados da pesquisa e observar o grau de importância dos estudos individualmente para quaisquer comparações durante a síntese dos dados (KITCHENHAM, 2005).

Adicionalmente, a avaliação da qualidade pode servir como recomendação de estudos para futuras pesquisas, fornecendo informações a respeito da qualidade das informações de cada estudo avaliado (KITCHENHAM, 2005).

Artigos que, porventura, sejam excluídos por não se adequarem aos critérios de qualidade devem ser citados, juntamente com as razões para sua exclusão. Após este passo, os artigos incluídos na pesquisa passarão pela fase de extração de dados.

Os critérios (1) a (4) se aplicam a todos os estudos primários avaliados, enquanto os critérios (5) a (8) se aplicam especificamente aos respectivos tipos de trabalho mencionados.

### 12.1. Processo de Avaliação de Qualidade

Os estudos primários selecionados são lidos em totalidade e então são avaliados quanto aos critérios de qualidade. Para avaliar o grau de adequação aos critérios de qualidade, será adotada a estratégia de avaliação proposta por Costa (2010), onde se utiliza a escala de Likert-5, permitindo respostas gradativas de 0 (discordo totalmente) à 4 (concordo totalmente), como apresentada no quadro 3.1. Para auxiliar a avaliação seguindo a escala de Likert-5 para cada critério de qualidade foram definidas escalas, como se pode observar no quadro 3.2.

<b>Escala de Likert-5</b>	
Concordo totalmente (4)	Deve ser concedido no caso em que o trabalho apresente no texto os critérios que atendam totalmente a questão.
Concordo parcialmente (3)	Deve ser concedido no caso em que o trabalho atenda parcialmente aos critérios da questão.
Neutro (2)	Deve ser concedido no caso em que o trabalho não deixe claro se atende ou não a questão;
Discordo parcialmente (1)	Deve ser concedido no caso em que os critérios contidos na questão não são atendidos pelo trabalho avaliado;
Discordo totalmente (0)	Deve ser concedido no caso em que não existe nada no trabalho que atenda aos critérios da questão.

<b>Escala por Critério</b>	
<b>Critério</b>	<b>Escala</b>
1a.	4 - Define e justifica o estudo claramente. 3 - Define claramente o estudo, porém a justificativa não é clara. 2 - Define claramente o estudo, mas não justifica. 1 - A definição dos objetivos do estudo não é clara. 0 - Não define o estudo.
1b.	4 - Define o tipo de estudo, referenciando na literatura a metodologia. 3 - Define o tipo de estudo, porém sem referenciar a metodologia. 2 - Não define o tipo de estudo. É possível inferir facilmente. 1 - Não define o tipo de estudo. É possível inferir com dificuldade.



	0 – Não é possível inferir o tipo de estudo.
2a.	4 – Define claramente uma seção com o contexto da pesquisa. 3 – O contexto da pesquisa está incluído em uma seção não exclusiva. 2 – O contexto da pesquisa está disperso ao longo do texto. 1 – O contexto da pesquisa está disperso e é insubstancial. 0 – O contexto da pesquisa não é abordado.
2b.	4 – O texto apresenta uma seção de trabalhos relacionados. 3 – O texto apresenta trabalhos relacionados em uma seção não exclusiva. 2 – O texto apresenta trabalhos relacionados dispersos ao longo do texto. 1 – O texto não apresenta trabalhos relacionados, mas se apoia na literatura. 0 – O texto não apresenta trabalhos relacionados nem se apoia na literatura.
3a.	4 – Resultados são claramente apresentados na seção de conclusão. 3 – Resultados são claramente referenciados na seção de conclusão. 2 – Resultados apresentados na conclusão não são claros. 1 – Resultados referenciados na conclusão não são claros. 0 – Não são apresentados resultados.
3b.	4 – Os resultados estão totalmente aderentes ao objetivo do estudo. 3 – Os resultados estão aderentes ao objetivo do estudo, no entanto o autor faz ressalvas. 2 – Os resultados são parcialmente aderentes ao objetivo do estudo. 1 – Os resultados não estão aderentes ao objetivo do estudo. 0 – Não é alcançado nenhum resultado.
4a.	4 – Algum dos elementos é claramente descrito. 3 – Algum dos elementos é apresentado, mas não aprofundado. 2 – Algum dos elementos é avaliado, porém não descrito. 1 – Algum dos elementos é citado indiretamente. 0 – Nenhum dos elementos é apresentado direta ou indiretamente.
4b.	4 – Apresenta uma ferramenta disponível para uso em versão funcional. 3 – Apresenta ferramenta em versão funcional, porém não disponível para uso. 2 – Apresenta um projeto ou proposta de ferramenta, incluindo a descrição de funcionalidades. 1 – Apresenta apenas algoritmos de apoio ou análises de ferramentas de suporte à Rastreabilidade de Requisitos.

	0 – Não são apresentadas ferramentas de apoio.
5a.	4 – O método de experimento é definido e referenciado claramente. 3 – O método de experimento é definido claramente. 2 – O método de experimento é citado. 1 – O método de experimento não é citado, porém é possível inferir. 0 – Não é possível inferir o método de experimento.
6a.	4 – O texto descreve critérios para a escolha dos estudos. 3 – O texto não descreve critérios para a escolha dos estudos, porém apresenta estudos que discordam do estudo apresentado. 2 – O texto descreve apenas estudos aderentes ao estudo apresentado. 1 – O texto descreve estudos insuficientes. 0 – O texto não descreve estudos base.
7a.	4 – O protocolo de revisão é apresentado, descrito e seguido. 3 – O protocolo de revisão é apresentado e descrito, porém há evidências de que não foi seguido adequadamente. 2 – O protocolo de revisão não foi suficientemente descrito. 1 – O protocolo de revisão apenas foi citado ao longo do texto. 0 – Não há um protocolo de revisão.
8a.	4 – A área de atuação, tamanho e origem da organização são informados. 3 – Apenas duas das características do item 4 são informadas. 2 – Apenas uma das características do item 4 é informada. 1 – Nenhuma das características do item 4 é informada. 0 – O estudo não foi conduzido em uma ou mais organizações.

1. Deve-se então, dada à soma das pontuações, enquadrá-los em um dos cinco níveis de qualidade classificados por Beecham (2007), tais como, apresentados na tabela abaixo.

<b>Faixa de Notas</b>	<b>Avaliação</b>
Excelente	>86%
Muito Boa	66%-85%
Boa	46%-65%
Média	26%-45%
Baixa	< 26%

### **13. Estratégia de Extração de Informações**

Cada publicação selecionada é avaliada com relação ao tipo de abordagem que ela utiliza. Esta aderência é registrada na tabela Avaliação da Qualidade.

