
Engenharia de Confiabilidade de Software: Um Mapeamento Sistemático

Joicymara S. Xavier



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uberlândia
2015

Joicymara S. Xavier

Engenharia de Confiabilidade de Software: Um Mapeamento Sistemático

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Autran Macêdo

Coorientador: Rivalino Matias

Uberlândia

2015

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

X3e
2015

Xavier, Joicymara S., 1988-
Engenharia de confiabilidade de software: um mapeamento
sistemático / Joicymara S. Xavier. - 2015.
135 f. : il.

Orientador: Autran Macêdo.
Coorientador: Rivalino Matias.
Dissertação (mestrado) - Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Ciência da Computação.
Inclui bibliografia.

1. Computação - Teses. 2. Software - Confiabilidade - Teses. I.
Macêdo, Autran. II. Matias, Rivalino. III. Universidade Federal de
Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
IV. Título.

CDU: 681.3

Àquela que abdicou da sua vida para que eu pudesse sonhar com a minha: minha mãe!

Agradecimentos

Agradeço imensamente a Deus, pela força que não me deixa desistir de seguir em frente. Por conduzir os meus caminhos e pelos incontáveis milagres realizados em minha vida.

Ao meu orientador, Prof. Autran Macêdo, pelos valiosos ensinamentos. Por conseguir me conduzir com paciência e maestria durante esses anos. A sua dedicação, postura, o seu apoio incondicional foram essenciais para minha formação e crescimento, tanto pessoal quanto profissional. A você, Autran, minha sincera e eterna gratidão. A meu coorientador, Prof. Rivalino Matias Júnior, pelo empenho dedicado à elaboração deste trabalho, pela firmeza que tanto me fez crescer, o meu muito obrigada.

À minha família, meu bem mais precioso. Ao meu pai (*in memoriam*), meu exemplo de serenidade e paciência, por todos os ensinamentos deixados e pela educação que é um dos bens mais valiosos que eu tenho. À minha mãe (*in memoriam*), minha heroína. Pelas lágrimas e suor derramados em prol do meu crescimento e minha educação. Pela pessoa que me ensinou a ser, pelo exemplo que para sempre irei seguir. Nada disso hoje seria possível sem o seu sacrifício, que foi o começo de tudo. Aos meus irmãos, Geysa e Danilton, por suportarem a distância e me apoiarem incondicionalmente nesse desafio. Por serem hoje, meu chão.

Aos meus amigos, que mesmo nas longas ausências nunca deixaram de me apoiar. Principalmente aqueles que, em Uberlândia, assumiram o lugar de família, Romêrson, Emanuely, Geycy, Rafael, Taffarel, Juliete, Luciane, Marcela, Bruna, Cíntia, Natália, Marianna, Luciana e Myllene.

Aos colegas de laboratório pela convivência e risadas, pela incessante ajuda e por todos os momentos de cooperação que me ajudaram a superar cada etapa deste trabalho. A todos os funcionários da UFU, em especial ao Erisvaldo, por toda ajuda e excelente trabalho. Por fim, meu agradecimento ao CNPQ pelo incentivo financeiro.

“A persistência é o caminho do êxito.”
(Charles Chaplin)

Resumo

Os sistemas computacionais estão presentes em praticamente todos os principais processos da sociedade moderna. Esta alta dependência faz com que a falha destes sistemas tenha impacto significativo na vida das pessoas. A principal causa de falhas em sistemas computacionais são os componentes de software. Essas falhas são objeto de estudo da Engenharia de Confiabilidade de Software ou *Software Reliability Engineering* (SRE). SRE é uma área multidisciplinar que integra, principalmente, probabilidade, estatística e computação. São variados os campos de pesquisa em SRE. Sendo assim, algumas perguntas podem ser formuladas a seu respeito, por exemplo: Quais campos de pesquisa em SRE têm sido mais investigados recentemente? Quais são os atuais grupos de pesquisa em SRE mais ativos em nível internacional? Quais problemas são abordados pelos trabalhos mais citados em SRE? Quais são os tópicos de pesquisa emergentes em SRE? Diante disso, este trabalho apresenta os resultados de um mapeamento sistemático sobre confiabilidade de software. Este mapeamento considerou os últimos 12 anos de publicações em 92 veículos (revistas e conferências internacionais), totalizando uma amostra de 300 artigos. Foram utilizadas técnicas estatísticas para agrupar artigos similares e identificar tópicos de pesquisa em confiabilidade de software. Os resultados mostram que trabalhos que adotam uma abordagem puramente teórica predominam sobre aqueles que adotam abordagem empírica. Contudo, nos últimos anos, percebe-se que a abordagem teórico-empírica tem se tornado mais acentuada. Além disso, observou-se que o fator “colaboração externa” apresenta-se como um dos mais evidentes nos centros de pesquisa influentes. Finalmente, entre os tópicos de pesquisa abordados neste trabalho, os resultados indicam uma demanda por pesquisas teórico-empíricas em testes de software para os próximos anos.

Palavras-chave: Confiabilidade de Software. Mapeamento Sistemático. Agrupamentos.

Abstract

Computer systems are present in practically all of the main processes of modern society. This high dependency means that failures in such systems have a direct impact upon people lives. Computer system failures are mainly caused by software defects. The software reliability engineering (SRE) is the field that studies how to make software more reliable. SRE is a multi-discipline field, which integrates in essence, probability, statistics, and computer science. The research areas in SRE are varied, so, diversity questions could be raised as to: Which fields of SRE have been most recently investigated? Which are the international SRE groups most active? Which problems are covered by the most cited SRE studies? Which are the emerging topics in SRE? Therefore, this study presents the results from a systematic mapping study on software reliability. This study considered the last 12 years of publications covering 92 publication venues (journals and international conferences), with a total sampling of 300 papers. Statistical techniques were used for clustering papers in terms of similarity, as well as for identifying the research topics of software reliability. The results showed that studies which adopt a purely theoretical approach were predominantly greater in number than empirical studies. However, we observe that over the last few years the theoretical-empirical approach has become more relevant. Besides this, it was also observed that the factor “external collaboration” presented itself as one of the most evident in those research centers of greater influence. Finally, among the research topics covered in this study, we identified a trend through the results, towards a demand in theoretical-empirical research in software testing for the coming years.

Keywords: Software Reliability. Systematic Mapping Study. Clustering.

Lista de ilustrações

Figura 1 – Árvore da Dependabilidade	28
Figura 2 – Cadeia fundamental da dependabilidade	29
Figura 3 – O processo de mapeamento sistemático	35
Figura 4 – Dendograma resultante do método Ward	40
Figura 5 – Protocolo Mapeamento Sistemático	48
Figura 6 – Dendograma	52
Figura 7 – Número de artigos por ano pesquisa	55
Figura 8 – Distribuição dos artigos da amostra pelos veículos de publicação.	56
Figura 9 – Dendograma dos 300 artigos selecionados com nomeação dos grupos	57
Figura 10 – Distribuição dos artigos por agrupamento	58
Figura 11 – Número de trabalhos teóricos, empíricos e ambos por ano	59
Figura 12 – Rede de colaboração entre os pesquisadores mais citados.	64
Figura 13 – Distribuição dos pesquisadores em SRE por instituição	64
Figura 14 – Publicações anuais por agrupamento	65
Figura 15 – Distribuição dos tópicos AF, AC e MS entre os países	68
Figura 16 – Distribuição dos agrupamentos entre os países	69
Figura 17 – Mapeamento dos tópicos de pesquisa, abordagens e período pesquisado	71
Figura 18 – Box-plot do número de publicações das instituições de cada país	73
Figura 19 – Box-plot do número de citações das instituições de cada país	73

Lista de tabelas

Tabela 1 – Elementos a serem agrupados	38
Tabela 2 – Questões de Pesquisa (SINGHAL; SINGHAL, 2011).	43
Tabela 3 – Exemplo de grupos de palavras-chave (KGs).	51
Tabela 4 – Exemplo de mapeamento de KGs para os artigos	52
Tabela 5 – Distribuição dos estudos por tipo de pesquisa	59
Tabela 6 – Distribuição de tipos de artigo por agrupamento	59
Tabela 7 – Afiliação dos dez autores mais produtivos	60
Tabela 8 – Afiliação dos dez autores mais citados	61
Tabela 9 – Os três trabalhos mais citados por agrupamento.	62
Tabela 10 – Referências dos artigos citados na Tabela 9.	63
Tabela 11 – Relação do número de cada KG para cada tópico	70
Tabela 12 – Afiliação dos autores mais produtivos e influentes	74
Tabela 13 – Produção Bibliográfica.	80

Lista de siglas

AC Avaliação de Confiabilidade

AF Análise de Falhas

DSN *International Conference on Dependable Systems and Networks*

EBSE *Evidence-Based Software Engineering*

FTA *Fault Tree Analysis*

ICSE04 *International Conference on Software Engineering 2004*

ISSRE *International Symposium on Software Reliability Engineering*

MC Modelagem de Confiabilidade

MCCS Modelagem de Crescimento de Confiabilidade de Software

ME Modelagem Estocástica

MS Manutenção de Software

NTP Número de Trabalhos Publicados

NC Número de Citações

RBD *Reliability Block Diagram*

SME *Systematic Mapping Study*

SRE *Software Reliability Engineering*

SQE Soma dos Quadrados dos Erros

TS Testes de Software

Sumário

1	INTRODUÇÃO	21
1.1	Motivação	22
1.2	Objetivos e Desafios da Pesquisa	23
1.3	Contribuições	24
1.4	Organização da Dissertação	24
2	FUNDAMENTAÇÃO TEÓRICA	27
2.1	Engenharia de Confiabilidade de Software	27
2.1.1	Conceitos Básicos	28
2.2	Mapeamento Sistemático	33
2.3	Análise de Agrupamento: Método Hierárquico de Ward	37
2.4	Ferramenta de Apoio ao Mapeamento Sistemático: Mendeley .	40
2.5	Trabalhos Relacionados	41
3	METODOLOGIA	45
3.1	Planejamento	45
3.1.1	Questões de Pesquisa	45
3.1.2	Protocolo do Mapeamento	48
3.2	Condução do Estudo	49
3.2.1	Seleção de Estudos Primários	50
3.2.2	Extração de Dados e Agrupamento dos Estudos	50
3.3	Disseminação	53
4	RESULTADOS	55
4.1	Q1 - Quais tópicos de pesquisa em SRE têm sido mais investi- gados?	56
4.2	Q2 - Quais tipos de pesquisa têm sido adotadas?	58
4.3	Q3 - Qual a origem dos autores mais produtivos em SRE? . . .	60

4.4	Q4 - Qual é a origem dos trabalhos mais citados em SRE? . . .	60
4.5	Q5 - Com quem os autores mais influentes cooperam?	62
4.6	Q6: Quais são os tópicos de pesquisa emergentes em SRE? Quais os tópicos de pesquisa saturados?	65
5	MAPEAMENTO E ANÁLISE DOS RESULTADOS	67
5.1	Tópicos de pesquisa em SRE	67
5.2	Tipos de pesquisa	70
5.3	Instituições dos autores influentes e produtivos	72
5.4	Cooperações entre instituições	74
6	CONCLUSÃO	77
6.1	Principais Contribuições	77
6.2	Ameaças à Validade	78
6.3	Trabalhos Futuros	78
6.4	Contribuições em Produção Bibliográfica	80
	REFERÊNCIAS	81

APÊNDICES 87

APÊNDICE B	–	LISTA DE <i>KEYWORD GROUPS</i>	123
APÊNDICE C	–	ACRÔNIMOS DAS INSTITUIÇÕES	125
APÊNDICE D	–	VEÍCULOS DE PUBLICAÇÃO	129
D.1		Lista de Conferências	129
D.2		Lista de Periódicos	131

Introdução

Desde o surgimento do primeiro computador eletrônico digital, há mais de 70 anos (NULL; LOBUR, 2011), a dependência das pessoas por computadores em suas atividades diárias tem crescido. Atualmente, sistemas computacionais permeiam a sociedade moderna. Sistemas computacionais estão embarcados em relógios de pulso, utensílios domésticos, telefones, aeronaves, edifícios e sistemas hospitalares, afetando milhões de pessoas (LYU, 1996).

Um exemplo de sistema hospitalar é o monitoramento de pacientes cardíacos. Em hospitais bem equipados, sensores que detectam sinais elétricos associados com a atividade cardíaca são ligados à região do coração do paciente. Os sinais desses sensores são transmitidos para um computador, que por sua vez é programado para analisar os dados recebidos. Se a entrada de dados indicar que o paciente está bem, o sistema não gera saída alguma. Se os dados indicarem o começo de complicações, o sistema emite um alarme para a enfermagem sinalizando que o paciente precisa de cuidados humanos e o tipo de intervenção mais apropriada (PHAM, 2007). Uma falha nesse sistema pode facilmente levar um paciente a óbito, como foi o caso de um outro sistema hospitalar, o Therac-25.

O Therac-25, uma máquina de radioterapia controlada por computador, muito moderna para sua época, foi a responsável pela morte de 3 pacientes. Entre junho de 1985 e janeiro de 1987, aconteceram 6 acidentes em que os pacientes foram expostos a elevados níveis de radiação. Desses pacientes, 3 faleceram e os demais ficaram gravemente feridos. A causa do problema com o equipamento foi em função de uma falha no software de controle do mesmo (LEVESON; TURNER, 1993). Em um outro exemplo, o foguete não tripulável Ariane 5 explodiu após seu voo inaugural, em 4 de junho de 1996. O foguete, de \$500 milhões, explodiu 40 segundos após ter dado início ao voo, a uma altitude de cerca de 3700 metros. A causa do problema foi uma falha no software de controle inercial do foguete (LIONS et al., 1996).

Muitos sistemas complexos enfrentam a mesma situação: a confiabilidade do software sendo o gargalo da confiabilidade do sistema (LYU, 1996). Diversos exemplos em que falhas em sistemas computacionais implicam em perda de vidas ou de grandes quantias

de dinheiro podem ser encontrados. Esses exemplos mostram que a principal causa de falhas em sistemas computacionais são os componentes de software (SULLIVAN; CHILLAREGE, 1991; LI et al., 2006). Essas falhas são objeto de estudo da Engenharia de Confiabilidade de Software do inglês *Software Reliability Engineering* (SRE). Este trabalho apresenta um mapeamento sistemático sobre SRE.

1.1 Motivação

A SRE é uma área vasta e provê variados campos de pesquisa, tais como, modelagem de confiabilidade, testes de software, modelagem de crescimento de confiabilidade, análise de tempos de falha, modelos estocásticos e markovianos, árvore de falhas, dentre outros. Diante de tal diversidade, algumas perguntas podem ser formuladas, por exemplo: Quais campos de pesquisa em SRE têm sido mais investigados recentemente? Quais são os atuais grupos de pesquisa em SRE mais ativos em nível internacional? Quais problemas são abordados pelos trabalhos mais citados em SRE? Quais são os tópicos de pesquisa emergentes em SRE? Encontrar respostas para essas e outras questões correlatas é essencial para pesquisadores que desejam direcionar melhor seus esforços nessa área de pesquisa.

Em uma busca rápida sobre estudos em confiabilidade de software percebe-se que um grande número de publicações recentes são encontradas sobre os variados tópicos de pesquisa que compõem essa área. Essas publicações estão distribuídas em diversos veículos, indexadas por inúmeras bases de dados. Dessa forma, é importante resumir e fornecer uma visão geral das evidências encontradas nessas publicações.

Muitos campos de pesquisa têm metodologias específicas para resumir os estudos de suas áreas. No *International Conference on Software Engineering* 2004 (ICSE04) (KITCHENHAM; DYBA; JORGENSEN, 2004) sugeriram que engenheiros de software adotassem a *Evidence-Based Software Engineering* (EBSE) para sistematizar seus estudos. Desde então, muitas áreas dentro da Engenharia de Software têm se preocupado em realizar esse tipo de estudo e possivelmente ajudado muitos pesquisadores e estudantes a direcionarem suas pesquisas. Uma busca pelo número de citações obtidas por trabalhos desse paradigma, analisados em (KITCHENHAM et al., 2009), comprova que trabalhos voltados para a EBSE têm sido bastante procurados. Apesar da proposta do paradigma ser relativamente recente, o considerável número de citações desses trabalhos mostra que eles têm sido úteis e valorizados pelos engenheiros de software.

Dessa forma, um estudo que mapeie conceitos e apresente uma visão geral de uma área tão crítica e importante quanto SRE é essencial. Essa visão geral, apoiada pela EBSE, pode ser obtida através das revisões (KITCHENHAM, 2004) e mapeamentos sistemáticos (PETERSEN et al., 2008). As revisões sistemáticas analisam a fundo as evidências encontradas, enquanto o mapeamento as quantifica. Como SRE é uma área muito ampla, um mapeamento vem a ser mais indicado do que uma revisão sistemática (KITCHENHAM

et al., 2007). O mapeamento, neste caso, pode inclusive, apontar nichos de pesquisa para futuras revisões.

Até onde foi esta pesquisa encontrou-se apenas um estudo de revisão sistemática relacionado à SRE (SINGHAL; SINGHAL, 2011). Identificou-se nessa revisão alguns pontos que poderiam ser fortalecidos (esses pontos serão discutidos na Seção 2.5). Além disso, percebeu-se que questões mais abrangentes poderiam ser realizadas. Sendo assim, foram levantadas questões de pesquisa que pudessem traduzir os interesses de todos os envolvidos na área.

1.2 Objetivos e Desafios da Pesquisa

Este trabalho fornece uma forma estruturada de pesquisa e catalogação do material disponível sobre Confiabilidade de Software através de um mapeamento sistemático. Essa catalogação visa categorizar os estudos para que seja possível fornecer uma visão geral de SRE. Essa visão geral deverá ser obtida a partir das respostas aos seguintes questionamentos:

- ☐ Quais tópicos de pesquisa em SRE têm sido mais investigados?
- ☐ Quais tipos de pesquisa têm sido adotadas?
- ☐ Qual é a origem dos autores mais produtivos em SRE?
- ☐ Qual é a origem dos trabalhos mais citados em SRE?
- ☐ Com quem os autores mais influentes cooperam?
- ☐ Quais são os tópicos de pesquisa emergentes em SRE? Quais os tópicos de pesquisa saturados?

Essas perguntas foram definidas com a finalidade de direcionar a pesquisa. De modo mais específico, para alcançar o objetivo principal, este trabalho propõe-se a:

- ☐ Recuperar estudos relevantes que possam mapear a SRE e responder todas as questões de pesquisa propostas.
- ☐ Investigar tópicos de pesquisa que tratem de temas distintos dentro de SRE.
- ☐ Encontrar possíveis subáreas que posteriormente podem ser explorados através de Revisões Sistemáticas de Literatura.
- ☐ Desenvolver um esquema de classificação dos estudos que possa ser usado para realizar mapeamentos sistemáticos nessa e em outras áreas.

Alguns desafios devem ser vencidos para que os objetivos pretendidos sejam alcançados com êxito, tais como, identificar estudos que sejam relevantes. Para que o mapeamento tenha qualidade e reflita a realidade, deve-se criar critérios para que esses estudos sejam recuperados. A definição desses critérios irá evitar que estudos importantes sejam perdidos e assegurar que estudos que não são foco da pesquisa não sejam considerados. Nesse sentido, a *string* de busca é importante, porém, por mais cuidado que se tenha ao defini-la, os resultados obtidos ainda são passíveis de erros. Ao buscar por “*software reliability*” por exemplo, estudos que contém essa expressão em alguma parte do texto serão retornados. Porém, esses estudos podem não estarem relacionados a confiabilidade de software. Por isso, a análise individual de cada estudo recuperado se faz necessária.

1.3 Contribuições

A contribuição desta pesquisa se dá por meio de duas frentes: a utilização dos resultados por parte dos mais diversos pesquisadores da área e, a sugestão de um método para classificação de estudos que pode ser utilizado em qualquer área de pesquisa.