#### **13.1. Extração de resultados**

Para cada estudo primário analisado durante a etapa de extração de resultados, será feito um resumo (já identificando o que é proposto: modelo, método, processo, framework, apoio sistematizado, etc) e serão observados os detalhes referentes as questões secundárias da questão (ver 4.2). Desta forma, será analisado:

##### **13.1.1. Q1A – Qual o contexto de aplicação da abordagem encontrada?**

A abrangência da abordagem (referente a questão de pesquisa Q1) será analisada em relação ao contexto de desenvolvimento onde foi encontrada a abordagem (acadêmico, industrial) e depois tipo de metodologia utilizada (ágil, tradicional)

##### **13.1.2. Q1B – Quais as formas de rastreabilidade (para frente e para trás) são cobertas pela abordagem?**

Para cada abordagem encontrada devem ser identificada a forma de rastreabilidade que ela implementa. Pretende-se identificar informações referentes a bi-direcionalidade da rastreabilidade.

##### **13.1.3. Q1C – Quais os tipos de rastreabilidade (vertical e horizontal) são cobertos pela abordagem?**

Para cada abordagem identificada nos estudos, será analisado o tipo de rastreabilidade que a mesma implementa, a fim de verificar se a rastreabilidade oferecida refere-se ao relacionamento entre artefatos do mesmo tipo ou de tipos diferentes

##### **13.1.4. Q1D – Quais os ativos (papéis, artefatos) envolvidos?**

Para cada abordagem encontrada nos estudos, verificar quais os ativos envolvidos em seu desenvolvimento.

### **13.1.5. Q1E – Quais técnicas são cobertas pelo apoio sistematizado?**

Verificar entre as técnicas encontradas nos estudos, quais possuem apoio sistematizado.

### **13.1.6. Q1F – Quais Softwares foram utilizados?**

Deve-se registrar cada software mencionado no estudo, com uma breve descrição do mesmo (se houver).

### **13.1.7. Q1G – Qual a licença dos softwares utilizados?**

Para cada software descrito em Q1F deve-se identificar se este é Livre, Gratuito (porém não tem código aberto) ou Proprietário.

## **Sumarização dos Resultados**

### **13.2. Disposição das Informações**

O principal gráfico gerado em relação à questão de pesquisa é o da quantidade de artigos que abordam uma determinada técnica. Outros gráficos podem ser desenvolvidos ao longo da Extração dos Resultados, como, por exemplo:

- Quantidade e classificação de ferramentas analisadas;
- Quantidade de artigos selecionados por base de dados;
- Evolução ao longo do tempo da quantidade de estudos selecionados;
- Quantidade de estudos por tipo (Experimental, Teórico, Revisão Sistemática ou Relato de Experiência);

## APÊNDICE B – ESTUDOS INCLUÍDOS NA PESQUISA

<b>ID</b>	<b>Título</b>	<b>Ano de Publicação</b>	<b>Fonte</b>
EI1	5 th International Workshop on Traceability in Emerging Forms of Software Engineering (EFSE 2009)	2009	ACM
EI2	Improving traceability link recovery methods through software artifact summarization	2011	ACM
EI3	Towards Practical Software Traceability	2008	ACM
EI4	Capturing custom link semantics among heterogeneous artifacts and tools	2009	ACM
EI5	Trust in MDE Components : the DOMINO Experiment	2010	ACM
EI6	Tracing Software Product Line Variability – From Problem to Solution Space	2005	ACM
EI7	A subset of precise UML for Model-based Testing	2007	ACM
EI8	Extraction and visualization of traceability relationships between documents and source code	2010	ACM
EI9	Grand Challenges , Benchmarks , and TraceLab : Developing Infrastructure for the Software Traceability Research Community	2011	ACM
EI10	Traceability and completeness checking for agent-oriented systems	2008	ACM
EI11	Traceability Challenge 2011 : Using TraceLab to Evaluate the Impact of Local versus Global IDF on Trace Retrieval	2011	ACM
EI12	IR-Based Traceability Recovery Processes: An Empirical Comparison of "One-Shot" and Incremental Processes	2008	ACM
EI13	Make the Most of Your Time: How Should the Analyst Work with Automated Traceability Tools?	2007	ACM
EI14	Towards a model of analyst effort for traceability research	2011	ACM
EI15	Traceability visualization in metamodel change impact detection	2013	ACM
EI16	Surfing the net for software engineering notes	2006	ACM
EI17	A state-based approach to traceability maintenance	2010	ACM
EI18	Clustering support for automated tracing	2007	ACM
EI19	Engineering of requirements for a distributed teleteaching system	2013	ACM
EI20	Resolving Uncertainties during Trace Analysis	2004	ACM

EI21	A Formal Approach to Reuse Successful Traceability Practices in SPL Projects Categories and Subject Descriptors	2010	ACM
EI22	Recovering traceability links between feature models and source code of product variants	2012	ACM
EI23	Code patterns for automatically validating requirements-to-code traces	2012	ACM
EI24	Observations on the connectedness between requirements-to-code traces and calling relationships for trace validation	2011	ACM
EI25	Towards mining replacement queries for hard-to-retrieve traces	2010	ACM
EI26	Integrated solution engineering	2006	ACM
EI27	A Generic Traceability Framework for Facet-based Traceability Data Extraction in Model-driven Software Development	2010	ACM
EI28	Recovering and using use-case-diagram-to-source-code traceability links	2007	ACM
EI29	Extending Argumentation to Goal-oriented Requirements Engineering	2007	ACM
EI30	Humans in the Traceability Loop : Can ' t Live With ' Em , Can ' t Live Without ' Em	2005	ACM
EI31	Software Verification and Validation Research Laboratory (SVVRL) of the University of Kentucky: Traceability Challenge 2011: Language Translation	2011	ACM
EI32	Risk management with enhanced tracing of requirements rationale in highly distributed projects	2006	ACM
EI33	A case study on value-based requirements tracing	2005	ACM
EI34	Incremental Latent Semantic Indexing for Automatic Traceability Link Evolution Management	2008	ACM
EI35	Tracing the Rationale Behind UML Model Change Through Argumentation	2007	ACM
EI36	Automated Software Tool Support for Checking the Inconsistency of Requirements	2009	ACM
EI37	TraceLab: An experimental workbench for equipping researchers to innovate, synthesize, and comparatively evaluate traceability solutions	2012	ACM
EI38	A reusable traceability framework using patterns	2005	ACM
EI39	Traceclipse : An Eclipse Plug-in for Traceability Link Recovery and Management	2011	ACM
EI40	How Do We Trace Requirements ? An Initial Study of Analyst Behavior in Trace Validation Tasks	2011	ACM
EI41	Formalizing traceability relations for product	2011	ACM