Os interessados neste mapeamento podem ser divididos em três categorias: aqueles que estão em busca de uma área para trabalhar, os que estão iniciando seus estudos em SRE e os pesquisadores experientes na área. Para aqueles que estão a procura de uma área para trabalhar, em geral estudantes ou pesquisadores que desejam mudar de área, através desta pesquisa, poderão ter contato com alguns conceitos sobre SRE. Além disso, fornecerá uma visão geral da área, lugares onde buscar possíveis orientadores ou colaboradores para suas pesquisas. Para pessoas que estão iniciando suas pesquisas na área, alunos de mestrado e doutorado por exemplo, e precisam preparar uma visão geral do tópico que irão trabalhar, este mapeamento pode ser útil. Além de prover essa visão geral, este estudo fornece e categoriza os artigos que podem ser utilizados como referencial teórico. Quando esse estudante for publicar os seus resultados, este estudo pode auxiliá-lo ainda a encontrar um veículo adequado. Por fim, para os pesquisadores experientes, este estudo pode ajudá-los a identificar lacunas na pesquisa em SRE nos últimos anos e/ou tópicos que podem ser promissores no futuro.

A metodologia utilizada para realizar este mapeamento seguiu as sugestões propostas por (PETERSEN et al., 2008), porém, acrescentou-se um método estatístico, aliado às palavras-chaves dos estudos, para classificá-los com relação aos temas dentro da área. Esse método produziu resultados satisfatórios para esta pesquisa, podendo ser utilizado em mapeamentos em outras áreas.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte forma.

O Capítulo 2 apresenta a revisão da literatura correlata, as principais características da Engenharia de Confiabilidade de Software, os passos para se realizar um mapeamento sistemático e demais conceitos relacionados a este trabalho.

O Capítulo 3 apresenta os métodos utilizados para realização da pesquisa, incluindo o detalhamento das questões de pesquisa propostas, o protocolo para seleção dos estudos, os passos para condução da pesquisa e por fim, mostra como ela será divulgada.

No Capítulo 4 são apresentados os resultados obtidos para cada uma das questões de pesquisa. Nesse capítulo são mostrados os tópicos de pesquisa que foram identificados através do método estatístico. Esses tópicos são detalhados e explorados em cada uma das questões de pesquisa.

No Capítulo 5 é feito o mapeamento dos resultados obtidos. Nesse capítulo, os resultados para cada questão são combinados e analisados. As análises presente nesse capítulo sugerem algumas direções que a pesquisa em SRE pode vir a seguir nos próximos anos.

Por fim, o Capítulo 6 apresenta as considerações finais sobre a pesquisa, destacando suas principais contribuições, as ameaças à validade do estudo, os trabalhos futuros e as contribuições em produção bibliográfica.

Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos necessários à realização deste trabalho, a saber: Engenharia de Confiabilidade de Software (Seção 2.1), Mapeamento Sistemático (Seção 2.2), Análise de Agrupamento: Método Hierárquico de Ward (Seção 2.3), Ferramenta de Apoio ao Mapeamento Sistemático: Mendeley (Seção 2.4) e por fim, os Trabalhos Relacionados (Seção 2.5).

2.1 Engenharia de Confiabilidade de Software

A Engenharia de Confiabilidade é uma subdisciplina da Engenharia de Sistemas e tem ênfase na confiança do funcionamento de sistemas. A Engenharia de Confiabilidade, inicialmente voltada para hardware, teve suas primeiras aplicações durante a segunda guerra mundial e atualmente é aplicada a diversos setores da indústria (KAPUR et al., 2013). Cada setor da indústria possui seus padrões e teorias associadas para descrever as características das falhas de seus produtos. Por exemplo, sistemas físicos podem falhar por danos ambientais (ex. temperatura) ou imperfeições de seus materiais (ex. rachaduras), entre outros. Os princípios para a SRE são os mesmos dos aplicados a hardware, contudo, os mecanismos que causam falhas em software são diferentes dos presentes em falhas de hardware (LYU, 1996). As causas das falhas de software estão no projeto/codificação do software, por isso são mais difíceis de identificar, classificar e corrigir.

Confiabilidade é definida como “a capacidade de um sistema ou componente desempenhar as suas funções requeridas, sob condições estabelecidas, por um período de tempo especificado” (IEEE/ANSI, 1990). Ela é geralmente considerada como fator-chave para a qualidade, uma vez que se dispõe a quantificar falhas de software. Como resultado, confiabilidade é um ingrediente essencial para a satisfação do cliente (LYU, 1996).

Os objetivos da SRE são (O’CONNOR; KLEYNER, 2011):

- Aplicar conhecimentos de engenharia e técnicas especialistas para prevenir ou reduzir a probabilidade e frequência de falhas.

- ❑ Identificar e corrigir as causas de falhas que ocorrem, apesar dos esforços para impedi-las.
- ❑ Determinar formas de lidar com as falhas que ocorrem, se suas causas não tiverem sido corrigidas.
- ❑ Aplicar métodos para estimar a provável confiabilidade de novos projetos e para analisar dados de confiabilidade.

Antes de especificar modelos ou técnicas para se alcançar confiabilidade é necessário que se entenda alguns conceitos básicos. Esses conceitos serão apresentados na subseção a seguir.

2.1.1 Conceitos Básicos

Dependabilidade - Falta, erro e falha

O conceito de dependabilidade surgiu da necessidade de maior confiança nos sistemas e é regulamentada pela ISO 9000-4 (ISO, 1994). Essa área se preocupa com a confiança no funcionamento dos sistemas, tanto de hardware quanto de software. Os conceitos que envolvem a dependabilidade consistem de três partes: os atributos, as ameaças e os meios pelo qual a dependabilidade é atingida. A confiabilidade é um dos atributos da dependabilidade, como pode ser visto na Figura 1.

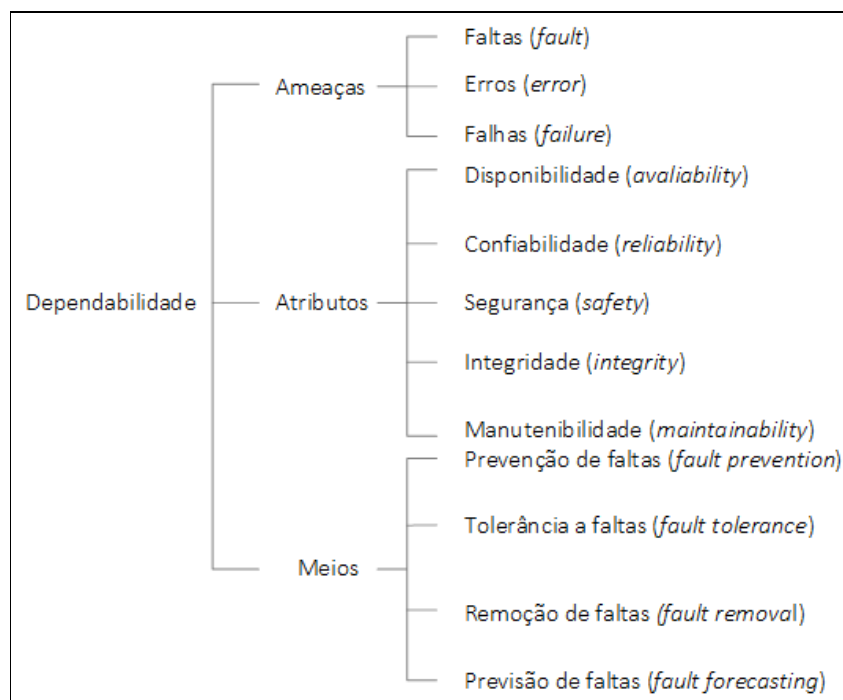


Figura 1 – Árvore de Dependabilidade (AVIZIENIS et al., 2001).

Os fatores que ameaçam a confiabilidade de um sistema são classificados por (AVIZIENIS et al., 2001) como sendo de três tipos: falta, erro e falha. Uma falta, também conhecida como defeito/*bug*, é considerada dormente até a sua ativação. Uma falta pode ser a declaração incorreta de uma variável, por exemplo. Se uma variável do tipo *int* for erroneamente declarada como *char*, quando essa variável precisar ser acessada, pode ser que um erro ocorra. Uma falta é a causa ou hipótese de um erro. Um erro é a parte do estado interno do sistema que pode levar à ocorrência de uma falha. A falha ocorre quando um erro atinge a interface de serviço, entregando um resultado diferente do esperado. O erro pode se propagar pelos diversos componentes de um sistema antes de causar uma falha. A relação entre falta, erro e falha é expressa pela Figura 2.

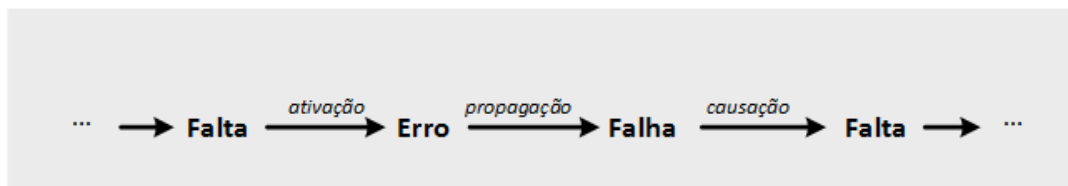


Figura 2 – Cadeia fundamental da dependabilidade (AVIZIENIS et al., 2001).

As setas nessa cadeia expressam uma relação de causalidade entre os elementos. Genericamente pode-se interpretar que, por propagação, vários erros podem ser gerados antes de ocorrer uma falha. O termo propagação do erro é usado para a transformação de um erro em outro erro, provocando ou não uma falha. A falha só será considerada como tal quando o erro for percebido pelo usuário do sistema como um desvio da função especificada.

Um sistema nem sempre falha da mesma maneira. As formas pelas quais um sistema pode falhar são seus modos de falha. Esses modos podem ser classificados de acordo a severidade da falha. Pode-se caracterizar serviço incorreto de acordo com quatro pontos de vista (AVIZIENIS et al., 2001):

- ❑ o domínio da falha;
- ❑ a controlabilidade das falhas;
- ❑ a consistência das falhas, quando um sistema possui dois ou mais usuários;
- ❑ as consequências de falhas no ambiente.

Técnicas, métodos e ferramentas para aumentar a confiabilidade do sistema são propostas em diversos estudos. Para alcançar a confiabilidade, ou qualquer um dos outros atributos dentro da dependabilidade, um conjunto de quatro técnicas básicas pode ser utilizado: prevenção de faltas, tolerância a faltas, remoção de faltas e previsão de faltas.

Prevenção de faltas tem como objetivo evitar que o sistema contenha faltas. Isso pode ser alcançado através de técnicas de controle de qualidade utilizado durante o projeto e desenvolvimento do software. Essas técnicas incluem o refinamento iterativo da exigência do usuário, a engenharia do processo de especificação de software, o uso de bons métodos de projeto de software e o incentivo à escrita de código limpo são, em geral, abordagens para prevenir faltas em software (LYU, 1996).

Outras abordagens que têm sido utilizadas para prevenir faltas em software incluem métodos formais e reuso de software (LYU, 1996). Os métodos formais permitem que as especificações de requisitos sejam desenvolvidas e mantidas através de linguagens matemáticas, como por exemplo as Redes de Petri. A reutilização de software tem como objetivo diminuir o tempo e custo de desenvolvimento aumentando assim, os índices de produtividade. Um dos principais benefícios da reutilização é a melhoria da qualidade do sistema, uma vez que os componentes reusados já foram testados. Componentes testados tendem a ser mais confiáveis e consistentes do que um componente recém desenvolvido.

Tolerância a faltas se destina a preservar a prestação de serviço correto na presença de faltas ativas. A tolerância geralmente é implementada através da detecção de erros e a subsequente recuperação do sistema (AVIZIENIS et al., 2001). A tolerância a faltas de software está preocupada em utilizar de todas as técnicas necessárias para permitir que o sistema continue funcionando caso alguma falta dormente venha a se manifestar.

As técnicas de tolerância a faltas incluem técnicas de monitoramento, atomicidade de ações, verificação de decisões e tratamento de exceções. As abordagens de tolerância a faltas em software incluem: a técnica de blocos de recuperação (*recovery blocks technique*) (RANDELL, 1975), a técnica de programação *N-version* (*N-version programming technique*) (CHEN; AVIZIENIS, 1977) e a técnica de programação *N self-checking* (YAU; CHEUNG, 1975). Essas abordagens têm encontrado uma ampla gama de aplicações na indústria em geral (LYU, 1996).

Remoção de faltas é realizada durante todo o ciclo de vida de um sistema, ou seja, tanto na fase de desenvolvimento quanto na operacional. A remoção de faltas consiste em três etapas: verificação, diagnóstico e correção. Verificar envolve checar se o sistema adere a dadas propriedades, denominadas de condições de verificação. Se ele não aderir, são executadas as outras duas etapas. Após a correção, o processo de verificação deve ser repetido a fim de checar se a remoção da falta não gerou consequências indesejadas (AVIZIENIS et al., 2001).

As técnicas de verificação podem ser classificadas de acordo a execução ou não do sistema (AVIZIENIS et al., 2001):

- Verificação estática: a verificação é feita sem a execução real do sistema, por meio de análise estática, modelos de verificação ou provas de teoremas.
- Verificação dinâmica: a verificação é feita em tempo de execução. As entradas

fornecidas ao sistema podem ser simbólicas ou reais.

Técnicas usualmente utilizadas são a verificação estática formal e injeção de faltas, para os testes que necessitam de faltas ou erros como parte dos padrões de teste.

Previsão de faltas é realizada efetuando-se uma avaliação do comportamento do sistema com relação à ocorrência ou ativação da falta. Essa avaliação tem dois aspectos: qualitativo e quantitativo. O aspecto qualitativo tem como objetivo identificar, classificar e ranquear os modos ou combinações de eventos (falhas de componentes ou condições ambientais) que levariam a falhas do sistema. Já o aspecto quantitativo tem como objetivo avaliar, em termos de probabilidades a medida em que alguns dos atributos de confiabilidade estão satisfeitos. Esses atributos são vistos como medidas de confiança (AVIZIENIS et al., 2001).

As técnicas para previsão de faltas envolvem métodos diferentes para cada um dos aspectos de avaliação. Os métodos utilizados para avaliação qualitativa podem ser, por exemplo, análise de efeito e modos de falha. Cadeias de Markov e Redes de Petri Estocásticas podem ser utilizadas na análise quantitativa. As Cadeias de Markov consistem de uma lista de possíveis estados do sistema, os possíveis caminhos entre eles e as taxas, de falha ou reparo, entre esses estados. As Redes de Petri Estocásticas são uma variação das Redes de Petri. São representadas como um grafo bipartido e levam o tempo em consideração, o que as torna adequadas para modelar confiabilidade (KHANDELWAL, 2013).

Outros métodos podem ser utilizados tanto na análise qualitativa quanto na quantitativa, como por exemplo, árvore de faltas (*Fault Tree Analysis* (FTA)) e diagramas de blocos de confiabilidade (*Reliability Block Diagram* (RBD)). FTA utiliza lógica booleana para combinar uma série de eventos que podem causar falhas. É utilizada para entender as causas de falhas em sistemas e, conseqüentemente, identificar as melhores maneiras de reduzir riscos. RBD utiliza blocos para representar os componentes e mostrar como cada um deles contribui para o sucesso ou fracasso do sistema. O sistema é representado por um conjunto de blocos com configuração em série ou em paralelo, de acordo com o comportamento que os componentes apresentam em relação aos outros. Se um componente em série falhar, todo o sistema falha; já se os componentes do sistema estiverem em paralelo, será necessário todos os componentes falharem para se ter uma falha de todo o sistema.

Perfil Operacional

O processo de falha e conseqüentemente, a confiabilidade do software é diretamente dependente do ambiente ou do perfil operacional específico do sistema (MUSA; IANNINO; OKUMOTO, 1987). O perfil operacional de um sistema é definido como o conjunto de operações que o software pode executar juntamente com suas probabilidades de ocorrência

durante seu uso em produção (LYU, 1996). Todos os modelos de confiabilidade de software assumem que os testes são executados com base nos perfis operacionais. As informações e a frequência de uso de diferentes operações/funções do software são quantificadas pelos perfis operacionais. Por exemplo, se um evento A ocorre 80% das vezes e B 20%, o perfil de A é 0,8 e B, 0,2.

Os perfis operacionais podem ser utilizados para selecionar os casos de teste e orientar os esforços de desenvolvimento, teste e manutenção sobre os componentes de maior uso ou risco. Além de diferenciar o ambiente, tipo de usuário e modo de operação de cada sistema ou componente.

Medição de Confiabilidade

Técnicas de medição de confiabilidade são utilizadas para estimar ou prever a confiabilidade de um sistema. As técnicas de estimação determinam a confiabilidade atual do software através da aplicação de técnicas de inferência estatística em dados de falha obtidos durante os testes do sistema ou durante sua operação (LYU, 1996). A estimativa considera a confiabilidade de um ponto no passado até o ponto atual. Modelos baseados em estimativa têm como propósito principal determinar uma retrospectiva da confiabilidade do sistema.

As atividades de previsão determinam a confiabilidade do software no futuro baseado nas métricas e medições disponíveis. Dependendo do estágio de desenvolvimento, dois tipos de técnicas são envolvidas: previsão de confiabilidade (*reliability prediction*) e previsão antecipada (*early prediction*). A previsão de confiabilidade pode ser estimada quando os dados de falhas do sistema estão disponíveis. Por exemplo, quando o software está em teste ou em operação. As técnicas de previsão podem ser utilizadas para parametrizar e verificar modelos de confiabilidade de software. Essas técnicas poderão realizar previsões de confiabilidade futuras.

A previsão antecipada pode ser realizada quando os dados de falhas não estão disponíveis. O software pode estar, por exemplo, no estágio de projeto ou de codificação. Dessa forma, a confiabilidade pode ser estimada através das métricas obtidas do processo de desenvolvimento. As características do produto resultante também podem ser utilizadas para determinar confiabilidade do software após os testes ou entrega.

Modelos de Confiabilidade de Software

Um modelo de confiabilidade especifica a forma geral de dependência do processo de falha nos principais fatores que a afetam: introdução de faltas, remoção de faltas e o ambiente operacional (LYU, 1996).

A taxa de falha de um sistema de software geralmente diminui conforme vai se descobrindo e removendo as faltas. Nesse sentido, determinar quanto tempo de teste será

necessário para atingir o objetivo de confiabilidade é importante. A modelagem de confiabilidade prevê a curva de taxa de falhas através de evidências estatísticas. Além de prever o tempo necessário para testar o software, a modelagem tem também como objetivo prever a confiabilidade esperada depois de terminados os testes.

A confiabilidade de software é um processo estocástico que pode ser descrito através de distribuições de probabilidade (LYU, 1996). Em geral, o software apresenta um crescimento da confiabilidade durante os testes e operação. Esse crescimento ocorre uma vez que faltas podem ser detectadas e removidas quando um software falha. Por outro lado, o software pode vir a ter uma diminuição da confiabilidade devido a mudanças no seu uso operacional ou devido a modificações incorretas. Sendo assim, diversos modelos são encontrados na literatura com o objetivo de abordar cada uma dessas características.

Modelagem de Crescimento de Confiabilidade

Um modelo de crescimento de confiabilidade demonstra as mudanças de confiabilidade do sistema ao longo do tempo. Como as falhas do sistema vão sendo descobertas e reparadas, a confiabilidade tende a melhorar durante os testes e depuração. Para prever a confiabilidade, o modelo de crescimento de confiabilidade conceitual deve então, ser traduzido em um modelo matemático (SOMMERVILLE, 2011).

Modelagem de crescimento de confiabilidade envolve a comparação de medidas de confiabilidade em determinados pontos ao longo do tempo, com funções conhecidas que mostram possíveis mudanças na confiabilidade. Por exemplo, uma função sugere que a confiabilidade de um sistema cresce linearmente a cada lançamento. Pela correspondência observada do crescimento da confiabilidade com uma destas funções é possível prever a confiabilidade de um sistema em algum ponto no futuro. Modelos de crescimento de confiabilidade podem, portanto, serem utilizados para apoiar o planejamento do projeto.

Modelos de crescimento de confiabilidade são amplamente empregados para avaliar a confiabilidade do software antes de sua liberação para uso operacional. Esses modelos geralmente são funções obtidas através do ajuste estatístico de curvas paramétricas. Esses modelos geralmente são complexos e refletem o regime de testes que frequentemente foi utilizado durante o processo de desenvolvimento (DHARMASENA; ZEEPHONGSEKUL; JAYASINGHE, 2011).

2.2 Mapeamento Sistemático

Estudos de Mapeamento Sistemático (*Systematic Mapping Study* (SME)) são projetados para prover uma ampla visão geral de uma área de pesquisa. Têm o objetivo de estabelecer se existe evidência de pesquisa em um tópico e prover um indicativo da quantidade da evidência (KITCHENHAM et al., 2007). SMEs devem ser realizados de acordo

uma estratégia de busca pré-definida e rigorosa de forma a permitir que a integridade da pesquisa seja avaliada. SMEs pertencem ao paradigma de pesquisa baseado em evidências, assim como as Revisões Sistemáticas de Literatura (KITCHENHAM, 2004). Essas práticas começaram a ser exploradas na medicina. Pesquisas indicaram que a recomendação médica baseada na opinião especializada não era tão confiável quanto recomendações baseadas na acumulação de resultados de experimentos científicos. Desde então, muitas áreas têm adotado essas abordagens. Na Engenharia de Software, a aplicação desse paradigma surgiu em 2004, criando o conceito de Engenharia de Software Baseada em Evidências (EBSE) (KITCHENHAM, 2004).

A ESBE busca fornecer os meios para que as melhores evidências da pesquisa sejam integradas com a experiência prática na tomada de decisão nas fases de desenvolvimento e manutenção de software (DYBA; KITCHENHAM; JORGENSEN, 2005). É comum começar uma pesquisa bibliográfica por artigos do tipo *survey*. Os *surveys* também fornecem revisões de literatura, porém são diferentes do mapeamento sistemático pois não são norteados por questões de pesquisa. Os *surveys* são apropriados para descrever e discutir o desenvolvimento ou “estado da arte” de um determinado assunto (ROTHER, 2007). Por outro lado, o mapeamento é projetado para fornecer uma visão mais ampla, de modo a prover uma indicação da quantidade de evidência sobre um determinado tópico de pesquisa (KITCHENHAM et al., 2007).

A fim de fornecer orientações para a realização de SMEs em Engenharia de Software, (KITCHENHAM et al., 2007) e (PETERSEN et al., 2008) resumizam os passos e cuidados necessários para que seja possível atingir bons resultados com mapeamentos sistemáticos. Esses passos são baseados nas orientações já consolidadas na medicina. Cada uma das etapas do mapeamento é formada por uma sequência de passos. Cada passo do processo gera um resultado. O resultado final do processo termina com o mapeamento sistemático, que, posteriormente deve ser disseminado. Os passos para realização da pesquisa seguiram o processo proposto por (PETERSEN et al., 2008) e, por questão de organização, foram divididos em três etapas: Planejamento, Condução e Disseminação. Embora o processo não seja estritamente sequencial na prática, adotou-se o protocolo proposto por (PETERSEN et al., 2008). Os passos são mostrados pela Figura 3, sendo que, os processos superiores descrevem os passos para a realização do trabalho e os inferiores indicam os resultados produzidos por esses passos. Cada um desses passos são detalhados a seguir. O passo “Definição das questões de pesquisa” refere-se à etapa de Planejamento e os passos seguintes à Condução. A etapa Disseminação foi adicionada seguindo recomendações de (KITCHENHAM et al., 2007).

Planejamento

O passo que compõe a etapa de Planejamento do SME é a “**Definição das Questões de Pesquisa**”. É a partir das questões de pesquisa que o mapeamento é elaborado

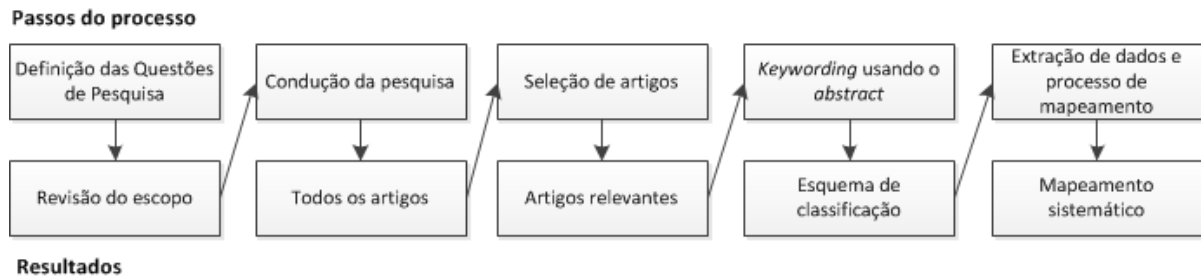


Figura 3 – O processo de mapeamento sistemático (PETERSEN et al., 2008).

e conduzido. As questões de pesquisa irão definir o **escopo do mapeamento**. (KITCHENHAM, 2004) sugere que as questões sejam baseadas nos critérios PICOC (População, Intervenção, Comparação, *Outcomes* e Contexto) (BEELMANN; PETTICREW; ROBERTS, 2006; KITCHENHAM et al., 2007):

A **População** refere-se ao universo que será pesquisado. Em Engenharia de Software pode ser, um papel (ex. programador, testador, gerente, entre outros), uma área de aplicação (ex. sistemas de tecnologia da informação, modelagem, etc), entre outros.

As **Intervenções** são as metodologias, tecnologias, ferramentas, processos de software que abordam questões específicas, por exemplo, tarefas de especificação de requisitos, testes de sistema, etc.

A **Comparação** é levada em consideração quando deseja-se confrontar uma abordagem com a metodologia/ferramenta/tecnologia de que trata a intervenção.

Os **Outcomes** estão relacionados com os resultados que se deseja obter com a pesquisa.

O **Contexto** define em qual âmbito/ambiente a pesquisa será realizada.

Com as questões de pesquisa definidas, é necessário desenvolver um protocolo para conduzir o mapeamento. A definição de um protocolo é importante para direcionar e sistematizar a pesquisa, deixar claro as estratégias que serão utilizadas para selecionar os estudos e, principalmente, eliminar o viés do pesquisador (KITCHENHAM, 2004). O protocolo deve incluir todos os elementos presentes no mapeamento e deve ser cuidadosamente avaliado por especialistas da área.

Condução

O objetivo de um trabalho de mapeamento é encontrar o maior número de estudos primários relativos à questão de pesquisa quanto possível, utilizando uma estratégia de busca imparcial. Estudos primários são os estudos individuais realizados com o intuito

de apresentar um corpo teórico ou realizar uma pesquisa empírica. Pesquisas que analisam estudos primários são chamadas de estudos secundários, tal como, este mapeamento sistemático. O rigor do processo de busca e a presença das questões de pesquisa são fatores que distinguem os mapeamentos de revisões tradicionais (KITCHENHAM et al., 2009). A etapa de **condução da pesquisa** irá retornar todos os artigos existentes na área pesquisada. Uma vez que o máximo de estudos primários tenham sido obtidos, eles precisam ser avaliados quanto à sua real relevância para o estudo.

A **triagem dos estudos** é necessária para identificar, dentre todos os estudos recuperados em buscas automáticas nas bases de dados, quais aqueles que atendem aos critérios exigidos para o SME. Essa triagem é feita obedecendo aos critérios de seleção pré-definidos, chamados de inclusão e exclusão. Os critérios de seleção destinam-se a identificar os estudos primários que fornecem evidências diretas sobre as questões de pesquisa (KITCHENHAM et al., 2007). Os critérios de seleção devem ser declarados durante o planejamento, embora possam ser refinados ao longo do processo de busca.

A etapa **keywording usando o abstract** tem o objetivo de facilitar o esquema de classificação. O *keywording* é feito em duas etapas. Primeiramente, os revisores leem o *abstract* e procuram por palavras-chave e conceitos que refletem a contribuição do estudo. Fazendo isso, o revisor também identifica o contexto da pesquisa. Na segunda etapa, o conjunto de palavras-chave de diferentes artigos são combinados para desenvolver uma compreensão de alto nível sobre a natureza e contribuição da pesquisa. Quando é escolhido um conjunto final de palavras-chave, estas podem ser agrupadas e utilizadas para formar as categorias para o mapeamento (PETERSEN et al., 2008).

A fase de **extração de dados e processo de mapeamento** tem o objetivo de projetar métodos de extração de dados para registrar, com precisão, as informações obtidas dos estudos primários. Visando reduzir a possibilidade de viés, os métodos de extração de dados devem ser definidos e testados. O **Mapeamento Sistemático** envolve agrupar e sumarizar os dados extraídos dos estudos primários. Para sintetizar os dados no mapeamento podem ser utilizadas representações gráficas da distribuição dos estudos pelos tipos de classificação (KITCHENHAM et al., 2007).

Disseminação

A disseminação envolve a escrita e propagação dos resultados para que as partes interessadas possam usufruir deles. A divulgação dos resultados de um mapeamento é muito importante, por isso, é recomendado que a estratégia de disseminação seja planejada durante a preparação do protocolo (KITCHENHAM et al., 2007).

No meio acadêmico, geralmente assume-se que a disseminação está relacionada à publicação de artigos em revistas e/ou conferências acadêmicas. Os relatórios devem ser revisados e avaliados, encerrando o processo de mapeamento sistemático. Mais deta-

lhes sobre os componentes do processo de mapeamento sistemático serão abordados no Capítulo 3.

2.3 Análise de Agrupamento: Método Hierárquico de Ward

Ao se buscar pelos estudos primários relativos a SRE, foi encontrado um grande volume de estudos (535 artigos). Mesmo depois da triagem dos estudos, ainda restaram muitos (300 artigos), tornando a categorização manual desses estudos inviável. Sendo assim, optou-se por agrupá-los utilizando um método para identificar e unir os artigos similares. Existem diversas técnicas estatísticas que foram desenvolvidas para este fim. Essas técnicas compõem a chamada Análise de Agrupamento, ou *Clustering*.

Na análise de agrupamento, tem-se a intenção de classificar objetos, itens ou indivíduos de acordo com suas semelhanças (EVERITT, 1992). Os itens similares são alocados em um mesmo grupo e, conseqüentemente, aqueles que são alocados em grupos diferentes são considerados dissimilares. Os métodos de agrupamento podem ser divididos em hierárquicos e não-hierárquicos. Os métodos hierárquicos classificam os objetos em grupos em diferentes etapas, produzindo uma árvore de classificação, os dendogramas (BUS-SAB, 1990). Os métodos não-hierárquicos agrupam elementos em k grupos, onde k é a quantidade de grupos definida previamente (FERREIRA, 2011).

Ao iniciar esta pesquisa não tinha-se conhecimento de qual o número de agrupamentos seria ideal para o conjunto de estudos obtidos. Dessa forma, os métodos não-hierárquicos seriam inviáveis pois eles precisam desse número. Sendo assim, optou-se pelos métodos hierárquicos. Diversos métodos hierárquicos são propostos na literatura, dentre eles, o método Ward é considerado um dos mais robustos (FERREIRA, 2011). O método Ward é considerado o mais robusto porque é o único dos métodos hierárquicos que, além de analisar a homogeneidade dos agrupamentos criados, leva em consideração a heterogeneidade desses agrupamentos em relação aos outros. Dessa forma, optou-se por utilizá-lo para agrupar os estudos dessa pesquisa.

O Método Ward (WARD JR, 1963) é um método hierárquico aglomerativo e iterativo que consiste basicamente de dois passos: primeiramente é calculado a Distância Euclidiana, d , entre dois objetos e em seguida, eles são agrupados de acordo com essa distância. Esses passos são repetidos até que reste somente um grupo. Os grupos são determinados de forma a obter-se homogeneidade dentro deles e heterogeneidade entre eles.

Os métodos da análise de agrupamentos requerem uma medida de proximidade (similaridade ou dissimilaridade) entre os elementos que serão agrupados. Uma das distâncias mais utilizadas é a Distância Euclidiana. A ideia básica é considerar cada elemento como um ponto no espaço euclidiano e, desse modo, o cálculo nos fornece a distância física entre esses dois elementos.

A Distância Euclidiana entre dois elementos X e Y , sendo $X = [X_1, X_2, \dots, X_n]$ e $Y = [Y_1, Y_2, \dots, Y_n]$, é definida por:

$$d_{xy} = \sqrt{(X_1 - Y_1)^2 + (X_2 - Y_2)^2 + \dots + (X_p - Y_p)^2} = \sqrt{\sum_{i=1}^p (X_i - Y_i)^2} \quad (1)$$

As distâncias entre cada um dos objetos é representada na forma de uma matriz simétrica. Essa matriz é chamada de matriz de similaridade.

Seja a Tabela 1 composta por 6 elementos, D será a matriz de similaridade entre esses elementos. D é obtida a partir do cálculo da Distância Euclidiana entre eles, logo, a distância do elemento 1 ao 4, por exemplo, é 2 (coluna 1, linha 4 da matriz).

Tabela 1 – Elementos a serem agrupados.

Elemento	X	Y
1	4	3
2	2	7
3	4	7
4	2	3
5	3	5
6	6	1

$$D_{rs} = \begin{bmatrix} 0 & 4,47 & 4,00 & 2,00 & 2,24 & 2,83 \\ 4,47 & 0 & 2,00 & 4,00 & 2,24 & 7,21 \\ 4,00 & 2,00 & 0 & 4,47 & 2,24 & 6,32 \\ 2,00 & 4,00 & 4,47 & 0 & 2,24 & 4,47 \\ 2,24 & 2,24 & 2,24 & 2,24 & 0 & 5,00 \\ 2,83 & 7,21 & 6,32 & 4,47 & 5 & 0 \end{bmatrix}$$

No primeiro momento, tem-se n grupos com 1 objeto cada, ou seja, cada objeto corresponde a um grupo. A cada ciclo do algoritmo, são agrupados os dois grupos que minimizam o incremento da Soma dos Quadrados dos Erros (SQE) em relação ao estágio anterior. Inicialmente, tem-se $n(n-1)/2$ pares de grupos. Ao aglomerar os grupos que atendem à função objetivo, a matriz de similaridade deve ser atualizada e então o processo é repetido (FERREIRA, 2011).

A SQE para um dado objeto dentro de um grupo é calculada por:

$$SQE_l = \sum_{i=1}^{n_l} \sum_{j=1}^{n_l} (X_{ij}^{(l)} - \bar{X}_{.j}^{(l)})^2 \quad (2)$$

onde,

$X_i^{(l)}$ é o i -ésimo objeto do l -ésimo grupo,

$l = 1, 2, \dots, k,$

$i = 1, 2, \dots, n_l,$

$j = 1, 2, \dots, n_l,$

n_l é o número de objetos do l -ésimo grupo,

n é o número total de objetos,

$\bar{X}_{.j}^{(l)}$ é a média do l -ésimo grupo relativa a j -ésima variável.

A SQE, para o conjunto de todos os grupos, é dada pela soma das SQEs dos grupos individuais em relação a todos os grupos, por:

$$SQE = \sum_{l=1}^k SQE_l \quad (3)$$

Considerando dois grupos quaisquer, R e S , em um dado estágio da execução do algoritmo, deve-se aglomerar o par que minimiza a SQE provocada pela junção de R e S que será denominada de ΔSQE . As SQEs dos demais grupos, exceto R e S permanecem inalteradas. Logo, a troca em SQE provocada por essa junção é dada por:

$$\Delta SQE = SQE_{rs} - SQE_r - SQE_s \quad (4)$$

Assim, deve-se determinar os grupos R e S tal que, ao serem aglomerados, minimizem a expressão 4, ou seja, denotem a menor dissimilaridade dentre todos os possíveis pares de grupos.

Seguindo o exemplo anterior, observando-se a matriz D , o par $(r = 1, s = 4)$ com proximidade 2, serão os primeiros a serem agrupados, pois apresentam a menor proximidade e foram os primeiros a aparecerem na matriz. Sendo assim, a matriz de similaridade deverá ser atualizada com as distâncias do novo agrupamento a cada um dos elementos restantes, conforme matriz D_2 .

$$D_2 = \begin{bmatrix} 0 & d(1,4)2 & d(1,4)3 & d(1,4)4 & d(1,4)5 & d(1,4)6 \\ d2(1,4) & 0 & 2,00 & 4,00 & 2,24 & 7,21 \\ d3(1,4) & 2,00 & 0 & 4,47 & 2,24 & 6,32 \\ d4(1,4) & 4,00 & 4,47 & 0 & 2,24 & 4,47 \\ d5(1,4) & 2,24 & 2,24 & 2,24 & 0 & 5,00 \\ d6(1,4) & 7,21 & 6,32 & 4,47 & 5 & 0 \end{bmatrix}$$

Do segundo passo em diante, o cálculo das distâncias é feito levando-se em consideração o ΔSQE para todo agrupamento R e S observado. Os passos são repetidos até que reste somente um grupo. Por fim, um dendograma é obtido como saída do método Ward, resultando no arranjo hierárquico final dos agrupamentos. A Figura 4 apresenta a saída para o exemplo dado. No eixo x são dispostos os objetos agrupados e no eixo y as distâncias entre esses objetos em cada ciclo. As ligações entre os objetos representam os grupos formados em cada interação, sendo que, quanto maior o número de interações, menor o número de grupos e maior a distância entre os objetos que compõem cada grupo. A distância máxima admitida pode determinar o número de grupos a serem considerados. A saber, para a distância = 2,69 tem-se 4 grupos, 5,37 têm-se 2 grupos e para a distância máxima igual a 8,06 tem-se 1 grupo.

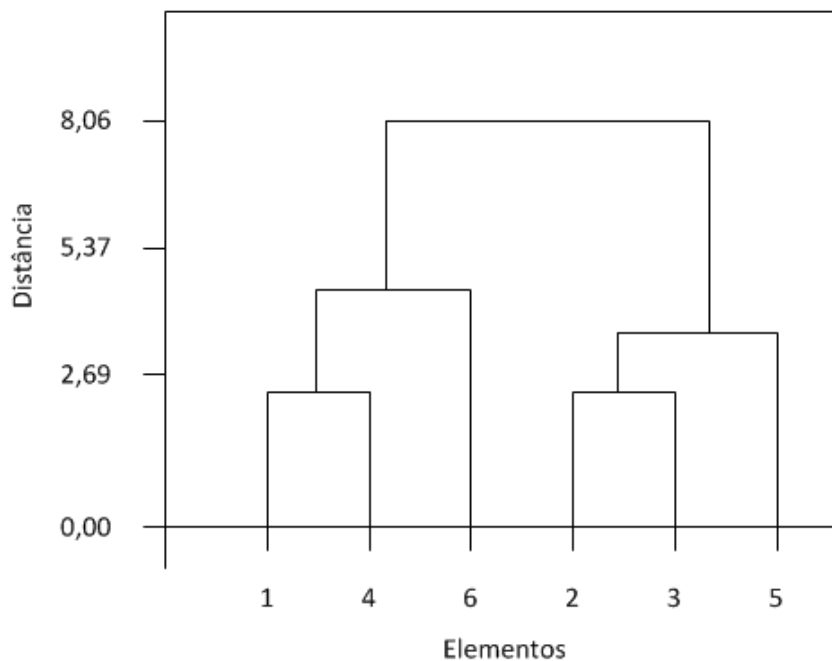


Figura 4 – Dendograma resultante do método Ward.

2.4 Ferramenta de Apoio ao Mapeamento Sistemático: Mendeley

Diversas ferramentas para apoiar trabalhos de mapeamento e revisões sistemáticas são propostas (MARSHALL; BRERETON; KITCHENHAM, 2013), no entanto, grande parte não apoia por completo todo o processo, principalmente no que diz respeito à análise dos estudos. Dessa forma, optou-se por utilizar um gerenciador de referências, o Mendeley¹, para coletar, organizar, manipular e classificar os artigos.