	lines		
EP42	A Transformation Instance-Based Approach to Traceability	2010	ACM
EI43	Feature location via information retrieval based filtering of a single scenario execution trace	2007	ACM
EI44	Concern tracing and change impact analysis: An exploratory study	2009	ACM
EI45	Enabling Automated Traceability Maintenance by Recognizing Development Activities Applied to Models	2008	ACM
EI46	Source code indexing for automated tracing	2011	ACM
EI47	Separation of scattered concerns	2011	ACM
EI48	TQL: A query language to support traceability	2009	ACM
EI49	An XML based approach to support the evolution of model-to-model traceability links	2005	ACM
EI50	When and how to visualize traceability links?	2005	ACM
EI51	Combining textual and structural analysis of software artifacts for traceability link recovery	2009	ACM
EI52	Tracing Architecturally Significant Requirements : A Decision-Centric Approach	2011	ACM
EI53	Transforming trace information in architectural documents into re-usable and effective traceability links	2011	ACM
EI54	Tracing architectural concerns in high assurance systems	2011	ACM
EI55	A tactic-centric approach for automating traceability of quality concerns	2012	ACM
EI56	TRACING VARIATIONS IN SOFTWARE PRODUCT FAMILIES	2007	ACM
EI57	Towards traceability of model-based testing artifacts	2007	ACM
EI58	Departures from optimality: Understanding human analyst's information foraging in assisted requirements tracing	2013	ACM
EI59	Enhancing traceability using ontologies	2007	ACM
EI60	Change impact analysis from business rules	2010	ACM
EI61	Cross-Artifact Traceability Using Lightweight Links	2009	ACM
EI62	Automating Traceability for Generated Software Artifacts	2004	ACM
EI63	EA-Tracer: Identifying Traceability Links between Code Aspects and Early Aspects	2012	ACM
EI64	Supporting Requirements in a Traceability Approach between Business Process and User Interfaces	2008	ACM
EI65	UML formalization is a traceability problem	2005	ACM
EI66	Baselines in requirements tracing	2005	ACM



EI67	Modeling traceability of concerns in architectural views	2007	ACM
EI68	Scenarios for System Requirements Traceability and Testing	2011	ACM
EI69	Analysis of crosscutting across software development phases based on traceability	2006	ACM
EI70	Trace Retrieval for Evolving Artifacts	2009	ACM
EI71	Tracing requirements to tests with high precision and recall	2011	ACM
EI72	STRADA : A Tool for Scenario-based Feature-to-Code Trace Detection and Analysis	2007	ACM
EI73	Recovering traceability links in software artifact management systems using information retrieval methods	2007	ACM
EI74	Theme : An Approach for Aspect-Oriented Analysis and Design	2004	Compendex
EI75	Towards a benchmark for traceability	2011	Compendex
EI76	Regression test selection on system requirements	2008	Compendex
EI77	Goal-centric traceability for managing non-functional requirements	2005	Compendex
EI78	A Value-Based Approach for Understanding Cost-Benefit Trade-Offs During Automated Software Traceability	2005	Compendex
EI79	Determining the Cost-Quality Trade-Off for Automated Software Traceability	2005	Compendex
EI80	GoPoMoSA : A Goal-Oriented Process Modeling and Simulation Advisor	2009	Compendex
EI81	A traceability semantics approach for supporting product value analysis	2010	Compendex
EI82	Tool support for generation and validation of traces between requirements and architecture	2010	Compendex
EI83	An Approach of Requirements Tracing in Formal Refinement	2010	Compendex
EI84	Improving Trace Accuracy through Data-Driven Configuration and Composition of Tracing Features	2013	Compendex
EI85	Reconstructing requirements coverage views from design and test using traceability recovery via LSI	2005	Compendex
EI86	On Structuring Formal, Semi-Formal and Informal Data to Support Traceability in Systems Engineering Environments	2004	Compendex
EI87	Improving the change management process	2008	Compendex
EI88	OO techniques applied to a real-time, embedded, spaceborne application	2006	Compendex
EI89	Using scenarios to support traceability	2005	Compendex
EI90	Problem-solution mapping for forward and reengineering on architectural level	2011	Compendex

EI91	Traceability in Viewpoint Merging : A Model Management Perspective	2005	Compendex
EI92	A comparative evaluation of two user feedback techniques for requirements trace retrieval	2012	Compendex
EI93	Recovering traceability links between requirements and source code in the same series of software products	2013	Compendex
EI94	Test Intents : Enhancing the Semantics of Requirements Traceability Links in Test Cases	2013	Compendex
EI95	An end-to-end industrial software traceability tool	2007	Compendex
EI96	Do better IR tools improve the accuracy of engineers' traceability recovery?	2011	Compendex
EI97	Requirements Traceability in Automated Test Generation - Application to Smart Card Software Validation	2005	Compendex
EI98	Formal Derivation of Security Design Specifications from Security Requirements	2008	Compendex
EI99	A rich traceability model for social interactions	2011	Compendex
EI100	Agile Requirements Traceability Using Domain-Specific Modelling Languages	2012	Compendex
EI101	Traceability between function point and source code	2011	Compendex
EI102	Tooling for the full traceability of non-functional requirements within model-driven development	2010	Compendex
EI103	Explicit use-case representation in object-oriented programming languages	2011	Compendex
EI104	Towards overcoming human analyst fallibility in the requirements tracing process	2011	Compendex
EI105	Engineering Data Model for Machine Automation Systems	2011	Compendex
EI106	A Pragmatic Approach to Traceability in Model-Driven Development	2009	Compendex
EI107	Trustrace : Mining Software Repositories to Improve the Accuracy of Requirement Traceability Links	2013	Compendex
EI108	An automated mechanism for organizing and retrieving core asset artifacts for product derivation in SPL	2009	Compendex
EI109	Towards a Subject-Oriented Model-Driven Framework	2006	Compendex
EI110	Model-Based Design for Large High-Integrity Systems : A Discussion on Verification and Validation	2010	Compendex
EI111	Maintainability-Based Requirements Prioritization by Using Artifacts Traceability and Code Metrics	2013	Compendex
EI112	Automatic Tracing of Decisions to Architecture	2011	Compendex