¹ <http://www.mendeley.com>

O Mendeley é um gerenciador de referências, criado em 2008 e distribuído gratuitamente, sendo necessário apenas um cadastro e registro no site para liberação do uso. O software consiste de uma aplicação desktop, Mendeley Desktop, e um website <<http://www.mendeley.com/dashboard>>. O aplicativo desktop permite gerenciar os artigos localmente e sincronizá-los com a versão web. A versão web permite o gerenciamento, compartilhamento, geração de estatísticas e backup desses artigos na nuvem.

O Mendeley auxilia, desde a obtenção dos metadados e arquivos PDF dos artigos através de bases de dados populares (por exemplo, IEEE Xplorer, ACM Digital Library, Science Direct) até a geração de referências. É possível gerar referências para editores de texto como o Microsoft Word, LibreOffice e também no formato bibtex para editores Latex. Também é possível realizar anotações, marcações e criar *tags* para classificar, de forma pessoal, os artigos.

Um benefício do Mendeley, que não é encontrado em outros gerenciadores de referências até então, é a facilidade em obter os dados completos dos artigos. Os dados tais como título, autores, periódico publicado, ano de publicação, volume, DOI, etc, são resgatados tanto através da busca feita dentro do próprio aplicativo quanto pelo acesso às bibliotecas digitais. A obtenção dos dados através das bibliotecas digitais é intermediada por um *plug-in* javascript, o Web Importer, que deve ser instalado na barra de tarefas do navegador. O *plug-in* é compatível com dezenas de bases de dados acadêmicas e permite selecionar um artigo ou uma lista de artigos para ser sincronizado na conta previamente cadastrada. Posteriormente, esses artigos estarão disponíveis tanto no ambiente web quanto no Mendeley Desktop.

O Mendeley também possibilita a exportação das informações (metadados dos artigos acrescidos de edições feitas pelo usuário) através de diferentes formatos, a saber, xml, bibtex e ris. As informações exportadas permitem que os dados extraídos possam ser portados e utilizados em outros sistemas.

2.5 Trabalhos Relacionados

Desde 2004, diferentes áreas em Engenharia de Software têm desenvolvido revisões e mapeamentos sistemáticos para apoiar suas pesquisas. Exemplos dessa utilização podem ser encontrados em Testes de Linha de Produtos de Software (ENGSTRÖM; RUNESON, 2011), Métodos de Avaliação de Usabilidade para Web (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), Métricas para Predição de Faltas (RADJENOVIC et al., 2013), entre outras.

Em (ENGSTRÖM; RUNESON, 2011) foi realizado um mapeamento sistemático para levantar estudos a respeito de testes em linhas de produtos de software (SPL). Os autores analisaram 64 estudos e os classificaram em relação ao foco, tipo de pesquisa e o tipo de contribuição. As evidências mostraram que é necessário uma maior atenção para métodos

de validação e avaliação para fornecer uma base melhor para testes de SPL.

Um mapeamento sistemático com o objetivo de resumir o conhecimento no que diz respeito a métodos de avaliação de usabilidade empregados a aplicativos web foi realizado em (FERNANDEZ; INSFRAN; ABRAHÃO, 2011). Foram selecionados 206 artigos cobrindo 14 anos de pesquisas na área. Os resultados mostraram que o método mais utilizado em aplicativos web são os Testes de Usuário. O trabalho pôde ainda identificar várias lacunas de pesquisa e fornecer uma visão geral do estado da arte na área. Dessa forma, novas atividades de investigação podem ser apropriadamente posicionadas.

Em (RADJENOVIC et al., 2013) foi realizada uma revisão sistemática com o objetivo de identificar métricas e avaliar sua aplicabilidade na previsão de faltas de software. A revisão inclui 106 artigos publicados entre 1991 e 2011. Os trabalhos foram classificados de acordo com as métricas e propriedades do contexto. Os resultados obtidos mostraram que métricas orientadas a objetos foram utilizadas quase duas vezes mais, em comparação com as métricas tradicionais de código fonte ou de processo. Os resultados mostraram, ainda, que mais estudos devem ser realizados em grandes sistemas de software industriais para encontrar métricas mais relevantes para a indústria e para saber quais as métricas devem ser utilizadas em cada contexto.

Alguns trabalhos de revisão sistemática utilizam análise de agrupamento para classificar os estudos analisados. (DINGSØYR et al., 2012), uma revisão sistemática sobre metodologias ágeis, utiliza o método Ward para agrupar os trabalhos conforme a citação entre os autores. Esse trabalho parte do princípio de que quando dois trabalhos citam os mesmos autores eles tendem a estar abordando o mesmo assunto. Dessa forma, conseguiram encontrar os principais temas discutidos na literatura ágil. (KEUPP; PALMIÉ; GASSMANN, 2012) utiliza o método Ward para agrupar os estudos conforme as palavras encontradas no título dos artigos. Esse trabalho extraiu de 342 artigos, 217 palavras relevantes contidas nos títulos. O agrupamento dessas palavras gerou 25 grupos de palavras que representam tipos de inovação utilizadas na gestão estratégica. A análise desses tópicos permitiu identificar inconsistências que podem incentivar novos trabalhos nessa área.

Buscando por pesquisas similares às apresentadas, na área de Confiabilidade de Software, apenas uma revisão sistemática foi encontrada (SINGHAL; SINGHAL, 2011). Essa revisão abrangeu 141 artigos oriundos de revistas, e teve foco nas cinco questões apresentadas na Tabela 2.

Q1 procurou determinar quantas e quais são as revistas onde são publicados os artigos sobre SRE. Q2 buscou identificar possíveis deficiências encontradas por pesquisadores em Confiabilidade de Software ao se buscar por trabalhos relacionados à suas pesquisas. A Q3 identificou limitações na internet e em bibliotecas digitais ao se procurar por artigos de Confiabilidade de Software. Q4 procurou identificar tendências e oportunidades para tópicos cujos artigos foram classificados (modelos, análises, uso de estatística, pla-

Tabela 2 – Questões de Pesquisa (SINGHAL; SINGHAL, 2011).

ID	Questões de Pesquisa
Q1	Quantas e quais revistas incluem artigos sobre Confiabilidade de Software?
Q2	Até que ponto os pesquisadores de Confiabilidade de Software estão conscientes da amplitude de potenciais fontes em Confiabilidade de Software?
Q3	Como identificar, de forma fácil, artigos relevantes em Confiabilidade de Software?
Q4	Quais são os temas de pesquisa e métodos em Confiabilidade de Software mais investigados e como isso mudou ao longo do tempo?
Q5	Quais são os métodos de pesquisa mais frequentemente aplicados e em qual contexto de estudo? Como isso mudou ao longo do tempo?

nejamento de testes, abordagens e outros). Q5, similarmente a Q4, buscou identificar as tendências e possíveis oportunidades para as diferentes abordagens de pesquisa em SRE. As abordagens consideradas foram: *survey*, trabalhos teóricos, trabalhos experimentais, estudos de caso, revisões e simulações.

Apesar de (SINGHAL; SINGHAL, 2011) terem dado uma importante contribuição para a SRE, observou-se que um estudo mais amplo poderia ser feito. Além do mais, (SINGHAL; SINGHAL, 2011) analisaram somente trabalhos publicados em revistas, desconsiderando importantes trabalhos de conferências relevantes. Dessa forma, o presente trabalho busca ampliar as questões propostas por (SINGHAL; SINGHAL, 2011) e considerar não só as revistas, mas também, conferências relevantes para a SRE. Este trabalho também busca identificar tendências acerca de tópicos e métodos de pesquisa, tal como a Q4 e Q5 de (SINGHAL; SINGHAL, 2011), porém, concentrou-se a busca em um período menor de observação, com o objetivo de maximizar a validade externa do estudo. A validade externa de um estudo é maximizada à medida que os efeitos observados no estudo podem ser aplicados fora dele, ou seja, a medida em que os resultados obtidos para uma amostra podem ser generalizados para toda a população (KITCHENHAM et al., 2007). Além disso, as tendências são detectadas observando-se um número maior de fatores e de tópicos que não foram observados em (SINGHAL; SINGHAL, 2011). Outros fatores que estendem esse trabalho, bem como as questões de pesquisa que o nortearão serão discutidas no Capítulo 3.

Metodologia

A metodologia adotada para a realização deste trabalho segue as diretrizes propostas em (KITCHENHAM, 2004), (KITCHENHAM et al., 2007), (BRERETON et al., 2007) e (PETERSEN et al., 2008) e foi dividida nas fases de Planejamento (Seção 3.1), Condução (Seção 3.2) e Disseminação (Seção 3.3).

3.1 Planejamento

Na fase de planejamento é necessário definir e projetar todos os elementos do mapeamento. O planejamento deste trabalho foi dividido em duas etapas: primeiro foram especificadas as questões de pesquisa e depois foi desenvolvido o protocolo do mapeamento. Os materiais e métodos relacionados a cada uma destas etapas serão apresentadas nas subseções seguintes.

3.1.1 Questões de Pesquisa

As questões de pesquisa expressam o que se deseja obter como resultado deste mapeamento. Elas foram baseadas nos critérios PICOC (Seção 3.2) a seguir:

- ❑ **População:** Estudos publicados em Engenharia de Confiabilidade de Software;
- ❑ **Intervenção:** Métodos, técnicas, softwares, modelos, abordagens;
- ❑ **Comparação:** Não se aplica;
- ❑ **Outcomes:** Sintetizar estudos, identificar tendências em SRE;
- ❑ **Contexto:** Confiabilidade em sistemas de software.

Sendo assim, as questões de pesquisa consideradas para este mapeamento sistemático são apresentadas a seguir.

Q1: Quais tópicos de pesquisa em SRE têm sido mais investigados?

SRE é uma área ampla e multidisciplinar. Ela herda conceitos vindos de várias disciplinas, tais como, engenharia, computação, estatística, entre outros. Essa característica abre margem para que variados campos sejam explorados, chamados aqui de tópicos de pesquisa. Diante dessa diversidade, é necessário identificar esses tópicos e analisá-los individualmente. Um dos objetivos da análise é verificar quais tópicos são mais investigados em detrimento de outros. Sendo assim, utilizou-se o método de agrupamento hierárquico de Ward, apresentado na seção 2.3, para agrupar os artigos de um mesmo tópico.

Q2: Quais tipos de pesquisa têm sido adotadas?

Pesquisas podem ser de diferentes tipos. Esses tipos delimitam a forma como os projetos ou planos de investigação são formulados. Além disso, sistematizam e orientam a revisão bibliográfica, a coleta e a análise de dados, assim como a discussão dos resultados (MARCONI; LAKATOS, 2010).

Investigar quais tipos de pesquisa são utilizados em determinada área é importante para caracterizar a forma como os projetos são conduzidos. Algumas áreas são notadamente teóricas, outras mais experimentais, outras mais opinativas. Descobrir qual tem sido os tipos de pesquisa adotados em SRE pode não só direcionar novas pesquisas, como também identificar a preferência ou carência de pesquisa de determinado tipo.

No contexto deste trabalho, entende-se por “tipo de pesquisa” um dos seguintes: teórico, empírico e ambos. Os trabalhos teóricos são aqueles que se dedicam a reconstruir teorias, ideias, conceitos, objetivando aprimorar fundamentos teóricos (DEMO, 1994). Trabalhos empíricos, por sua vez, manipulam, de forma planejada, fatores, para verificar o efeito provocado por essa manipulação (BARBETTA; REIS; BORNIA, 2010). Os tipos de trabalhos considerados como “ambos” são aqueles que abordam tanto aspecto teórico quanto empírico.

Q3: Qual é a origem dos autores mais produtivos em SRE?

Instituições que abrigam as pesquisas sobre SRE estão espalhadas pelo mundo todo. A origem dos autores está relacionada ao local onde eles estão afiliados, seja uma universidade, um centro de pesquisa, etc. É de interesse da comunidade científica, principalmente para pesquisadores novinhos, saber onde estão os demais pesquisadores em SRE. Isso é importante para estabelecer parcerias, saber quais tipos de problemas cada centro de pesquisa tem se dedicado a resolver, entre outros. Consequentemente, é possível identificar também, quais assuntos têm sido tratados por poucos ou nenhum desses centros. Saber onde esses pesquisadores estão é importante também para alunos de pós-graduação

interessados em SRE. Esses alunos podem precisar buscar possíveis orientadores e as respostas a esta questão de pesquisa pode ajudá-los nesse sentido. Por isso, foi apontado, através da afiliação dos autores e co-autores, a origem dos artigos publicados e mapeados os países onde essas instituições se encontram.

Q4: Qual é a origem dos trabalhos mais citados em SRE? O que eles abordam?

Os trabalhos mais importantes de uma área podem refletir aqueles temas que têm despertado maior interesse dos pesquisadores, os trabalhos de maior qualidade e também apontar os autores e instituições que têm sido mais influentes na área. Um trabalho será mais importante à medida em que for citado por outros trabalhos (WAZLAWICK, 2009). Trabalhos que são muito citados ao longo do tempo acabam se tornando trabalhos clássicos. A origem dos trabalhos diz respeito ao local onde ele foi realizado (instituição, país, etc). Assim, encontrar a origem desses trabalhos pode indicar onde estão as instituições mais importantes, quais delas são mais ativas, etc. Para isso, foram coletadas as citações de todos os trabalhos da amostra, investigados a afiliação dos seus autores e os tópicos os quais eles pertencem.

Q5: Com quem os autores mais influentes cooperam?

O crescimento e amadurecimento da pesquisa em uma determinada área não depende de apenas um pesquisador ou um grupo isolado. O conjunto de todos os pesquisadores, cada um com seus próprios interesses é que fomenta e introduz avanços na área pesquisada. Esses avanços são reforçados por cooperações que os grupos de pesquisa estabelecem entre si. Essas cooperações pleiteiam uma maior integração e melhores resultados para os problemas identificados. Nesse sentido, é importante saber onde estão os grupos considerados influentes em SRE e com quais outros grupos eles cooperam. Esta questão de pesquisa visa identificar essas relações através da investigação das afiliações dos autores e co-autores de cada artigo. Dessa forma, é possível perceber as cooperações internas (autores que possuem a mesma afiliação) e cooperações externas (autores que estão no mesmo artigo, porém possuem afiliações diferentes).

Q6: Quais são os tópicos de pesquisa emergentes em SRE? Quais os tópicos de pesquisa saturados?

Muitas evidências de pesquisa são encontradas sobre SRE. Podem existir tópicos que já foram bastante investigados, como podem haver aqueles que ainda carecem de investigação. Ou ainda podem existir aqueles que estão emergindo e poderão ser tendência de pesquisa no futuro. Sendo assim, investigar essas questões pode ajudar a nortear pesquisas futuras.

Dessa forma, esta questão tem como objetivo apontar, dentre os tópicos identificados, qual é a possível direção que a pesquisa em SRE pode tomar nos próximos anos. Essa direção será investigada através da análise dos fatores citados nas questões anteriores, podendo ser mensurada através da identificação de tópicos emergentes e tópicos que estão se saturando. A indicação de tópicos emergentes e saturados está relacionado com os estudos contidos na amostra. Ou seja, se um determinado tópico da amostra apresenta um crescimento no número de estudos ao longo do tempo, considera-se esse tópico emergente. Por outro lado, se a quantidade de estudos de um determinado tópico diminui com o passar do tempo assume-se que ele está se saturando.

3.1.2 Protocolo do Mapeamento

Depois de determinadas as questões de pesquisa, foi desenvolvido o protocolo do mapeamento. O protocolo tem o objetivo de especificar as atividades a serem realizadas para conclusão do Mapeamento Sistemático.

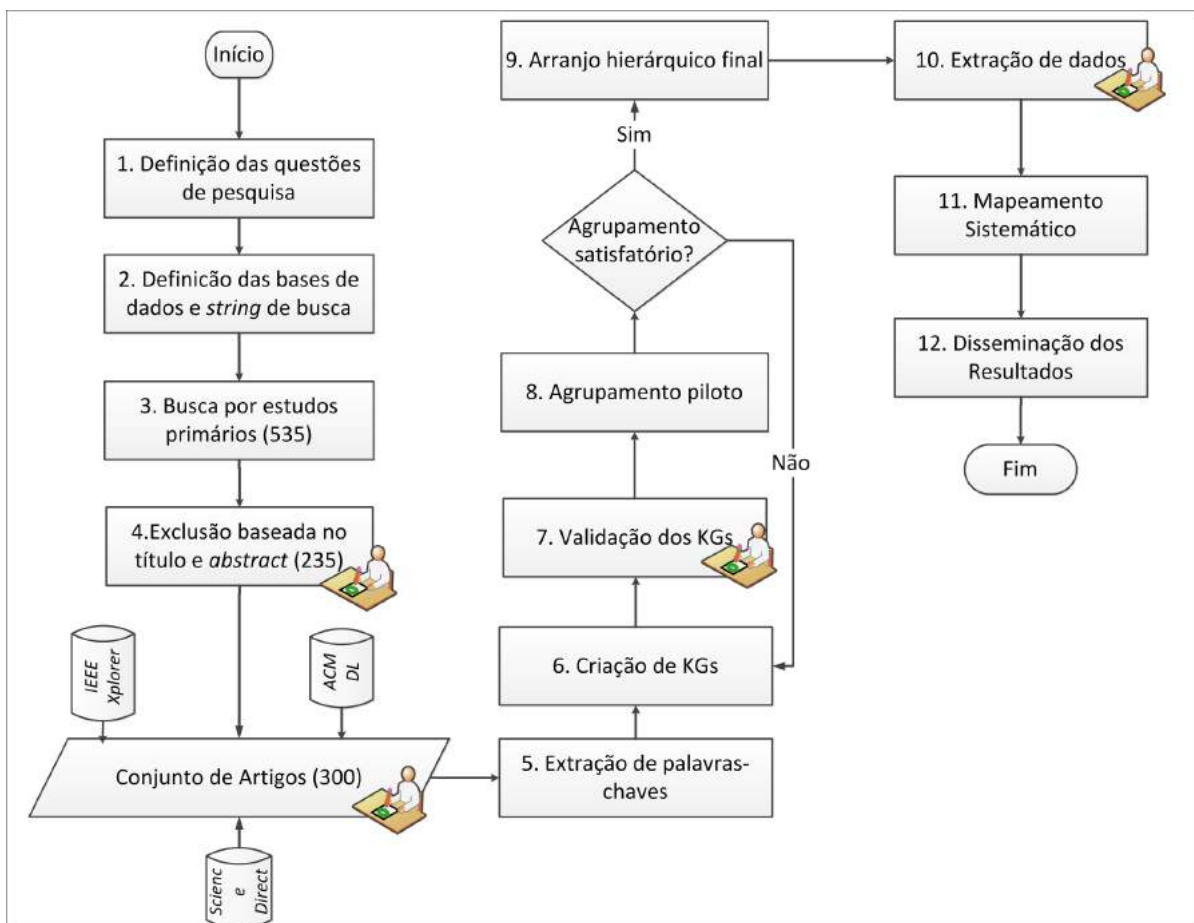


Figura 5 – Protocolo Mapeamento Sistemático

A Figura 5 apresenta o protocolo desenvolvido para conduzir o mapeamento. A figura identifica todas as etapas a serem cumpridas na realização do trabalho. Cada etapa é

identificada por um número. A etapa 1 e o desenvolvimento do protocolo, fazem parte desta fase de Planejamento. As etapas 2 a 11 compõem a fase de Condução do Estudo que será discutida na Seção 3.2. A etapa 12 é destinada a divulgar os achados deste estudo e será apresentada na Seção 3.3. As etapas que possuem um ícone de aprovação são aquelas que precisaram passar por um processo de validação antes que se prosseguisse para a etapa seguinte. Todas as validações foram feitas por especialistas da área. A saber, o orientador e o coorientador desta dissertação. Todos os impasses foram discutidos até que se chegasse a um consenso.

3.2 Condução do Estudo

Esta seção apresentara todas as etapas que foram realizadas durante a pesquisa. Essas etapas são descritas na Figura 5 nas etapas 2 a 11 e serão detalhadas nesta seção.