	and Implementation		
EI113	An Enhanced Architectural Knowledge Metamodel Linking Architectural Design Decisions to Other Artifacts in the Software Engineering Lifecycle	2011	Compendex
EI114	Towards Metamodel Support for Variability and Traceability in Software Product Lines Categories and Subject Descriptors	2011	Compendex
EI115	Visual Support In Automated Tracing	2007	Compendex
EI116	Trace Queries for Safety Requirements in High Assurance Systems	2012	Compendex
EI117	Re-engineering legacy Web applications into RIAs by aligning modernization requirements, patterns and RIA features	2013	Compendex
EI118	Design Traceability	2005	Compendex
EI119	Integrated tool support for hardware-related software development	2009	Compendex
EI120	SafeSlice : A Model Slicing and Design Safety Inspection Tool for SysML	2011	Compendex
EI121	Traceability in Model-Based Testing	2012	Compendex
EI122	Model- Based System of Systems Engineering with UPDM	2010	Compendex
EI123	Bridging the Chasm - Tracing from Architectural Frameworks to SysML	2007	Compendex
EI124	Helping Analysts Trace Requirements : An Objective Look requirements tracing tool	2004	Compendex
EI125	Traceability-based Change Awareness	2009	Compendex
EI126	Designing and Implementing a Tool for Distributed Collaborative Traceability and Rationale Management	2008	Compendex
EI127	IMPLEMENTING A DOCUMENT-BASED REQUIREMENTS TRACEABILITY : A CASE STUDY	2005	Compendex
EI128	Why Software Requirements Traceability Remains a Challenge	2009	Compendex
EI129	SPECIFYING AND SEPARATING CONCERNS FROM REQUIREMENTS TO DESIGN: A CASE STUDY	2005	Compendex
EI130	INTEGRATING AN ECLIPSE-BASED SCENARIO MODELING ENVIRONMENT WITH A REQUIREMENTS MANAGEMENT SYSTEM	2006	Compendex
EI131	An Approach to Feature Based Modelling by Dependency Alignment for the Maintenance of the Trustworthy System	2004	Compendex
EI132	Supporting commonality and variability analysis of requirements and structural models	2012	Compendex
EI133	Tool Support for Traceable Product Evolution	2004	Compendex

EI134	Means-ends and whole-part traceability analysis of safety requirements	2010	Compendex
EI135	ARARA : ARTIFACTS AND REQUIREMENTS AWARENESS REINFORCEMENT AGENTS	2008	Compendex
EI136	Customizing Traceability Links for the Unified Process	2007	Compendex
EI137	Traceability for Managing Evolutionary Change	2006	Compendex
EI138	A trace metamodel proposal based on the model driven architecture framework for the traceability of user requirements in data warehouses	2012	Compendex
EI139	Improved Representation of Traceability Links in Requirements Engineering Knowledge using Sunburst and Netmap Visualizations	2011	Compendex
EI140	Automating the Trace of Architectural Design Decisions and Rationales Using a MDD Approach	2008	Compendex
EI141	Tool support for computer-aided requirement traceability in architectural design: The case of DesignTrack	2007	Compendex
EI142	CASE STUDIES TO EXPLORE INDEXING ISSUES IN PRODUCT DESIGN TRACEABILITY FRAMEWORK	2011	Compendex
EI143	An Audit Model for ISO 9001 Traceability Requirements in Agile-XP Environments	2013	Compendex
EI144	The Barriers to Traceability and their Potential Solutions: Towards a Reference Framework	2012	Compendex
EI145	Tagging knowledge acquisition sessions to facilitate knowledge traceability	2004	Compendex
EI146	Towards Identification and Recognition of Trace Associations in Software Requirements Traceability	2012	Compendex
EI147	VISUALIZING CHANGING REQUIREMENTS WITH SELF-ORGANIZING MAPS	2004	Compendex
EI148	An Evaluation of Requirements Management and Traceability Tools	2011	Compendex
EI149	Model-Driven User Requirements Specification using SysML	2008	Compendex
EI150	From requirements change to design change: a formal path	2004	Compendex
EI151	ATRAM-An Analysis Tool of Requirement and Architecture Management	2003	Compendex
EI152	Model-based Concurrent Systems Design for Safety	2004	Compendex
EI153	Using a formal requirements management tool for System Engineering: first results at ESO	2006	Compendex

EI154	Tracing Software Requirements Artefacts	2003	Compendex
EI155	Semi-automatic generation of UML models from natural language requirements	2011	Compendex
EI156	Retrieval by construction: a traceability technique to support verification and validation of uml formalizations	2005	Compendex
EI157	Identifying Requirements Conflicts and Cooperation: How Quality Attributes and Automated Traceability Can Help	2004	Compendex
EI158	a Framework for Comparing Requirements Tracing Experiments	2005	Compendex
EI159	Improving automated requirements trace retrieval: a study of term-based enhancement methods	2009	Compendex
EI160	A lightweight traceability assessment method for medical device software	2013	Compendex
EI161	A Step-wise Approach for Integrating QoS throughout Software Development	2011	Compendex
EI162	Do Data Dependencies in Source Code complement Call Dependencies for Understanding Requirements Traceability ?	2012	Compendex
EI163	A scoped approach to traceability management	2009	Compendex
EI164	Semantic decoupling: reducing the impact of requirement changes	2010	Compendex
EI165	Ontological Traceability over the Unified Process	2007	Compendex
EI166	Medical device standards' requirements for traceability during the software development lifecycle and implementation of a traceability assessment model	2013	Compendex
EI167	Improving Software Quality through Requirements Traceability Models	2006	Compendex
EI168	Graph-based traceability: a comprehensive approach	2009	Compendex
EI169	From Software Traceability to Global Model Management and Back Again	2011	Compendex
EI170	Domain model-driven software engineering: A method for discovery of dependency links	2012	Compendex
EI171	A survey of traceability in requirements engineering and model-driven development	2009	Compendex
EI172	Utilizing supporting evidence to improve dynamic requirements traceability	2005	Compendex
EI173	Supporting software understanding with automated requirements traceability	2005	Compendex
EI174	Semantics of trace relations in requirements models for consistency checking and inferencing	2009	Compendex
EI175	Effective Requirements Traceability : Models , Tools , and Practices	2008	Compendex



EI176	An industrial case study in reconstructing requirements views	2008	Compendex
EI177	A visual language for modeling and executing traceability queries	2012	Compendex
EI178	Towards automated traceability maintenance.	2012	Compendex
EI179	Transforming and tracing reused requirements models to home automation models	2013	Compendex
EI180	Exploring a Bayesian and linear approach to requirements traceability	2011	Compendex
EI181	A streamlined, cost-effective database approach to manage requirements traceability	2011	Compendex
EI182	Rule-based generation of requirements traceability relations	2004	Compendex
EI183	Introducing requirements traceability support in model-driven development of web applications	2009	Compendex
EI184	A machine learning approach for tracing regulatory codes to product specific requirements	2010	IEEE
EI185	An Ontology-Based Approach for Multiperspective Requirements Traceability between Analysis Models	2010	IEEE
EI186	Model-Based Traceability	2009	IEEE
EI187	MUPRET: An Ontology-Driven Traceability Tool for Multiperspective Requirements Artifacts	2009	IEEE
EI188	SEURAT : Integrated Rationale Management	2008	IEEE
EI189	Requirements Tracing : Discovering Related Documents through Artificial Pheromones and Term Proximity	2011	IEEE
EI190	Software traceability with topic modeling	2010	IEEE
EI191	Improving automated documentation to code traceability by combining retrieval techniques	2011	IEEE
EI192	ADAMS Re-Trace : Traceability Link Recovery via Latent Semantic Indexing	2008	IEEE
EI193	Towards Traceable Test-Driven Development	2009	IEEE
EI194	Towards traceability from project management to system models	2009	IEEE
EI195	Challenges for semi-automatic trace recovery in the automotive domain	2009	IEEE
EI196	Tools for Traceability in Secure Software Development	2008	IEEE
EI197	Maintaining invariant traceability through bidirectional transformations	2012	IEEE
EI198	A combination approach for enhancing automated traceability	2011	IEEE
EI199	Getting back to basics: Promoting the use of a traceability information model in practice	2009	IEEE
EI200	Using traceability mechanisms to support	2004	IEEE