A **busca por estudos primários** iniciou-se com a **definição das bases de dados e da *string* de busca**. Utilizou-se três bases de dados para buscar os estudos: IEEE Xplore¹, ACM Digital Library² e Science Direct³. Essas bases de dados foram escolhidas porque congregam os principais veículos de publicação sobre SRE apontados por especialistas. Alguns exemplos desses veículos são: *International Symposium on Software Reliability Engineering* (ISSRE), *IEEE Transactions on Reliability*, *International Conference on Dependable Systems and Networks* (DSN), entre outros. Os demais veículos podem ser encontrados no Apêndice D.

Devido a especificidade da busca em cada uma das bases de dados, foram aplicadas *strings* diferentes em cada base. As *strings* aplicadas aos metadados dos artigos em cada uma das bases de dados foram as seguintes:

IEEE Xplore: ((software) AND reliability);

ACM Digital Library: (software) and (reliability) and (PublishedAs:journal OR PublishedAs:proceeding);

Science Direct: pub-date > 2002 and pub-date < 2014 and TITLE-ABSTR-KEY(software) and TITLE-ABSTR-KEY(reliability).

A busca foi refinada pelo período de publicação considerado, 2002 até 2013 e por aqueles autores que continham pelo menos 10 publicações nesse período em cada base de dados. Esse período de publicação foi escolhido porque julgou-se que 12 anos seria tempo suficiente para dar uma visão geral da área e apontar tendências futuras. A limitação do número de publicações foi necessária para que o número de artigos resultante fosse viável para a realização da pesquisa. Além do mais, o objetivo do estudo é levantar os trabalhos

¹ <http://ieeexplore.ieee.org>

² <http://dl.acm.org>

³ <http://www.sciencedirect.com>

recentes, dessa forma, considerou-se 12 anos tempo suficiente para este fim. Sendo assim, esta busca retornou um total de 535 artigos.

3.2.1 Seleção de Estudos Primários

Depois de coletados os artigos, foi necessário fazer uma triagem da coleção. Essa triagem foi feita levando-se em consideração os critérios de inclusão/exclusão. Os critérios de inclusão estão relacionados à busca automática dos artigos nas bases de dados e os de exclusão à triagem feita depois que os artigos foram obtidos. Sendo assim, considerou-se os seguintes critérios:

Critérios de inclusão: Foram incluídos artigos de conferências e revistas, publicados entre 2002 e 2013 em veículos indexados pelo IEEE Xplore, ACM Digital Library ou Science Direct reportando estudos empíricos ou descobertas teóricas a respeito de Confiabilidade de Software.

Critérios de exclusão: Foram excluídos trabalhos que não são estudos primários, trabalhos que tratam de confiabilidade em um contexto diferente de software, trabalhos que não tratam de software diretamente e trabalhos duplicados.

Os critérios de inclusão e exclusão foram aplicados a partir da leitura do título e *abstract* dos artigos. Quando a **leitura do abstract** não apresentava as informações suficientes, analisou-se o texto completo. As decisões de inclusão ou exclusão de um artigo foram validadas por um especialista (o orientador desta pesquisa) e os impasses existentes resolvidos por outro (o coorientador deste estudo). No final desse processo obteve-se um **conjunto de 300 artigos**.

3.2.2 Extração de Dados e Agrupamento dos Estudos

A extração dos dados necessários para responder às questões de pesquisa e mapear os estudos, foi feita através da leitura e obtenção das palavras-chave dos artigos. Inicialmente os artigos foram agrupados por palavras-chave. O objetivo desse agrupamento foi detectar os temas de pesquisa dos artigos da amostra.

Foram **extraídas as palavras-chave** dos 300 artigos da amostra. As palavras duplicadas e aquelas que não tinham valor significativo para SRE foram excluídas, tais como termos genéricos como “software engineering”, nomenclaturas e siglas específicas de um trabalho, como “SWR strategy”, “CASRE”, entre outros, restando 78 palavras. Algumas palavras-chave podem ser consideradas sinônimas dentro de um contexto, por exemplo, “detecção de faltas”, “minimização de faltas”, “correção de faltas” e “tolerância à faltas”, em SRE, são termos que podem ser representados por um único termo mais genérico, tal como “faltas de software”. Sendo assim, tais palavras foram classificadas em um mesmo

grupo, denominado *Keyword Group* (KG). Foram então **criados 40 KGs** a partir das *keywords* dos artigos. A Tabela 3 mostra o exemplo de alguns KGs. A primeira coluna fornece as palavras-chave que serão consideradas na classificação, a segunda coluna indica a qual KG aquela palavra pertence. As células que foram mescladas referem-se a palavras-chave que estão em um mesmo KG. Uma lista com os demais KGs é apresentada no Apêndice B. Os KGs foram validados pelos especialistas. De posse das *keywords* que compõem um KG, foram verificadas a presença e/ou ausência dos KGs para cada artigo.

Tabela 3 – Exemplo de grupos de palavras-chave (KGs).

Palavras-Chave	KGs
analytical models	KG1
application software	
software packages	KG2
software tools	
optimization methods	KG3
optimization algorithms	
computational modeling	KG4
computer architecture	KG5
debugging	KG6
imperfect debugging	
program debugging	

Um **agrupamento piloto** foi gerado com a finalidade de avaliar o rigor do método. Os agrupamentos formados foram validados de forma a confirmar se os artigos que tinham sido agrupados em um mesmo grupo discutiam de fato, temas em comum. Essa verificação foi feita tomando, aleatoriamente, uma amostra de aproximadamente 5 estudos de cada agrupamento e verificando se eles realmente discutiam a temática proposta pelo agrupamento. A relação foi confirmada para todos os estudos. Se essa relação não tivesse sido confirmada, outras formas de agrupamento deveriam ser testadas e o processo repetido até que todos os agrupamentos fossem satisfatórios. Por fim, os temas identificados como “em comum” com todos os artigos foram os responsáveis pela nomeação do tópico.

Utilizou-se, neste trabalho, o Método Ward (Seção 2.3) para agrupar os artigos semelhantes. A matriz resultante do mapeamento de KGs é a entrada para o método Ward. Um exemplo dessa matriz é apresentado na Tabela 4. Nas linhas, encontram-se os artigos e, nas colunas, a identificação dos grupos. Cada artigo pode ser associado a um ou mais KGs. Um artigo é associado ao KG_t , sendo $1 \leq t \leq 40$, se pelo menos uma de suas palavras-chave for uma daquelas que compõem o KG_t . Sendo assim, a presença de pelo menos uma palavra de determinado grupo é representado por 1 e a ausência por 0.

O resultado do mapeamento dos artigos para os KGs foi utilizado como entrada para o método Ward. Cada artigo representa um objeto no plano euclidiano. As coordenadas desses objetos são dadas através da presença ou ausência de cada KG. A Figura 6

Tabela 4 – Exemplo de mapeamento de KGs para os artigos.

Artigo	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	4
Two-dimensional software reliability measurement technologies	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Change-point modeling for software reliability assessment depending on two types of reliability growth factors	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
An Adaptive Reliability Analysis Using Path Testing for Complex Component-Based Software Systems	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Software reliability accelerated testing method based on test coverage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

apresenta os agrupamentos obtidos com a execução do método Ward, via Minitab⁴. A figura mostra o dendograma com arranjo hierárquico resultante da execução do método. Os objetos no eixo x representam cada um dos 300 artigos. No eixo y são mostradas as distâncias consideradas em cada uma das fases de execução do algoritmo. Quanto menor é a distância, mais homogêneos são os agrupamentos.

A Figura 6 destaca sete agrupamentos. Esse foi o número de agrupamentos escolhido para levar-se em consideração na pesquisa. Em outras palavras, foram identificados sete tópicos de pesquisa distintos. Escolheu-se 7 grupos, tal como sugerido em (MILLER, 1956), por considerar razoável a homogeneidade conferida a esses grupos. Um número menor de grupos implicaria em artigos mais distantes no mesmo grupo. Um número maior resultaria em artigos com temas próximos em grupos separados. Sendo assim, a distância máxima para os grupos considerados é de 9,43.

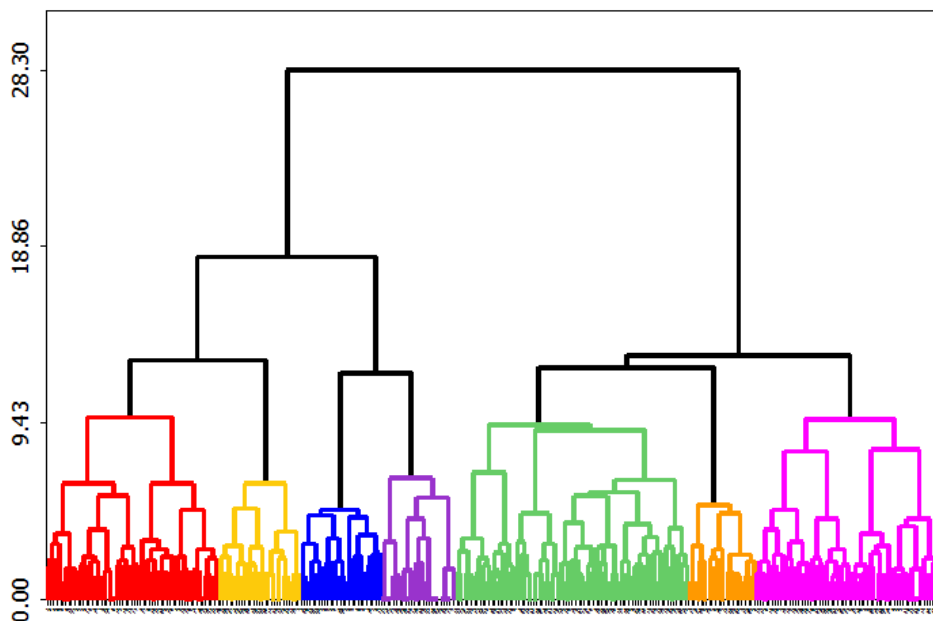


Figura 6 – Dendograma resultante da execução do método Ward.

Depois dos artigos terem sido agrupados, foram **extraídos os dados** necessários para se responder às questões, culminando no **Mapeamento Sistemático**.

⁴ <http://minitab.com>

3.3 Disseminação

A fase de disseminação é a última do mapeamento sistemático. Ela está centrada na produção de relatórios para divulgação e circulação dos resultados do estudo. Comunicar os resultados de um mapeamento sistemático é importante pois eles precisam estar disponíveis para as partes potencialmente interessadas. Para isso, optou-se pela publicações dos resultados em anais de conferências. Os relatórios são formatados seguindo os padrões estabelecidos por cada conferência. Os tipos de gráficos e estrutura foram inspirados em (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), (BRERETON et al., 2007) e (DOGAN; BETIN-CAN; GAROUSI, 2014). Esta dissertação por si só já é um meio de disseminação. As demais contribuições serão abordados no Capítulo 5.

Resultados

Neste capítulo serão apresentados os resultados obtidos com a realização da pesquisa. Investigações foram feitas com o objetivo de responder cada uma das questões apresentadas na Seção 3.1.1.

Considerando o período amostral, a Figura 7 apresenta o número de estudos publicados por ano. Pode-se observar que, entre 2006 e 2011, a cada dois ou três anos há um crescimento do número de artigos. Uma explicação para esses picos é que isso ocorre devido ao tempo necessário para os autores amadurecerem seus trabalhos de pesquisa, que parece ser em média 2 anos. De acordo com o padrão observado, considerou-se normal a descendência da linha a partir de 2011, onde o próximo pico esperado seria no ano de 2013. Contudo, a coleta dos artigos para compor a amostra terminou em julho de 2013, onde muitas conferências e revistas ainda não haviam publicado os manuscritos aceitos em 2012. Adicionalmente, o pequeno número de artigos em 2013 está relacionado ao fato de que muitas conferências ainda estavam acontecendo.

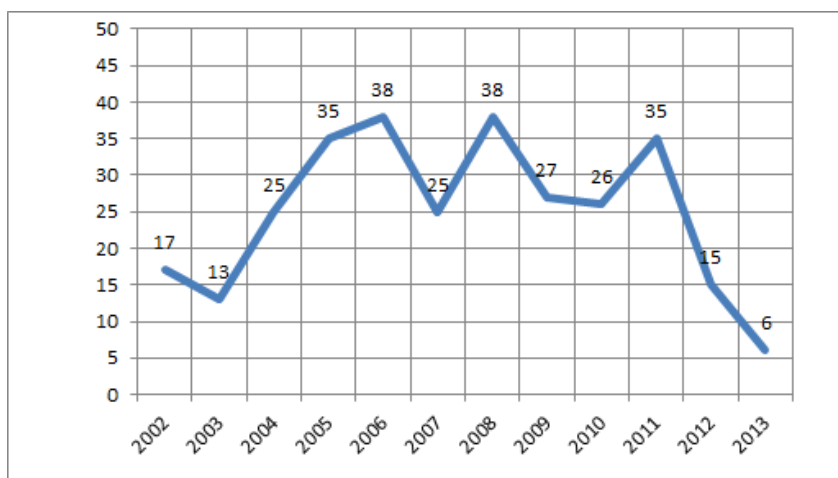


Figura 7 – Número de estudos por ano.

A Figura 8 apresenta distribuição de estudos pelos 20 veículos de publicação que mais tiveram artigos no conjunto de dados. O eixo y se refere aos veículos de publicação e o eixo

x ao número de artigos da amostra que foram publicados em cada um desses veículos. Os 20 veículos se dividem em dois tipos: conferências e periódicos. As conferências foram referenciados através das siglas e os periódicos pelo ISSN. Esses dois tipos são diferenciados por cores distintas. Para cada veículo foi identificado o respectivo extrato da Capes. Por exemplo, o ISSRE é uma conferência com Qualis A2. Vale lembrar que o extrato Qualis é uma classificação feita por uma agência de fomento brasileira, a Capes, dessa forma, é natural que veículos mais novos ou que possuam poucas publicações não entrem nessa classificação. As classificações foram obtidas do extrato Qualis consultados em Abril/2015¹. Uma lista com os nomes dos veículos referenciados e os demais veículos de publicação pode ser encontrada no Apêndice D.

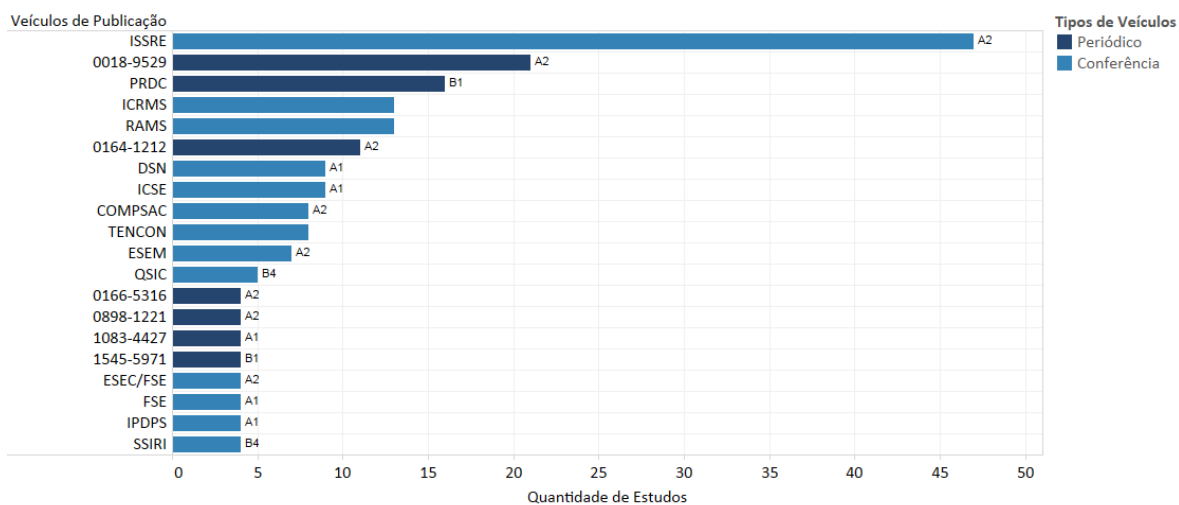


Figura 8 – Distribuição dos artigos da amostra pelos veículos de publicação.

Nas seções seguintes serão discutidos os achados pertinentes a cada um das questões de pesquisa apresentadas na Seção 3.1.1.

4.1 Q1 - Quais tópicos de pesquisa em SRE têm sido mais investigados?

Os 300 estudos foram divididos em 7 agrupamentos distintos através do Método Ward (Seção 2.3). Os agrupamentos foram nomeados de acordo com as palavras-chave mais frequentes. A Figura 9 apresenta esses agrupamentos. A Figura 9 é similar à Figura 6 que foi apresentada no Capítulo 3 com o resultado da execução do método Ward. A diferença entre elas é que na Figura 9 foram acrescentados as siglas referentes à nomeação de cada um dos agrupamentos (diferenciados por cores distintas), a saber:

vermelho = Modelagem Estocástica (ME);

¹ <http://qualis.capes.gov.br/>

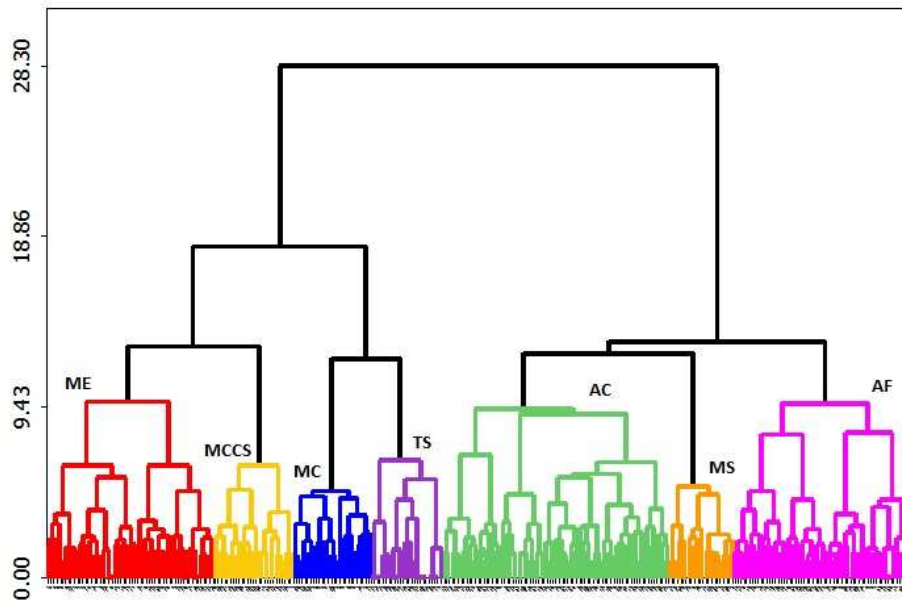


Figura 9 – Dendrograma dos 300 artigos selecionados com nomeação dos grupos.

amarelo = Modelagem de Crescimento de Confiabilidade de Software (MCCS);

azul = Modelagem de Confiabilidade (MC);

roxo = Testes de Software (TS);

verde = Avaliação de Confiabilidade (AC);

laranja = Manutenção de Software (MS);

rosa = Análise de Faltas (AF).

Os nomes dados aos agrupamentos são referentes aos assuntos predominantes em cada um deles. Investigou-se os KGs mais recorrentes em cada agrupamento para se chegar a uma nomenclatura apropriada ao tema central. Porém, um agrupamento pode conter outros temas secundários, fazendo com que determinado assunto não seja exclusivo do agrupamento para o qual foi nomeado. Por exemplo, o agrupamento “Modelagem Estocástica” recebeu esse nome porque 92% dos artigos desse agrupamento apresentam palavras-chave do KG relacionado a modelos estocásticos (KG15). Outros KGs, tais como, KG8, KG19 e KG25, também estão contemplados nesse agrupamento. Contudo, o KG predominante é o KG15, que deu nome a esse agrupamento. Cada artigo é mapeado para apenas um agrupamento, logo, é possível que um estudo trate, por exemplo, de “Avaliação de Confiabilidade” e também de “Modelagem Estocástica”. O método Ward buscará mapeá-lo para um dos dois, prevalecendo aquele que é mais recorrente no mesmo.