	software product line evolution		
EI201	Experiments on Processing and Linking Semantically Augmented	2004	IEEE
EI202	Resolving Multiperspective Requirements Traceability Through Ontology Integration	2008	IEEE
EI203	Breaking the big-bang practice of traceability: Pushing timely trace recommendations to project stakeholders	2012	IEEE
EI204	Effort and Quality of Recovering Requirements-to-Code Traces: Two Exploratory Experiments	2010	IEEE
EI205	MV - TMM: A Multi View Traceability Management Method	2008	IEEE
EI206	Case-based stories for traceability education and training	2012	IEEE
EI207	An Ontology Based Improved Software Requirement Traceability Matrix	2009	IEEE
EI208	Tracing and Mapping : Supporting Software Quality Predictions	2005	IEEE
EI209	A Case Study on the Evolution of a Component-based Product Line	2012	IEEE
EI210	Requirements Management Tool with Evolving Traceability for Heterogeneous Artifacts in the Entire Life Cycle	2010	IEEE
EI211	Extension Features-Driven Use Case Model for requirement traceability	2009	IEEE
EI212	Can LSI help reconstructing requirements traceability in design and test?	2006	IEEE
EI213	Reuse through Requirements Traceability	2008	IEEE
EI214	Requirement Traceability in Software Development Process: An Empirical Approach	2008	IEEE
EI215	Hidden Implementation Dependencies in High Assurance and Critical Computing Systems	2006	IEEE
EI216	A Proposal for Defining a Set of Basic Items for Project-Specific Traceability Methodologies	2008	IEEE
EI217	Modeling Functional Requirements to Support Traceability Analysis	2006	IEEE
EI218	Towards End-to-End Traceability: Insights and Implications from Five Case Studies	2009	IEEE
EI219	Poirot : A Distributed Tool Supporting Enterprise-Wide Automated Traceability	2006	IEEE
EI220	A means of establishing traceability based on a UML model in business application development	2011	IEEE
EI221	Traceability for the maintenance of secure software	2008	IEEE
EI222	Documenting Requirements Traceability Information for Small Projects	2007	IEEE



EI223	Deriving Traceability Relationships of Multiperspective Software Artifacts from Ontology Matching	2009	IEEE
EI224	Event-Based Traceability for Managing Evolutionary Change	2003	IEEE
EI225	A heterogeneous solution for improving the return on investment of requirements traceability	2004	IEEE
EI226	NEXT-GENERATION TECHNIQUES FOR TRACKING DESIGN REQUIREMENTS COVERAGE IN AUTOMATIC TEST SOFTWARE	2006	IEEE
EI227	Assessing the effect of requirements traceability for software maintenance	2012	IEEE
EI228	Semi-automated traceability maintenance: An architectural overview of traceMaintainer	2009	IEEE
EI229	Rule-Based Maintenance of Post-Requirements Traceability Relations	2008	IEEE
EI230	Use Case to Source Code Traceability: The Developer Navigation View Point	2009	IEEE
EI231	Requirements Traceability: a Systematic Review and Industry Case Study	2012	IEEE
EI232	A Value Estimation Method for Feature-Oriented Requirements Tracing	2008	IEEE
EI233	3CI: A Tool for Crosscutting Concern Identification	2008	IEEE
EI234	Smarter Architecture & Engineering: Game changer for requirements management: A position paper	2011	IEEE
EI235	Towards Medical Device Behavioural Validation Using Petri Nets	2013	IEEE
EI236	Alignment of Requirements Specification and Testing: A Systematic Mapping Study	2011	IEEE
EI237	TraceME: Traceability Management in Eclipse	2012	IEEE
EI238	Issues in Testing Dependable Event-Based Systems at a Systems Integration Company 3 . CASE 1 : Test of an Interlocking Subsystem	2007	IEEE
EI239	A unified requirements model ; integrating features , use cases , requirements , requirements analysis and hazard analysis Chair for Applied Software Engineering	2007	IEEE
EI240	In Vivo Evaluation of Large-Scale IR-Based Traceability Recovery	2011	IEEE
EI241	Enabling traceability reuse for impact analyses: A feasibility study in a safety context	2013	IEEE
EI242	Evaluation of Traceability Recovery in Context : A Taxonomy for Information Retrieval Tools	2012	IEEE
EI243	Traceability management framework for patient data in healthcare environment	2010	IEEE

EI244	Requirements Engineering Tools	2011	IEEE
EI245	The Software V&V Tasks for a Safety-Critical Software Based Protection System in Nuclear Power Plants	2005	IEEE
EI246	Best Practices for Automated Traceability	2007	IEEE
EI247	Just-in-time traceability for mechatronics systems	2012	IEEE
EI248	ADAMS : ADvanced Artefact Management System	2006	IEEE
EI249	Slicing the V-Model -- Reduced Effort, Higher Flexibility	2013	IEEE
EI250	Assisting the synchronization of UCM-based architectural documentation with implementation	2009	IEEE
EI251	A scenario-driven approach to trace dependency analysis	2003	IEEE
EI252	Feature-to-code traceability in a collection of software variants: Combining formal concept analysis and information retrieval	2013	IEEE
EI253	Linking Documentation and Source Code in a Software Chrestomathy	2012	IEEE
EI254	Wolf: Supporting impact analysis activities in distributed software development	2012	IEEE
EI255	Towards a Traceability Visualisation Tool	2012	IEEE
EI256	Semi-automatic establishment and maintenance of valid traceability in automotive development processes	2012	IEEE
EI257	Towards 2D Traceability in a Platform for Contract Aware Visual Transformations with Tolerated Inconsistencies	2006	IEEE
EI258	Requirements Tracery	2011	IEEE
EI259	Tracing Whodunit	2008	IEEE
EI260	The Quest for Ubiquity : A Roadmap for Software and Systems Traceability Research	2012	IEEE
EI261	Using Traceability to Support SOA Impact Analysis	2011	IEEE
EI262	Automated Fault Tree Generation and Risk-Based Testing of Networked Automation Systems Tanvir Hussain and Robert Eschbach	2010	IEEE
EI263	A Study on the Effect of Traceability Links in Software Maintenance	2013	IEEE
EI264	Transformation from requirement into behavior design: A review	2012	IEEE
EI265	Traceability Link Evolution Management with Incremental Latent Semantic Indexing	2007	IEEE
EI266	Supporting product line development through traceability	2005	IEEE
EI267	Relational-model based change management for non-functional requirements: Approach and	2011	IEEE