Os estudos presentes na amostra cobriram 304 autores de 92 afiliações diferentes, distribuídos entre 15 países. A Figura 10 mostra a distribuição dos trabalhos entre os agrupamentos. O agrupamento que apresenta maior volume de artigos é AC, com 26% dos artigos, seguido de AF com 21%. Os menores são MS e TS, ambos com 8%.

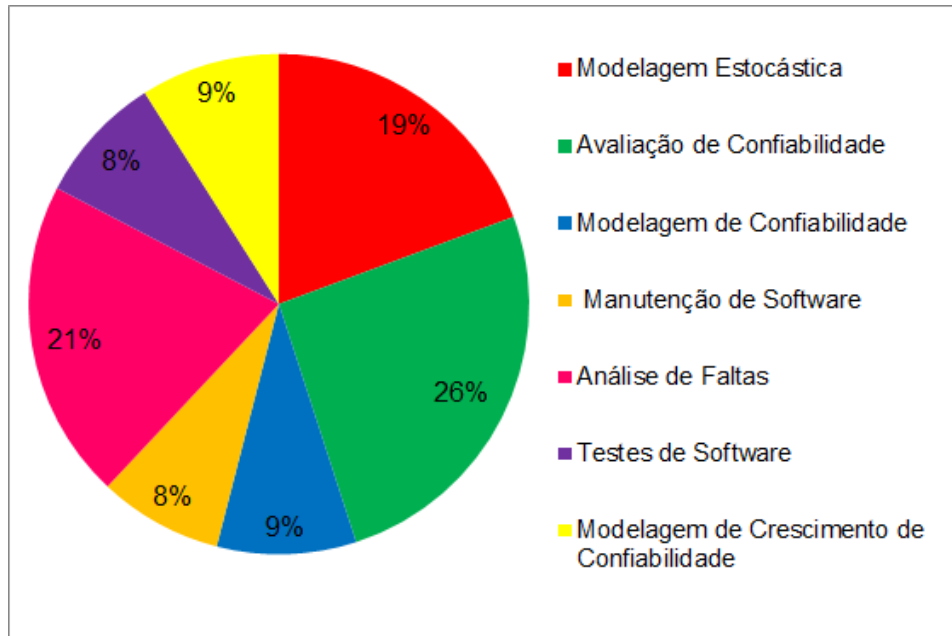


Figura 10 – Distribuição dos estudos por agrupamento.

4.2 Q2 - Quais tipos de pesquisa têm sido adotadas?

Entende-se por tipo de pesquisa a abordagem utilizada pelos autores na realização dos trabalhos. Para este trabalho foram consideradas três tipos de abordagem:

- ☐ teórica – aquela em que o autor apresenta um corpo conceitual, sem a realização de experimentos;
- ☐ empírica – o autor realiza experimentos sobre um corpo conceitual apresentado por terceiros ou por em um trabalho anterior;
- ☐ teórico-empírica (ambos) – o autor propõe um corpo conceitual e os avalia empiricamente.

A Tabela 5 apresenta a distribuição dos tipos de pesquisa através dos agrupamentos. Para cada agrupamento (primeira coluna) é dado a porcentagem de artigos dentro de cada um dos tipos de pesquisa (colunas 2, 3 e 4) em relação ao todo (amostra de 300 artigos). Observa-se através da Tabela 5 que os trabalhos empíricos e teórico-empíricos (coluna “Ambos”) totalizam 31%. Enquanto os trabalhos puramente teóricos representam quase 69% dos artigos. A Tabela 6, por outro lado, apresenta a porcentagem para cada tipo de pesquisa em cada agrupamento. O significado das colunas dessa tabela é o mesmo da Tabela 5. Por exemplo, dos 58 artigos presentes em “Modelagem Estocástica”, 81% são teóricos, 7% são empíricos e 12% são teórico-empíricos. Os agrupamentos “Análise de Falhas” e “Testes de Software” congregam a maioria dos trabalhos empíricos. Em contrapartida, os agrupamentos “Manutenção de Software” e “Modelagem Estocástica” são os que mais possuem trabalhos teóricos.

Tabela 5 – Distribuição dos estudos por tipo de pesquisa.

Agrupamento	Teórico	Empírico	Ambos
Modelagem Estocástica	15,67%	1,33%	2,33%
Avaliação de Confiabilidade	17,67%	3,00%	5,00%
Modelagem de Confiabilidade	5,00%	1,33%	2,67%
Manutenção de Software	6,33%	0,67%	1,00%
Análise de Falhas	14,00%	3,67%	3,00%
Testes de Software	4,00%	1,33%	3,00%
Modelagem de Crescimento de Confiabilidade	6,33%	0,67%	2,00%
TOTAL	69,00%	12,00%	19,00%

Tabela 6 – Distribuição de tipos de estudo por agrupamento.

Agrupamento	Teórico	Empírico	Ambos
Modelagem Estocástica	81%	7%	12%
Avaliação de Confiabilidade	68%	12%	20%
Modelagem de Confiabilidade	55%	15%	30%
Manutenção de Software	79,17%	8,33%	12,50%
Análise de Falhas	68%	18%	14%
Testes de Software	48%	16%	36%
Modelagem de Crescimento de Confiabilidade	70%	8%	22%

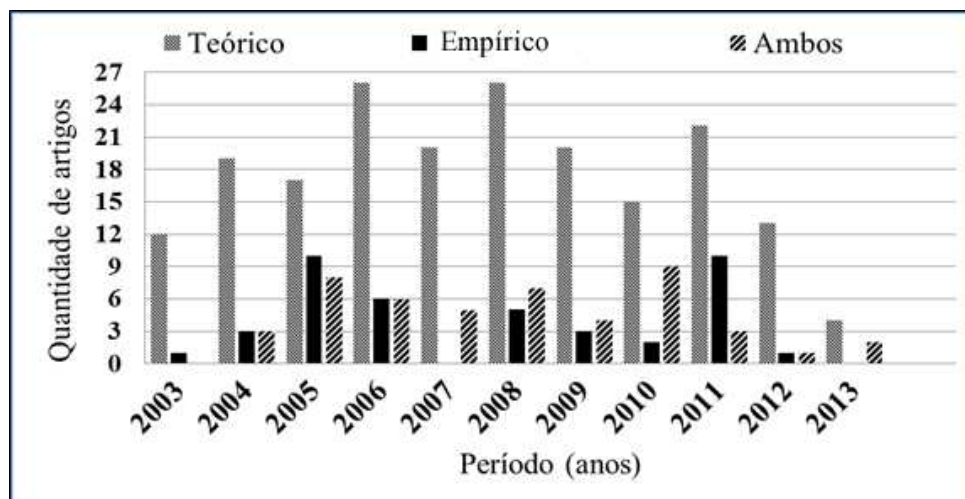


Figura 11 – Número de trabalhos teóricos, empíricos e ambos por ano.

A Figura 11 apresenta a distribuição dos tipos de pesquisa ao longo do tempo. O eixo x representa o período de publicação em anos e o eixo y a quantidade de artigos. Em geral, os principais tópicos encontrados nos trabalhos teóricos são relacionados à Modelagem de Confiabilidade. Esses trabalhos tratam, principalmente, de Modelagem de Crescimento de Confiabilidade e Modelos Preditivos. Grande parte dos trabalhos empíricos abrangem Sistemas Embarcados, Diagnóstico de Falhas, Tolerância à Falhas, e Cobertura de Código

em Testes. Por outro lado, os trabalhos teórico-empíricos foram mais relacionados com processos de Poisson não Homogêneos e Testes de Software.

4.3 Q3 - Qual a origem dos autores mais produtivos em SRE?

A Tabela 7 apresenta a afiliação dos dez autores mais produtivos em termos de quantidade de publicação. Cada linha da tabela representa um autor, porém optou-se por preservar o nome dos autores e identificar apenas a instituição ao qual eles pertencem. Os autores são identificados portanto, pela instituição a que são afiliados (primeira coluna) seguido do país sede da mesma (segunda coluna). A terceira coluna, Número de Trabalhos Publicados (NTP), indica o número de trabalhos publicados pelo autor no período pesquisado (2002-2013), e a quarta coluna, Média, apresenta o número médio de citações desses trabalhos. Isto é, esse número corresponde à soma do número de citações dos trabalhos do autor, dividido pelo número de trabalhos publicados por ele. Observa-se que duas linhas dessa tabela mostram a mesma afiliação: Hiroshima University. Isto significa que dois autores entre os mais influentes são afiliados a essa instituição.

Tabela 7 – Afiliação dos dez autores mais produtivos.

Afiliação	País	NTP	Média
Hiroshima University	JPN	56	6,80
Duke University	USA	49	34,04
National Tsing Hua University	TWN	48	22,02
Hiroshima University	JPN	34	6,79
Chinese University of Hong Kong	HKG	27	33,90
University of Connecticut	USA	25	25,70
Tottori University	JPN	25	9,08
Indian Inst. of Tech. of Kanpur	IND	18	15,44
North Carolina State University	USA	18	46,83
Beihang University	CHN	16	2,00

4.4 Q4 - Qual é a origem dos trabalhos mais citados em SRE?

Os trabalhos mais citados tendem a ser considerados os mais importantes dentro de uma área. Dessa forma, foi criado um *ranking* das instituições onde estão os autores que mais têm trabalhos citados na amostra. As citações dos trabalhos foram coletadas em novembro de 2013 e podem ter sofrido alterações desde então.

A Tabela 8 apresenta a afiliação dos dez autores mais citados no conjunto de dados analisado. A descrição da Tabela 8 é semelhante à da Tabela 7, exceto pela quinta coluna, Número de Citações (NC). Esta corresponde à soma das citações de todos os trabalhos publicados pelo autor (coluna NTP), de acordo com o Google Scholar². Por exemplo, a soma das citações para os 49 trabalhos do autor da Duke University é 1668.

Tabela 8 – Afiliação dos dez autores mais citados.

Afiliação	País	NTP	NC
Duke University	USA	49	1668
National Tsing Hua University	TWN	48	1057
Chinese University of Hong Kong	HKG	27	916
North Carolina State University	USA	18	843
University of Connecticut	USA	25	643
Microsoft Research	USA	4	436
Duke University	USA	5	436
National Tsing Hua University	TWN	10	423
Hiroshima University	JPN	56	381
Duke University	USA	3	297

A Tabela 9 apresenta os três trabalhos mais citados para cada agrupamento. Para cada agrupamento (primeira coluna), são referenciados os três trabalhos mais citados (segunda coluna), o veículo onde aquele trabalho foi publicado (terceira coluna) e os respectivos números de citação (quarta coluna). Os trabalhos na segunda coluna foram indicados através de números. Esses números (IDs) correspondem a artigos contidos na seção Referências Bibliográficas (RBs). A Tabela 10 apresenta a relação entre IDs e artigos das RBs. Nessa tabela, por exemplo, o artigo de ID [1] corresponde ao artigo (NICOL; SANDERS; TRIVEDI, 2004), contido na RB.

No agrupamento “Modelagem Estocástica” (ME), os dois primeiros trabalhos são teóricos e o terceiro é teórico-empírico. Em “Avaliação de Confiabilidade” (AC), o segundo trabalho é teórico-empírico e o primeiro e o terceiro são puramente teóricos. No agrupamento “Modelagem de Confiabilidade” (MC), o primeiro e o terceiro são trabalhos empíricos e o segundo é teórico. Em “Manutenção de Software” (MS), o primeiro trabalho é teórico-empírico e os outros são somente teóricos. Dos três trabalhos mais citados do agrupamento “Análise de Faltas” (AF), o primeiro e o terceiro são teóricos e o segundo é empírico. No agrupamento “Testes de Software” (TS), o trabalho mais citado é empírico e o segundo e terceiro são teórico-empíricos. No agrupamento “Modelagem de Crescimento de Confiabilidade” (MCCS), o primeiro e o terceiro são ambos teórico-empíricos, e o segundo trabalho mais citado é do tipo teórico.

² <<http://scholar.google.com.br>>

Tabela 9 – Os três trabalhos mais citados por agrupamento.

Agrupamento	ID Artigo	Veículo de Publicação	NC
ME	[1]	IEEE Trans. on DSC 2004	284
	[2]	IEEE Trans. on SE 2003	122
	[3]	IEEE Transactions on Reliability 2002	101
AC	[4]	ACM/IEE 27th ICSE 2005	282
	[5]	ACM/IEE 30th ICSE 2008	141
	[6]	IEEE Transactions on,DSC 2007	140
MC	[7]	ACM Symp. on STA 2004	104
	[8]	IEEE Transactions on Reliability 2006	68
	[9]	IEEE Transactions on Software 2005	62
MS	[10]	Int'l. Symp. on Empirical SE 2002	121
	[11]	Symp. on Reliab. and Maintain. 2004	54
	[12]	IEEE Int'l.Workshop on Sw. Aging	42
AF	[13]	IEEE Transactions on DSC 2005	140
	[14]	Workshop on ASSISD 2006	115
	[15]	IEEE Int'l.Conf. on ASSAP 2004	58
TS	[16]	ESE Conf. + ACM Symp.on FSE 2011	36
	[17]	IEEE Trans. on,SE 2010	22
	[18]	IEEE Transactions on Reliability 2010	20
MCCS	[19]	Journal of Systems and Software 2007	75
	[20]	Journal of Systems and Software 2006	38
	[21]	Journal of Systems and Software 2008	28

4.5 Q5 - Com quem os autores mais influentes cooperam?

Os autores mais influentes, aqueles identificados como os mais citados na Q4 (Seção 4.4), foram investigados com relação às suas cooperações com outros autores. Essa investigação foi centrada, principalmente, em verificar as cooperações com outras instituições. A Figura 12 ajuda a identificar onde estão os pesquisadores mais citados e com quais instituições eles cooperam. Nessa figura, cada nó representa um centro de pesquisa e as arestas representam a coautoria nos trabalhos. Os nós são rotulados pelos acrônimos das respectivas instituições. Foram adotados os acrônimos de nomes conhecidos mundialmente. O significado de todos os acrônimos utilizados são descritos no Apêndice C.

Os nós sólidos (pretos) representam uma afiliação que contém um ou mais autores influentes. O número acima do acrônimo da afiliação representa a quantidade de autores influentes dentro daquela instituição. Além disso, é importante notar que algumas afiliações têm mais de um autor influente, como é o caso de NTHU e DUKE.

Cada nó também possui um número abaixo do seu rótulo, o qual se refere à quantidade de pesquisadores afiliados que publicaram trabalhos em SRE pelo menos uma vez. As

Tabela 10 – Referências dos artigos citados na Tabela 9.

ID Artigo	Referência
[1]	(NICOL; SANDERS; TRIVEDI, 2004)
[2]	(HUANG; LYU; KUO, 2003)
[3]	(HUANG; KUO, 2002)
[4]	(NAGAPPAN; BALL, 2005)
[7]	(NAGAPPAN; MURPHY; BASILI, 2008)
[6]	(HUANG; LIN, 2006)
[7]	(SAFF; ERNST, 2004)
[8]	(HUANG; LIN, 2006)
[9]	(GOKHALE; LYU, 2005)
[10]	(LI; VAIDYANATHAN; TRIVEDI, 2002)
[11]	(AMARI; MCLAUGHLIN, 2004)
[12]	(GROTTKE; MATIAS; TRIVEDI, 2008)
[13]	(VAIDYANATHAN; TRIVEDI, 2005)
[14]	(LI et al., 2006)
[15]	(XIE et al., 2007)
[16]	(YIN et al., 2011)
[17]	(PIETRANTUONO; RUSSO; TRIVEDI, 2010)
[18]	(MATIAS et al., 2010)
[19]	(SU; HUANG, 2007)
[20]	(LO; HUANG, 2006)
[21]	(LIN; HUANG, 2008)

arestas representam a colaboração em coautoria nos trabalhos em SRE. Sejam A e B nós, não necessariamente distintos, esses nós são ligados por uma aresta se existir pelo menos uma cooperação (trabalhos publicados em conjunto) entre autores dessas instituições. Uma aresta é rotulada por três números:

- (i) o número mais próximo de A, representa a quantidade de trabalhos que os pesquisadores de A têm em coautoria com os autores influentes de B;
- (ii) o número no meio, representa a quantidade de trabalhos que os autores influentes de A têm em coautoria com os autores influentes de B;
- (iii) o número mais próximo de B, representa a quantidade de trabalhos que os pesquisadores de B têm em coautoria com um dos autores influentes de A.

Por exemplo, o número 2, mais próximo de NEC, na aresta NEC-DUKE, significa que há dois trabalhos em que pesquisadores da NEC colaboraram com um dos três pesquisadores influentes da DUKE. O autolago em DUKE e NTHU representa a cooperação entre os autores influentes da mesma afiliação. No caso da DUKE, por exemplo, os autores influentes têm coautoria em dois trabalhos.

Adicionalmente, a Figura 13 mostra a distribuição da quantidade de pesquisadores em SRE dentro de cada instituição, sendo que, cada círculo representa uma instituição

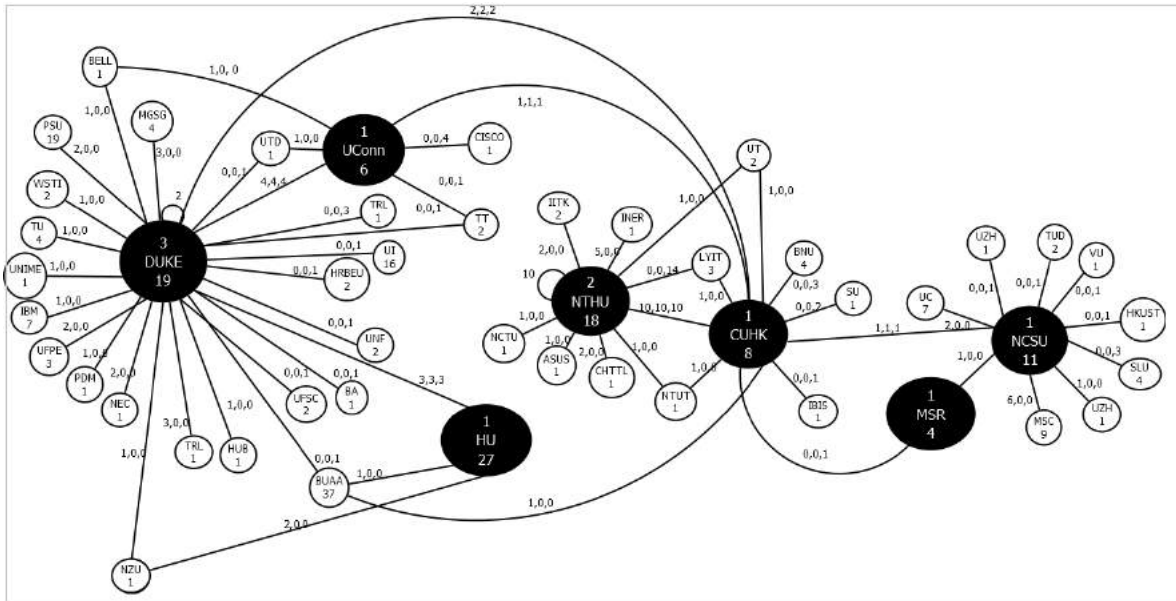


Figura 12 – Rede de colaboração entre os pesquisadores mais citados.