	experiment		
EI268	A Metamodel for Tracing Non-functional Requirements	2009	IEEE
EI269	Traceability Enhancement Technique through the Integration of Software Configuration Management and Individual Working Environment	2010	IEEE
EI270	Traceability map: foundations to automate for product line engineering	2005	IEEE
EI271	A representation method to simplify traceability links between software artifacts	2013	IEEE
EI272	Towards recovering and maintaining trace links for model sketches across interactive displays	2013	IEEE
EI273	QuaTrace : A Tool Environment for ( Semi- ) Automatic	2003	IEEE
EI274	Web System Trace Model Using a Web Application Architecture Framework	2005	IEEE
EI275	Traceability Challenge 2013 : Query + Enhancement for Semantic Tracing ( QuEST )	2013	IEEE
EI276	Trace Matrix Analyzer (TMA)	2013	IEEE
EI277	Ontology-based trace retrieval	2013	IEEE
EI278	ARARA – A Collaborative Tool to Requirement Change Awareness	2010	IEEE
EI279	Information Systems Modeling with Use Cases	2007	IEEE
EI280	Monitoring Requirements Evolution using Views	2007	IEEE
EI281	A Requirement Tool to Support Model-Based Requirement Engineering	2008	IEEE
EI282	Strategic Traceability for Projects	2013	IEEE
EI283	A Traceability Link Model for the Unified Process	2007	IEEE
EI284	Context-Sensitive Traceability Controlling	2013	IEEE
EI285	Using audio and collaboration technologies for distributed requirements elicitation and documentation	2010	IEEE
EI286	A Metamodeling Approach to Tracing Variability between Requirements and Architecture in Software Product Lines	2007	IEEE
EI287	Recovering Traceability Links between Code and Documentation for Enterprise Project Artifacts	2012	IEEE
EI288	A review of traceability research at the requirements engineering conference@21	2013	IEEE
EI289	Enhanced traceability in model-based CASE tools using ontologies and information retrieval	2011	IEEE
EI290	Variation-Oriented Requirements Analysis	2007	IEEE

	(VORA)		
EI291	Requirements and Scenarios : running Aspect-Oriented Software Architectures	2007	IEEE
EI292	Establishing Content Traceability for Software Applications	2013	IEEE
EI293	REquirements TRacing On target ( RETRO ) Enhanced with an Automated Thesaurus Builder :	2013	IEEE
EI294	Linking Functional Requirements and Software Verification	2009	IEEE
EI295	Recovering Test-to-Code Traceability Via Slicing and Conceptual Coupling	2011	IEEE
EI296	SCOTCH: Test-to-code traceability using slicing and conceptual coupling	2011	IEEE
EI297	Recovering traceability links between unit tests and classes under test: An improved method	2010	IEEE
EI298	Supporting Program Comprehension in Agile with Links to User Stories	2009	IEEE
EI299	Towards Feature-Aware Retrieval of Refinement Traces	2013	IEEE
EI300	An empirical study on project-specific traceability strategies	2013	IEEE
EI301	Supporting evolutionary development by feature models and traceability links	2004	IEEE
EI302	Optimizing Design for Variability Using Traceability Links	2008	IEEE
EI303	Establishing Traceability Links between Unit Test Cases and Units under Test	2009	IEEE
EI304	Using SysML for Modeling of Safety-Critical Software-Hardware Interfaces: Guidelines and Industry Experience	2011	IEEE
EI305	Visualizing requirements in distributed system development	2012	IEEE
EI306	Towards a Comprehensive Traceability Approach in the Context of Software Maintenance	2009	IEEE
EI307	STraS: A framework for semantic traceability in enterprise-wide SOA life-cycle management	2009	IEEE
EI308	The Design of Large Scale U-IT Application USN by USN Middleware	2007	IEEE
EI309	Supporting software evolution through dynamically retrieving traces to UML artifacts	2004	IEEE
EI310	Integrated Requirement Traceability , Multiview Modeling , and Decision-Making A systems Engineering Approach for Integrating Processes and Product	2012	IEEE
EI311	Consistency Verification and Quality Assurance ( CVQA ) Traceability Framework for SaaS	2013	IEEE

EI312	Enhancement of requirements traceability with state diagrams	2010	IEEE
EI313	Application of reinforcement learning to requirements engineering: requirements tracing	2013	IEEE
EI314	Verifiable architectural interface for supporting model-driven development with adequate abstraction level	2012	IEEE
EI315	An Augmented Vector Space Information Retrieval for Recovering Requirements Traceability	2011	IEEE
EI316	Agent-based knowledge keep tracking	2003	IEEE
EI317	Linking Requirements and Testing in Practice	2008	IEEE
EI318	DRESREM 2: An Analysis System for Multi-document Software Review Using Reviewers' Eye Movements	2008	IEEE
EI319	Tool Support for Automated Traceability of Test/Code Artifacts in Embedded Software Systems	2011	IEEE
EI320	On Usability in Requirements Trace Visualizations	2008	IEEE
EI321	Ontology-based model for Rational Unified Process artifacts traceability	2012	IEEE
EI322	A Context-based Information Retrieval Technique for Recovering Use-Case-to-Source-Code Trace Links in Embedded Software Systems	2013	IEEE
EI323	Phrasing in Dynamic Requirements Trace Retrieval	2006	IEEE
EI324	Requirements Traceability and Transformation Conformance in Model-Driven Development	2006	IEEE
EI325	Overcoming the traceability benefit problem	2005	IEEE
EI326	A Multi View based Traceability Management Method	2008	IEEE
EI327	Tracing User Interface Design Pre-requirement to Generate Interface Design Specification	2009	IEEE
EI328	Motivation Matters in the Traceability Trenches	2009	IEEE
EI329	Successful Deployment of Requirements Traceability in a Commercial Engineering Organization...Really	2010	IEEE
EI330	An Evaluation of Traceability Approaches to Support Software Evolution	2007	IEEE
EI331	Use Case-Based Acceptance Testing of a Large Industrial System: Approach and Experience Report	2006	IEEE
EI332	Toward Multilevel Textual Requirements Traceability Using Model-Driven Engineering and Information Retrieval	2012	IEEE