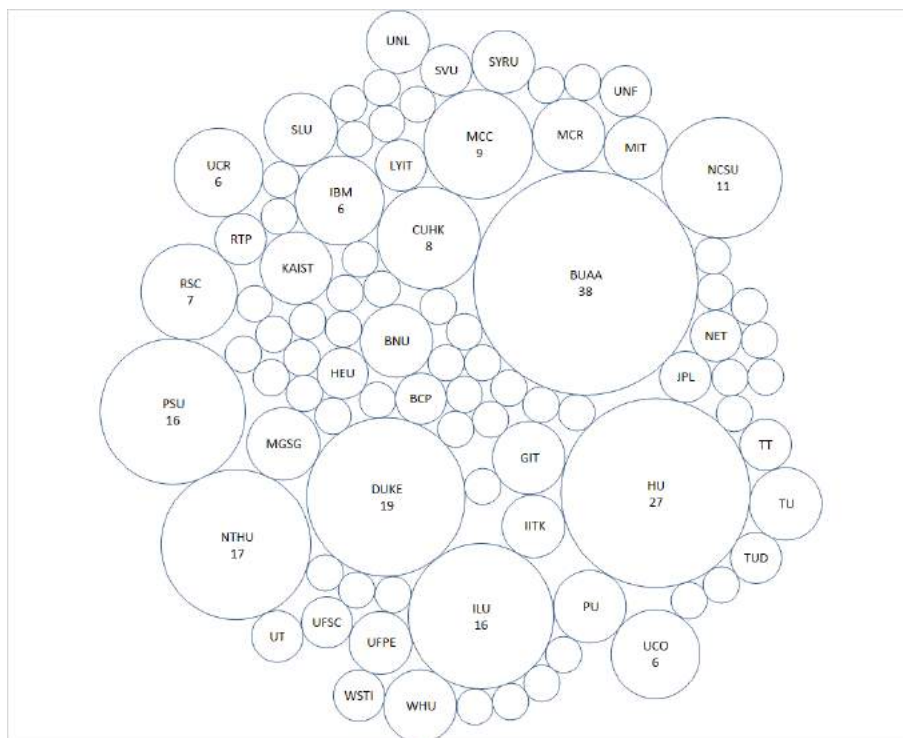


Figura 13 – Distribuição dos pesquisadores em SRE por instituição.

e seu tamanho depende do número de autores em SRE que ela possui. Quanto mais autores, maior o círculo. A Figura 13 considera todas as instituições presentes na amostra, enquanto a Figura 12 representa somente as que possuem algum autor influente.

4.6 Q6: Quais são os tópicos de pesquisa emergentes em SRE? Quais os tópicos de pesquisa saturados?

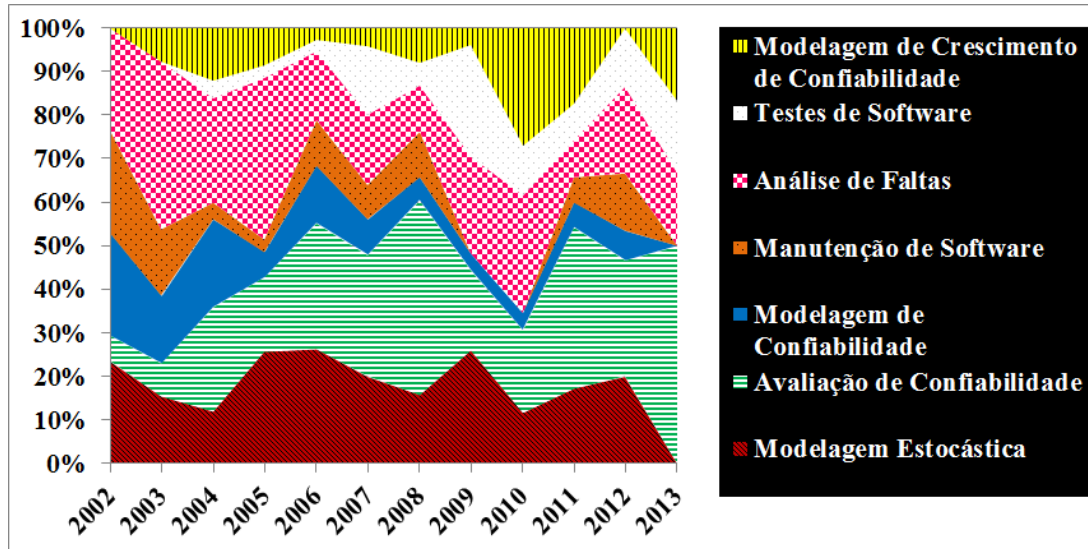


Figura 14 – Publicações anuais por agrupamento.

O gráfico da Figura 14 mostra o volume de publicações anuais por tópico de pesquisa. Considera-se tópicos emergentes aqueles que demonstram crescimento no número de publicações. Em contrapartida, os tópicos saturados são aqueles em que o número de pesquisas diminuiu. É possível perceber que o número de artigos em “Avaliação de Confiabilidade” teve um significativo progresso desde 2002, especialmente durante os últimos três anos da amostra. Um ligeiro incremento no número de artigos pode ser observado em “Testes de Software”. Por outro lado, a quantidade de artigos sobre “Modelagem de Confiabilidade” e “Análise de Falhas” têm reduzido gradualmente. Adicionalmente, a quantidade de publicações em “Modelagem Estocástica” parece ser a mais consistente nos últimos anos.

Uma análise dos resultados encontrados será apresentada no próximo capítulo.

Mapeamento e Análise dos Resultados

Este capítulo se destina a mapear e discutir os resultados encontrados com esta pesquisa. As seis questões de pesquisa foram combinadas a fim de estabelecer mapeamentos que possam fornecer uma visão geral sobre SRE. Esses mapeamentos permitem obter mais informações sobre como os resultados de cada questão estão relacionadas uns com os outros e quais as possíveis lacunas existentes na área. As seções foram divididas de forma a agregar os resultados mapeados, a saber, Tópicos de pesquisa em SRE, Tipos de pesquisa, Instituições dos autores influentes e produtivos e Cooperações entre instituições.

5.1 Tópicos de pesquisa em SRE

Os sete tópicos de pesquisa foram mapeados conforme os países de origem das publicações. O objetivo é saber em qual região tem sido discutido cada tópico. As Figuras 15 e 16 apresentam a concentração de cada um dos tópicos pelos países da afiliação do primeiro autor de cada artigo. A Figura 15 mostra a distribuição dos tópicos “Análise de Falhas”, “Avaliação de Confiabilidade” e “Manutenção de Software”. A Figura 16 apresenta a distribuição dos tópicos “Modelagem Estocástica”, “Modelagem de Confiabilidade”, “Modelagem de Crescimento de Confiabilidade” e “Testes de Software”. A descrição dessas duas figuras são idênticas. Os círculos representam a quantidade de estudos desenvolvidos na região. Foram rotulados os círculos que representam a região com o mínimo e o máximo de publicações para cada tópico. Anotações para ajudar a localizar os países que contém algum tópico são fornecidas na primeira vez que cada país é destacado na imagem. Por exemplo, para o agrupamento “Análise de Falhas” a região que pesquisa sobre este tópico e que possui menor número de artigos na amostra está localizada nos Estados Unidos (USA). Coincidentemente, a região que possui mais publicações dentro de “Análise de Falhas” também está nos Estados Unidos.

Pode-se perceber que, no geral, os Estados Unidos (USA) e o Taiwan (TWN) concentram grande parte dos artigos, incluindo todos os tópicos. Já em outros países, tais como, Índia (IND), Alemanha (DEU), Itália (ITA) e Brasil (BRA), nem todos os tópicos de

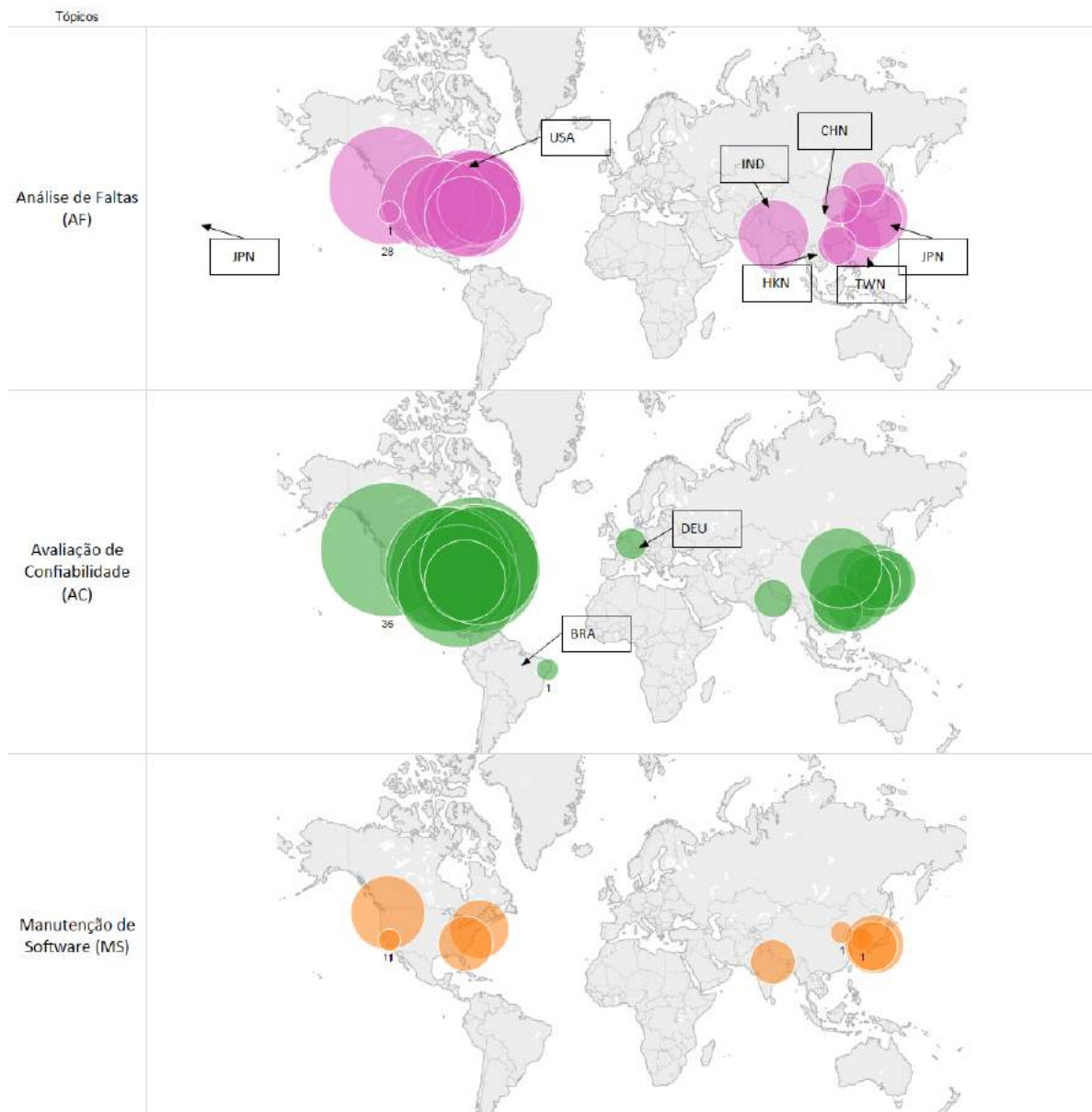


Figura 15 – Distribuição dos tópicos AF, AC e MS entre os países.

pesquisa foram encontrados. Na Índia tem-se trabalhado com “Análise de Falhas”, “Avaliação de Confiabilidade”, “Manutenção de Software” e “Modelagem Estocástica”. Na Alemanha encontra-se somente a presença de trabalhos em “Avaliação de Confiabilidade” e em “Testes de Software”. Já na Itália só aparece trabalhos em “Testes de Software”, e no Brasil, somente em “Avaliação de Confiabilidade”.

Os tópicos de pesquisa identificados no conjunto de artigos representam diferentes assuntos dentro da SRE. Esses assuntos podem não ser exclusivos de apenas um tópico. É possível observar que os assuntos de alguns tópicos permeiam em outros. A Tabela 11 ajuda a perceber essa característica. Os assuntos tratados nos artigos são identificados pela presença dos KGs em cada um dos artigos (ver seção 3.2.2).

A Tabela 11 mostra a porcentagem que cada KG representa em cada tópico de pes-

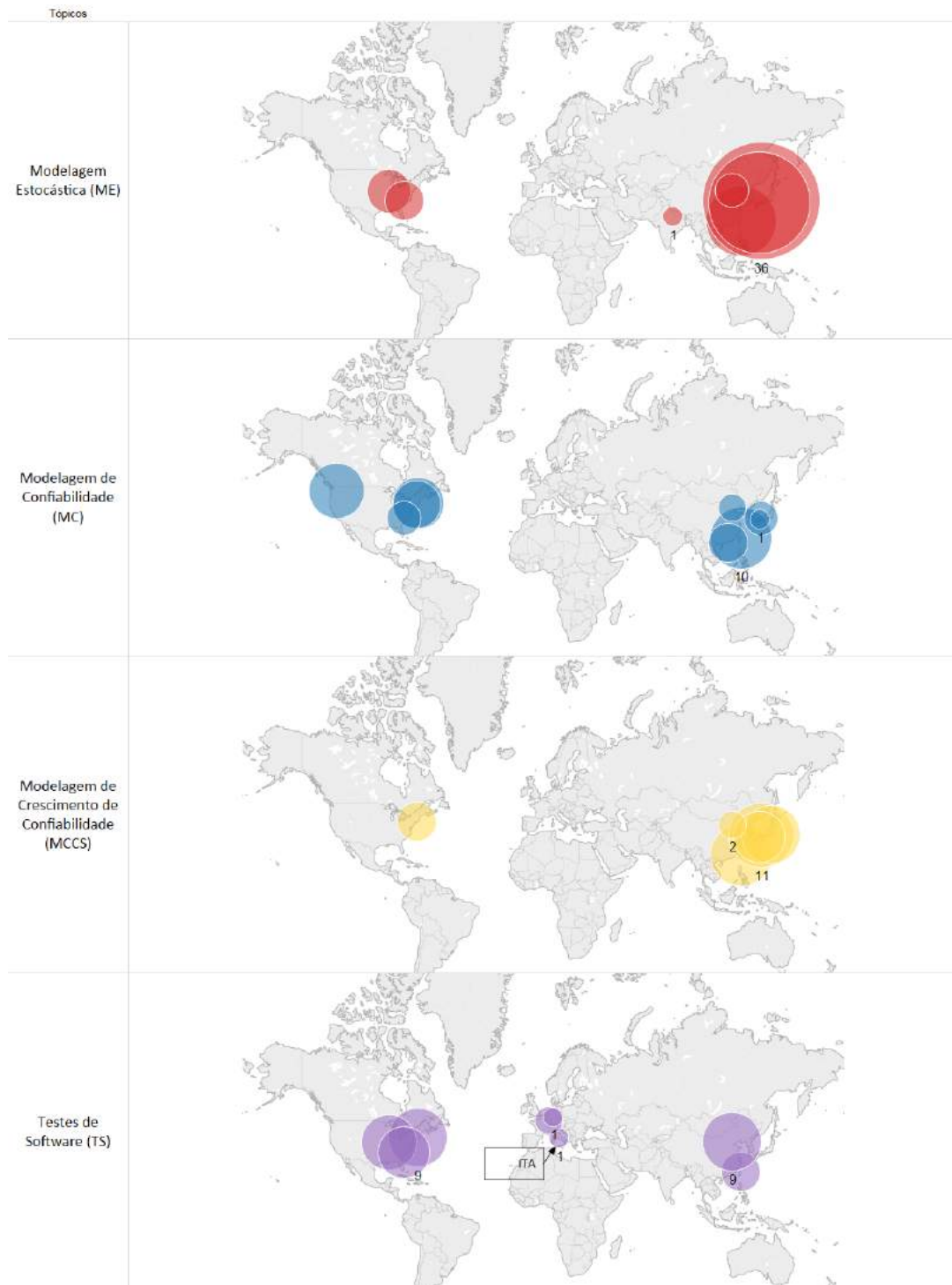


Figura 16 – Distribuição dos tópicos ME, MC, MCCS e TS entre os países.

quisa. As linhas representam os tópicos e as colunas cada um dos 40 KGs. O cálculo é feito pela razão entre o número de vezes que cada KG aparece em cada agrupamento e o número de artigos presentes naquele agrupamento. Em outras palavras, essa razão é dada por n_{KGi}/n_{Aj} , onde n_{KGi} refere-se ao número de artigos que contém alguma palavra do KG_i dentro do agrupamento j . n_{Aj} corresponde ao número de artigos que compõe o agrupamento j . Por exemplo, o agrupamento que deu origem ao tópico ME possui 59

artigos e dentre eles 4 têm palavras-chave de KG2. Sendo assim, a porcentagem de KG2 para o tópico ME será de 7%. As cores ajudam a identificar onde cada KG é mais presente (preto) ou mais ausente (cinza claro).

Tabela 11 – Relação do número de cada KG para cada tópico de pesquisa

	KG1	KG2	KG3	KG4	KG5	KG6	KG7	KG8	KG9	KG10	KG11	KG12	KG13	KG14	KG15	KG16	KG17	KG18	KG19	KG20
ME	0,00	0,07	0,03	0,03	0,00	0,08	0,00	0,34	0,02	0,00	0,07	0,02	0,07	0,00	0,92	0,02	0,02	0,00	0,51	0,07
AC	0,09	0,13	0,03	0,00	0,05	0,14	0,12	0,12	0,00	0,25	0,03	0,09	0,04	0,04	0,01	0,06	0,00	0,01	0,18	0,03
MC	0,08	0,50	0,04	0,00	0,08	0,35	0,04	0,73	0,00	0,00	0,00	0,08	0,00	0,12	0,12	0,12	0,04	0,00	0,88	0,08
MS	0,08	0,13	0,00	0,00	0,00	0,00	0,38	0,00	0,00	0,04	0,00	0,25	0,00	0,00	0,08	0,00	0,00	0,08	0,08	0,04
AF	0,02	0,32	0,03	0,03	0,02	0,05	0,13	0,55	0,00	0,02	0,00	0,37	0,00	0,03	0,05	0,00	0,00	0,03	0,08	0,00
TS	0,04	0,28	0,00	0,00	0,00	0,00	0,12	0,00	0,00	0,00	0,00	0,04	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,00
MCCS	0,00	0,00	0,00	0,00	0,00	0,07	0,00	0,22	0,00	0,00	0,07	0,15	0,04	0,00	0,41	0,00	0,00	0,00	0,48	0,00

	KG21	KG22	KG23	KG24	KG25	KG26	KG27	KG28	KG29	KG30	KG31	KG32	KG33	KG34	KG35	KG36	KG37	KG38	KG39	KG40
ME	0,02	0,00	0,00	0,20	0,36	0,00	0,03	0,00	0,08	0,00	0,02	0,00	0,03	0,03	0,05	0,02	0,00	0,02	0,10	0,00
AC	0,00	0,00	0,03	0,03	0,19	0,03	0,00	0,03	0,03	0,13	0,06	0,01	0,00	0,03	0,16	0,21	0,01	0,12	0,18	0,01
MC	0,04	0,04	0,00	0,04	0,77	0,00	0,12	0,00	0,08	0,00	0,12	0,04	0,00	0,04	0,00	0,00	0,00	0,00	0,00	0,19
MS	0,00	0,00	0,92	0,04	0,00	0,00	0,00	0,00	0,04	0,13	0,00	0,00	0,17	0,04	0,17	0,00	0,00	0,00	0,00	0,00
AF	0,00	0,00	0,03	0,03	0,00	0,02	0,00	0,02	0,00	0,02	0,24	0,13	0,00	0,29	0,00	0,05	0,13	0,10	0,00	0,02
TS	0,08	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,20	0,20	0,12	0,00
MCCS	0,00	0,11	0,04	0,07	0,07	0,00	0,07	0,00	0,96	0,00	0,04	0,00	0,00	0,04	0,00	0,00	0,00	0,07	0,11	0,00

Alguns tópicos, apesar de distintos, possuem uma correlação entre si. Essa correlação pode ser observada quando alguns assuntos, expressos pelos diferentes KGs, não são exclusivos de apenas um tópico. Isso pode ser visto nitidamente em relação ao tópico “Testes de Software” (TS). O agrupamento recebeu esse nome porque 100% dos artigos que o compõem contém as palavras-chave presentes no KG19, que é relativo a Testes de Software (ver Apêndice B). Porém, os tópicos “Modelagem de Confiabilidade” (MC), “Modelagem Estocástica” (ME) e “Modelagem de Crescimento de Confiabilidade” (MCCS) também possuem uma presença significativa desse KG. Isso se dá porque é através dos testes que dados são obtidos para a criação desses modelos, e ainda, grande parte dos modelos existentes se destinam a serem utilizados durante a fase de testes e *debugging*, validação ou na fase operacional (RAMAMOORTHY; BASTANI, 1982).