EI333	An Integrated Decision Model For Efficient Requirement Traceability In SPICE Compliant Development	2007	IEEE
EI334	A decision model for managing and communicating resource restrictions in embedded systems design	2008	IEEE
EI335	Traceability ReARMed	2009	IEEE
EI336	Improving Source Code Lexicon via Traceability and Information Retrieval	2011	IEEE
EI337	Can Information Retrieval Techniques Effectively Support Traceability Link Recovery?	2006	IEEE
EI338	Analyzing and Systematizing Current Traceability Schemas	2006	IEEE
EI339	Traceability between Software Architecture Models	2006	IEEE
EI340	Survey of Traceability Approaches in Model-Driven Engineering	2007	IEEE
EI341	Coordinated software development: A framework for reasoning about trace links in software systems	2009	IEEE
EI342	traceMaintainer - Automated Traceability Maintenance	2008	IEEE
EI343	Examining Communication Media Selection and Information Processing in Software Development Traceability: An Empirical Investigation	2009	IEEE
EI344	ArchTrace : Policy-Based Support for Managing Evolving Architecture-to-Implementation Traceability Links	2006	IEEE
EI345	Traceability Management meets Information Retrieval Methods * – Strengths and Limitations –	2008	IEEE
EI346	Value-Based Requirements Traceability: Lessons Learned	2007	IEEE
EI347	A Comprehensive Feature-Oriented Traceability Model for Software Product Line Development	2009	IEEE
EI348	Ontological approach for the semantic recovery of traceability links between software artefacts	2008	IEEE
EI349	Just Enough Requirements Traceability	2006	IEEE
EI350	A model-driven traceability framework for software product lines	2009	ISI
EI351	Run-Time Security Traceability for Evolving Systems	2010	ISI
EI352	Automated traceability analysis for UML model refinements	2009	ISI
EI353	On human analyst performance in assisted requirements tracing: Statistical analysis	2011	ISI
EI354	Technique Integration for Requirements	2007	ISI



	Assessment		
EI355	MeDEA: A database evolution architecture with traceability	2008	ISI
EI356	Towards a framework for improving requirement traceability	2012	ISI
EI357	Traceability Patterns : An Approach to Requirement-Component Traceability in Agile Software Development The Importance of Traceability	2008	ISI
EI358	Out of the labyrinth: Leveraging other disciplines for requirements traceability	2011	ISI
EI359	Macro-Level Traceability via Media Transformations 2 Media Use in Requirements Engineering	2008	ISI
EI360	XTraQue: traceability for product line systems	2007	ISI
EI361	An effective technique for the software requirements analysis of NPP safety-critical systems, based on software inspection, requirements traceability, and formal specification	2005	ISI
EI362	Change Impact Analysis with a Goal-Driven Traceability-Based Approach	2010	ISI
EI363	A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development	2010	ISI
EI364	Fine-grained management of software artefacts : the ADAMS system	2010	ISI
EI365	Traceability-based knowledge integration in group decision and negotiation activities	2007	ISI
EI366	Improving change management in software development: Integrating traceability and software configuration management	2008	ISI
EI367	SemTrace : Semantic Requirements Tracing Using Explicit Requirement Knowledge	2011	ISI
EI368	Continuous and automated evolution of architecture-to-implementation traceability links	2007	ISI
EI369	Model-Driven Engineering as a new landscape for traceability management: A systematic literature review	2012	ISI
EI370	Application of swarm techniques to requirements tracing	2011	ISI
EI371	Assessing traceability of software engineering artifacts	2010	ISI
EI372	A rationale-based architecture model for design traceability and reasoning	2007	ISI
EI373	VbTrace: using view-based and model-driven development to support traceability in process-driven SOAs	2009	ISI



EI374	Rastreabilidade do Conhecimento como Mecanismo de Melhoria de Qualidade de Software	2007	SBQS
EI375	Suporte automatizado à rastreabilidade em um processo de teste de software baseado em documentação †	2006	SBQS
EI376	Uma abordagem para gerência de requisitos integrada com práticas ágeis de gerência de projetos	2011	SBQS
EI377	Rastreabilidade e Análise de Impacto: Um caso de aplicação utilizando a ferramenta Visual Studio Team System	2009	SBQS
EI378	Glifo para Verificação da Utilização de Processos de Requisitos por meio da Rastreabilidade como apoio à Melhoria de Qualidade de Software	2009	SBQS
EI379	Geração Automática de um Grafo de Requisitos a partir do Código da Aplicação	2011	SBQS
EI380	Toward improved traceability of non-functional requirements	2005	Scopus
EI381	Enhancing information retrieval to automatically trace requirements and design artifacts	2011	Scopus
EI382	Advanced product planning: a comprehensive process for systemic definition of new product requirements	2007	Scopus
EI383	Requirements traceability in model-driven development: Applying model and transformation conformance	2007	Scopus
EI384	A STUDY ON THE APPLICATION OF A REQUIREMENTS TRACEABILITY AUTOMATION TOOL TO THE DOCUMENTATION OF A SATELLITE PROJECT	2012	Scopus
EI385	Resolving Semantic Heterogeneity in Multiperspective Requirements Traceability Using Ontology Matching	2011	Scopus
EI386	Requirements Traceability - When and How does it Deliver more than it Costs?	2006	Scopus
EI387	From research project objectives to milestones by means of requirements traceability Realization of an autonomous maritime system	2013	Scopus
EI388	Requirements Engineering in Agile Software Development	2010	Scopus
EI389	Utilizing Multifaceted Requirement Traceability Approach: a Case Study	2011	Scopus
EI390	Requirements traceability issues: generic model, methodology and formal basis	2005	Scopus
EI391	Defining and Retrieving Themes in Nuclear Regulations	2012	Scopus

EI392	Tracing requirements to defect reports: an application of information retrieval techniques	2005	Scopus
EI393	Ontology-based multiperspective requirements traceability framework	2009	Scopus
EI394	Apoio aos Processos de Gerência de Requisitos e Verificação e Validação em um Ambiente Integrado	2011	WAMPS
EI395	Implantação do MPS.BR (Melhoria do Processo de Software Brasileiro), Nível F, com TFS (Team Foundation Server) no Desenvolvimento Eficiente de Sistemas	2009	WAMPS
EI396	GIPS - Uma Ferramenta para Gerenciamento Integrado de Projetos de Software Baseada no MPS.BR	2012	WAMPS
EI397	Um Componente para Manutenção da Rastreabilidade Bidirecional entre Casos de Uso e Código Fonte	2013	WAMPS
EI398	Um Estudo Experimental sobre Abordagens de Apoio à Rastreabilidade de Requisitos	2010	WAMPS
EI399	Análise de Ferramentas para Apoio à Gerência de Projetos e Gerência de Requisitos de Software	2010	WAMPS
EI400	Ferramenta Controlle: Um case de sucesso na implementação da Gerência de Requisitos	2012	WAMPS
EI401	Controlle: Ferramenta de Apoio à Gerência de Requisitos	2011	WAMPS
EI402	Práticas de Acompanhamento do Projeto utilizando a ferramenta Atlassian Jira no contexto do Nível F do MPS.BR	2012	WAMPS
EI403	Improving the Quality of Software Requirements Specifications with Semantic Web Technologies	2012	WER
EI404	Towards Requirement Traceability in TROPOS	2004	WER
EI405	SIGERAR: Uma Ferramenta para Gerenciamento de Requisitos	2006	WER
EI406	Uma abordagem semi-automática para a manutenção de links de rastreabilidade	2008	WER
EI407	Uma Avaliação sobre Rastreabilidade de Software no Contexto do MPS.BR	2009	WER
EI408	Uma Análise Avaliativa de Ferramentas de Software Livre no Contexto da Implementação do Processo de Gerência de Requisitos do MPS.BR	2010	WER
EI409	Um Modelo de Rastreabilidade com suporte ao Gerenciamento de Mudanças e Análise de Impacto	2010	WER
EI410	O Gerenciamento de Requisitos no Ambiente COCAR	2007	WER
EI411	A Process for Requirement Traceability in Agent Oriented Development	2005	WER

## **APÊNDICE C – FORMULÁRIO PARA AVALIAÇÃO DE ESPECIALISTA**

### **Histórico de Revisões**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
<b>26/01/2015</b>	<b>0.1</b>	<b>Elaboração do Formulário de Revisão por Pares</b>	<b>Paulo Malcher</b>
<b>30/01/2015</b>	<b>1</b>	<b>Modificação do Formulário de Revisão por Pares após Revisão</b>	<b>Paulo Malcher</b>

# Revisão por Pares

## 1. Apresentação

Esta Revisão por pares tem como foco a avaliação do Catálogo de Técnicas de apoio a Rastreabilidade de Requisitos desenvolvido como parte de uma dissertação de mestrado.

## 2. Objetivo da Revisão por Pares

Esta revisão tem como objetivo avaliar o conteúdo do catálogo de técnicas de apoio a rastreabilidade de requisitos, bem como sua estrutura; e analisar se as considerações feitas esclarecem suas atribuições e estão de acordo com seus conhecimentos.

Devem ser revisados a descrição das técnicas, tipo, fases, dimensão, estado, contexto de aplicação, exemplo de implantação, ferramentas e suas principais referências.

## 3. Instruções para a Execução da Revisão por Pares

- a) Preencha a sua Identificação e o seu Perfil como especialista em gerência de requisitos (Seções 3 e 4).
- b) Leia o catálogo de técnicas de apoio a rastreabilidade de requisitos (em anexo), analisando se os dados informados são válidos, com relação a clareza, requisitos técnicos e ortografia. Avalie se as informações contribuem pra o gerenciamento de requisitos, mais especificamente para a rastreabilidade de requisitos.
- c) Durante a leitura, identifique pontos do conteúdo das considerações para as quais você deseja registrar um comentário.
- d) Utilize a Tabela constante no final deste documento (Seção 6) para registrar seus comentários:
  - A coluna **ID** representa um campo auto incremental de considerações provenientes das Revisões;
  - A coluna **Categoria** representa o tipo de consideração da Revisão. Estes tipos serão melhor explicados na Seção 6 deste documento;

- A coluna **Item** representa o nome da técnica constante no catálogo apresentado e que possui alguma consideração proveniente da Revisão;
  - A coluna **Comentário com a Justificativa** representa a consideração do Revisor quanto à criação do catálogo;
  - A coluna **Sugestão** representa a proposta do Revisor para contornar o problema, caso aplicável.
- e) Após concluir a análise do documento em anexo preencha a avaliação objetiva da proposta (Seção 5).
- f) Ao concluir a revisão, por favor, envie seu documento de revisão para: [prcmalcher@gmail.com](mailto:prcmalcher@gmail.com).

#### 4. Dados de Identificação do Revisor

**Nome do Revisor:**

**Data da Revisão:**

#### 5. Perfil do Revisor do Framework

- a) Qual seu nível de conhecimento em Gerencia de Requisitos
- Alto  Médio
- Baixo  Nenhum
- b) Qual seu nível de conhecimento em Rastreabilidade de Requisitos
- Alto  Médio
- Baixo  Nenhum
- c) Já Trabalhou implantando Gerência de Requisitos em uma organização?
- Sim. Quantos: 16
- Não
- d) Qual o seu tempo de experiência em gerenciamento de requisitos?
- Mais de cinco anos  Entre dois e cinco anos
- Entre um e dois anos  Menos de um ano
- Nenhum
- e) Qual o seu tempo de experiência em Implantação de Modelos para Melhoria do Processo de Gerência de Requisitos?



c) Como você considera a definição dos tipos de técnicas?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_

d) Como você considera a definição das fase(s) do processo de software envolvida(s) em cada técnica?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_

e) Como você considera a definição da dimensão da rastreabilidade em cada técnica?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_

f) Como você considera a definição do principal contexto de aplicação de cada técnica (quando houve)?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_



g) Como você considera a definição do exemplo de implementação de cada técnica (quando houve)?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_

h) Como você considera a definição de ferramentas disponíveis que implementam cada técnica (quando houve)?

- Completa
- Incompleta
- Inconsistente
- Não sei

Observações: \_\_\_\_\_

i) Você considera que o catálogo pode ser um referencial para ser utilizado na Gerência de Requisitos, no que tange a Rastreabilidade de Requisitos?

- Sim
- Parcialmente
- Não

Observações: Desde que contemple os ajustes solicitados na seção 7.

## Revisão do Catálogo

**Observação:** A linha em amarelo na Tabela abaixo representa um exemplo de preenchimento das colunas descritas na Seção 3 deste documento e deve ser apagada ao preencher a tabela.

Segue abaixo os itens utilizados para a coluna "**Categoria**"

- **TA (Técnico Alto)**, indicando que foi encontrado um problema em um item que, se não for alterado, comprometerá as considerações;
- **TB (Técnico Baixo)**, indicando que foi encontrado um problema em um item que seria conveniente alterar;

- **E (Editorial)**, indicando que foi encontrado um erro de português ou que o texto pode ser melhorado;
- **Q (Questionamento)**, indicando que houve dúvidas quanto ao conteúdo das considerações;
- **G (Geral)**, indicando que o comentário é geral em relação às considerações.
- **BP (Boas Práticas)**, indicando que o comentário está relacionado à lista de boas práticas.

ID	Categoria	Item	Comentário com a Justificativa	Sugestão
1	TB	Geral	Faltou detalhar a melhor ferramenta X no catálogo	Detalhar a Ferramenta X