5.2 Tipos de pesquisa

Um mapeamento foi feito levando-se em consideração os tipos de pesquisa e o período de publicação em cada tópico. Através do cruzamento desses dados é possível perceber a distribuição de cada tipo de pesquisa através dos tópicos, bem como a contribuição de cada tópico em cada ano considerado. A Figura 17 mostra os resultados obtidos a partir das questões Q1 (Tópicos de Pesquisa) e Q2 (Tipos de Pesquisa) e adicionalmente integra o número de publicações por ano de cada tópico. No eixo y foram dispostos as siglas referentes aos tópicos de pesquisa. No eixo x , à direita está o período pesquisado (2002 a 2013) e à esquerda os tipos de pesquisa abordados. Os círculos mostram a quantidade

de estudos que se encaixa no cruzamento dos resultados dessas categorias. Por exemplo, o tópico ME possui 47 estudos teóricos, 4 empíricos e 7 teórico-empíricos. Esse mesmo tópico possui 4 estudos publicados em 2002, 2 em 2003 e assim por diante, totalizando 19,3% dos artigos da amostra. Ainda, é mostrado na figura a porcentagem que cada abordagem e cada ano representa no conjunto de artigos.

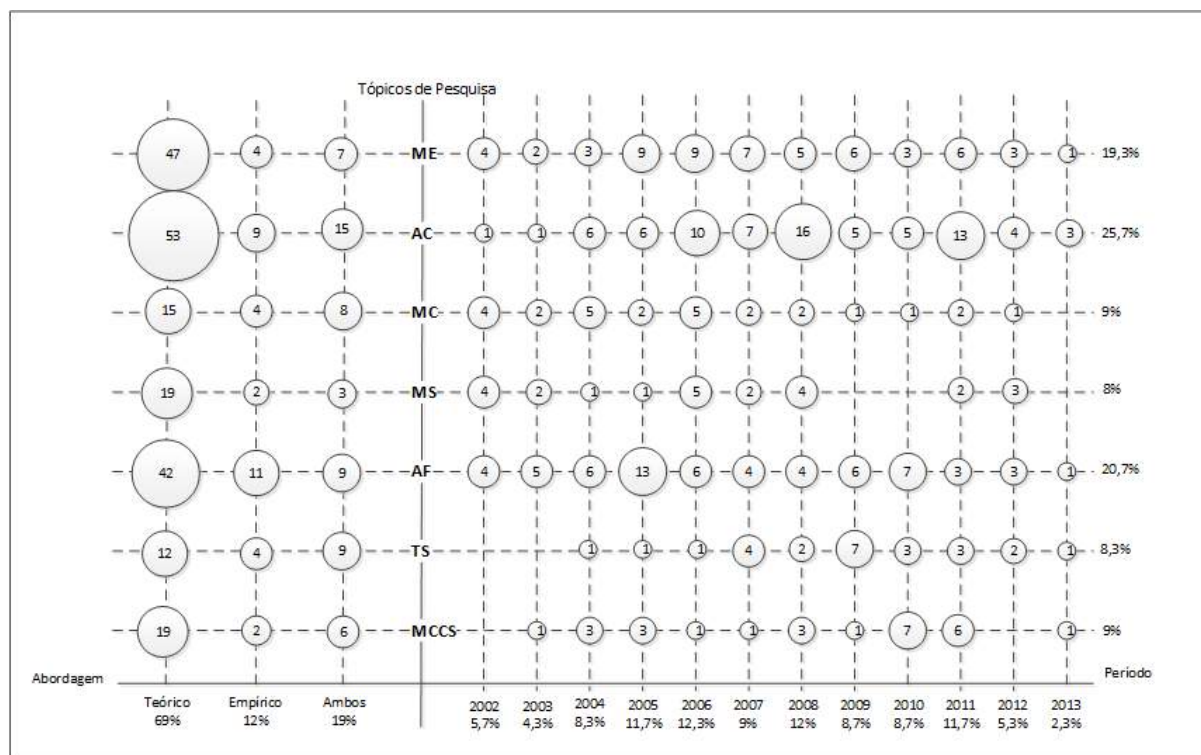


Figura 17 – Mapeamento dos tópicos de pesquisa, abordagens e período pesquisado.

Observa-se que trabalhos puramente teóricos são predominantes, 69%. Porém observa-se também que, gradativamente, o tipo teórico-empírico (ambos) tem ganhado espaço (ver Figura 11 na Seção 4.2). Trabalhos empíricos e teórico-empíricos, juntos, representam 31% do conjunto de dados analisado, dos quais somente 12% são puramente empíricos. Sugere-se a hipótese de que este pequeno número de estudos empíricos seja devido a duas razões principais:

- (i) dados experimentais em SRE são muito limitados;
- (ii) produzir dados de confiabilidade de software por meio de experimentos tipicamente requer muito tempo, que, por vezes, não é atrativo para o regime de publicação científica em vigor atualmente.

Essas questões poderiam ajudar a explicar o pequeno número de trabalhos empíricos ao longo dos últimos anos, especialmente em campos tais como MS e ME. Em contrapartida, os tópicos AF e AC se apresentaram como os que mais possuem trabalhos empíricos e

teórico-empíricos. Pode-se observar ainda que os anos que são marcados por um número maior de publicações (2005, 2006, 2008 e 2011) coincidem com os anos em que um desses dois tópicos tiveram mais publicações. Esses dois tópicos têm em comum a presença de KGs relacionados a detecção, minimização, correção e tolerância a faltas, ainda que estes sejam temas mais recorrentes em AF. Isso pode significar que os tópicos que têm fortalecido a pesquisa empírica e teórico-empírica tem sido principalmente esses dois, AF e AC. Adicionalmente, pode-se esperar que no futuro, outros tópicos que também tenham relação com Análise de Faltas venham a explorar mais a pesquisa empírico e teórico-empírico, tais como ME, MC e MCCS.

5.3 Instituições dos autores influentes e produtivos

Foram mapeadas nas Figuras 18 e 19 o número de publicações e de citações, respectivamente, para as instituições de cada país. Em cada estudo foi investigado seu número de citações e a afiliação do primeiro autor. Nesse sentido, as figuras são divididas pelos países. Dentro de cada país é calculado o número de estudos de cada instituição (Figura 18) e o número total de citações que os artigos de cada instituição recebeu (Figura 19). Os quartis superiores representam as instituições com maior número de publicações/citações do respectivo país. Os quartis inferiores, as instituições com menor número de publicações/citações. Por exemplo, no Japão (JPN) a instituição com o menor número de publicações é a NEC e a que possui maior número de publicações é HU (Figura 18). Duas instituições possuem número de publicações intermediários: TU e BCP. O mesmo acontece para o número de citações das instituições do Japão (Figura 19). Algumas instituições possuem *outliers*, estes são representados por pontos fora dos boxes. As instituições que se encontram nos quartis superiores ou como *outliers* são as que mais se destacam em SRE dentro de cada país.

Nas Seções 4.3 e 4.4 foram apresentados o *ranking* dos autores mais produtivos e dos autores mais influentes, sendo estes, respectivamente, os autores que mais publicam e os que têm artigos mais citados. A Tabela 12 é uma mesclagem das Tabelas 7 e 8, que apresentam, respectivamente, a origem dos autores mais produtivos e a origem dos autores mais influentes. Além de mesclá-las, foram adicionadas duas colunas às já existentes nas tabelas originais, as colunas RP e RI. A coluna RP (*Ranking* Produtivos), refere-se à colocação do respectivo autor na Tabela 7, se este existir nela. Da mesma forma, a coluna RI (*Ranking* Influentes), refere-se à colocação do respectivo autor na Tabela 8, se ele existir nesta tabela.

Percebe-se que valores altos de NTP nem sempre estão associados a altos valores de média de citações. A explicação para esse fato pode estar relacionada à “idade” de publicação de alguns artigos (quanto maior a idade de um artigo, maior é a possibilidade de esse artigo ter um maior número de citações). Percebe-se ainda, que alguns dos dez

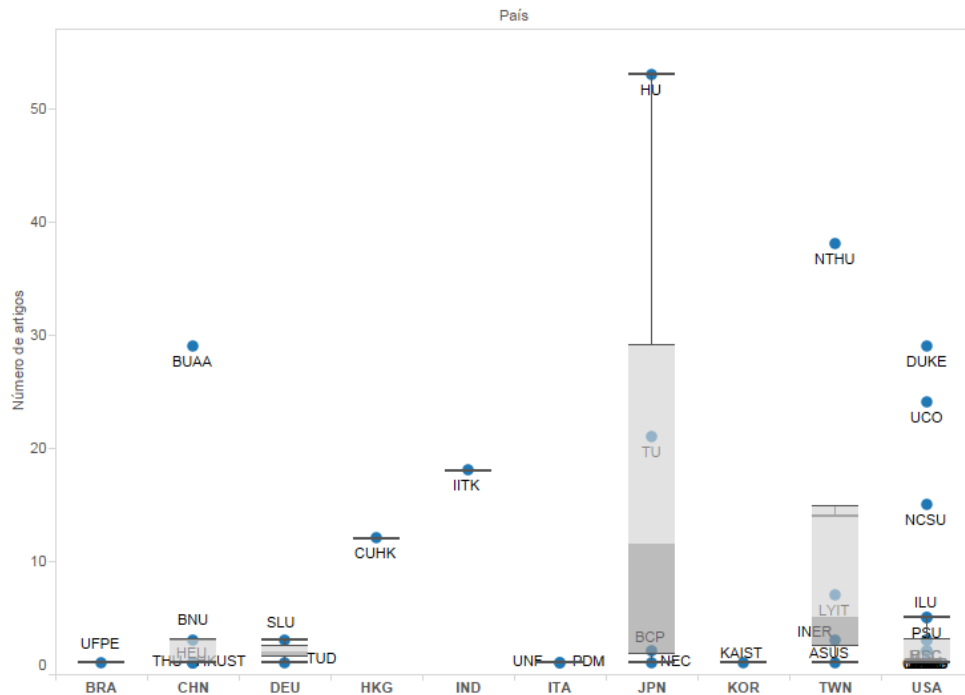


Figura 18 – Box-plot do número de publicações das instituições de cada país

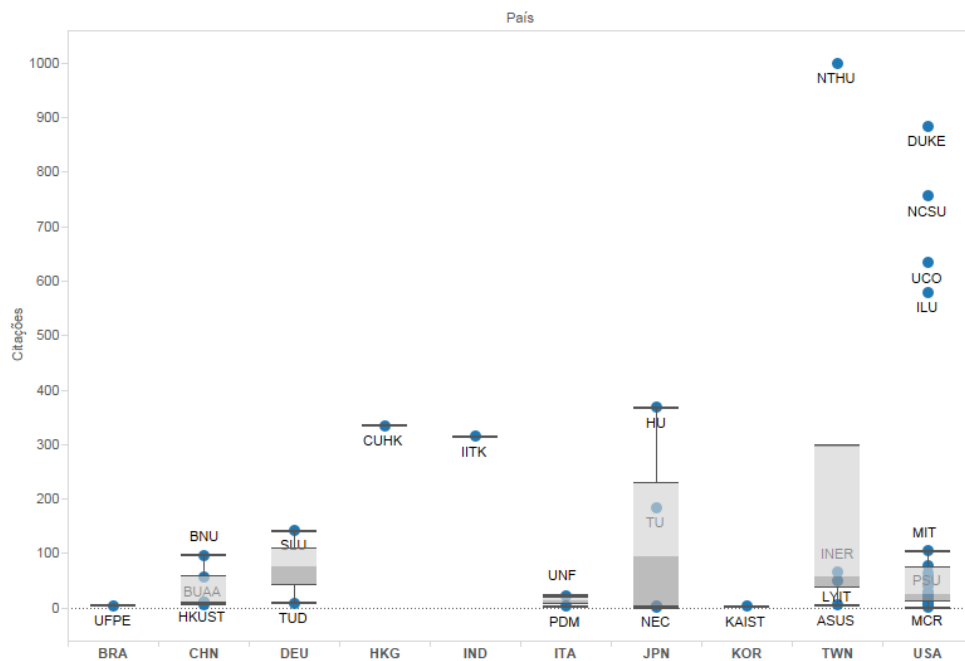


Figura 19 – Box-plot do número de citações das instituições de cada país

autores mais produtivos não estão entre os dez autores mais influentes. Isso mostra que alguns dos autores têm publicações importantes na área mesmo não tendo um volume grande de publicações. Adicionalmente, pode-se concluir que, considerando a quantidade de trabalhos publicados e citações, a Duke University, a Hiroshima University e a National Tsing Hua University têm tido significativa importância nesta área na última década.

Tabela 12 – Afiliação dos autores mais produtivos e influentes.

Afiliação	País	NTP	RP	NC	RI
Hiroshima University	JPN	56	1	381	9
Duke University	USA	49	2	1668	1
National Tsing Hua University	TWN	48	3	1057	2
Hiroshima University	JPN	34	4	-	-
Chinese University of Hong Kong	HKG	27	5	916	3
University of Connecticut	USA	25	6	643	5
Tottori University	JPN	25	7	-	-
Indian Inst. of Tech. of Kanpur	IND	18	8	-	-
North Carolina State University	USA	18	9	843	4
Beihang University	CHN	16	10	-	-
Microsoft Research	USA	4	-	436	6
Duke University	USA	5	-	436	7
National Tsing Hua University	TWN	10	-	423	8
Duke University	USA	3	-	297	10

Diante da perspectiva apresentada e considerando ainda, a Tabela 9 (Seção 4.4), que mostra os três trabalhos mais citados por agrupamento, observa-se que as pesquisas em SRE nos Estados Unidos e Taiwan têm tido importante influência nesta área na última década, considerando ambos os números de trabalhos publicados e citações. Esses dados corroboram os resultados apresentados nesta seção, já que os trabalhos mais citados de cada agrupamento vêm de ambos os países. Sendo que, Estados Unidos se relaciona mais com os grupos ME, AC, MC, MS, AF e TS e Taiwan com MCCS. Adicionalmente, nota-se que um autor da Duke University tem três trabalhos entre os mais citados.

5.4 Cooperações entre instituições

Na Seção 4.5 foram apresentados achados importantes relacionados à rede de colaboração dos dez autores mais citados (Figura 12). Observa-se que a Duke University (DUKE) é o centro de pesquisa com mais cooperação em SRE, entre as instituições analisadas. Nesse estudo, Duke tem 25 cooperações diretas diferentes que se propagam para vários países, como Itália, Japão, Canadá, Brasil e China, além dos Estados Unidos onde reside. Ela também tem colaboração direta com três centros de pesquisa influentes (University of Connecticut - UConn, Hiroshima University – HU e Chinese University of Hong Kong – CUHK). Em contrapartida, existem instituições que possuem um número expressivo de pesquisadores (maior do que 25) em SRE, porém têm um número limitado (menor do que três) de cooperações (ex. BUAA e HU). Além das cooperações, nota-se também que centros de pesquisa com número considerável de pesquisadores (ex. BUAA, ILU e PSU) não estão entre os mais influentes (ver Figura 13). As evidências deste estudo sugerem que

a cooperação entre centros de pesquisa parece ser mais importante do que a quantidade de pesquisadores afiliados.

Entre os centros que possuem pesquisadores influentes (DUKE, UConn, HU, NTHU, CUKH, HU, MSR, NCSU), HU é aquele que possui a menor relação entre o número de pesquisadores influentes (PI) por número de pesquisadores afiliados (PA), a saber, $1/27$. Observa-se que HU possui uma rede de colaboração pequena (apenas três), quando comparada contra os demais centros de pesquisa influentes. Com exceção do MSR, os demais centros de pesquisa influentes que possuem uma relação PI/PA melhor que HU, realizam um número significativamente maior de colaboração externa. Outra indicação da importância da colaboração externa é CUHK, que se apresenta mais influente do que centros com grande quantidade de pesquisadores afiliados, tal como BUAA. Logo, pode-se concluir que, em SRE, o fator “colaboração externa” é o que mais se destaca. Parece existir uma forte correlação entre a quantidade e qualidade destas conexões, sendo mais importante que outros fatores como a colaboração interna, número de pesquisadores, ou mesmo, o número de trabalhos publicados.

Conclusão

Este trabalho apresentou um mapeamento sistemático (PETERSEN et al., 2008) da área de Engenharia de Confiabilidade de Software. Esse mapeamento se baseou em 300 estudos primários publicados no período de 2002 até 2013. Foram estabelecidas seis questões de pesquisa, responsáveis por direcionar o estudo. As questões de pesquisa buscaram mapear os artigos de acordo os tópicos da área, tipos de pesquisa, origem dos autores, instituição dos autores mais produtivos e mais citados, entre outros. Para isso, criou-se critérios para recuperar e selecionar os artigos que pudessem responder à questões definidas.

Os principais resultados mostraram que a pesquisa em SRE tem uma grande concentração nos Estados Unidos e no Taiwan, sendo que esses países abragem fortemente todos os tópicos. Baseando-se nos números de pesquisadores, trabalhos, citações, quantidade e qualidade de conexões de colaboração, pode-se concluir que, na comunidade de pesquisa em confiabilidade de software, o fator “colaboração externa” é o que mais se destaca. Parece existir uma forte correlação entre a quantidade e qualidade destas conexões, sendo mais importante que outros fatores como a colaboração interna, número de pesquisadores, ou mesmo, o número de trabalhos publicados.

Observou-se uma tendência nos últimos anos, em que pesquisadores em SRE estão migrando gradualmente de abordagens puramente teóricas para abordagens teórico-empíricas. Nota-se ainda que os tópicos “Avaliação de Confiabilidade” e “Análise de Falhas” podem ser os responsáveis pelo ligeiro crescimento da pesquisa empírica e teórico-empírica. Dessa forma, acredita-se que no futuro outros tópicos que também tenham relação com análise de falhas venham a explorar mais a pesquisa empírica e teórico-empírica.

6.1 Principais Contribuições

Além dos resultados e conclusões obtidas dos estudos em si, podem ser destacadas algumas contribuições específicas desta pesquisa. Estas são listadas a seguir:

- ❑ Uma visão geral da área de confiabilidade de software.
- ❑ Uma caracterização de todos os artigos selecionados em termos dos atributos considerados nesse mapeamento (ex. tipos e tópicos de pesquisa, etc). Essa classificação pode estar disponível no site¹ do grupo de pesquisa.
- ❑ Uma sugestão de método para o “esquema de classificação”, baseado em grupos de palavras-chave e análise de agrupamento. Esse método pode ser usado em futuros mapeamentos sistemáticos nesta e em outras áreas.
- ❑ Sugestões de nichos de pesquisa que podem ser explorados por revisões sistemáticas no futuro.
- ❑ Algumas recomendações sobre futuras direções de pesquisa, tais como o possível crescimento do número de trabalhos empíricos e teórico-empíricos nos próximos anos.

Dessa forma, esse estudo pode vir a contribuir com um olhar panorâmico sobre SRE, auxiliando os mais diversos pesquisadores desta área a incrementarem suas pesquisas.

6.2 Ameaças à Validade

Na busca pelos artigos foram considerados os autores com mais de 10 publicações no período. Com isso, correu-se o risco de algum artigo importante de um autor menos ativo ter sido perdido. Essa foi uma decisão para tornar o tamanho da amostra viável para análise por uma única pessoa.

Outra ameaça é em relação à formação dos KGs. Por mais que se tenha validado os agrupamentos formados, eles podem ainda não ser o ideal. As palavras-chave consideradas para formação dos KGs foram extraídas de forma manual. A extração manual leva em consideração a análise humana, isso pode ter contribuído para que alguma palavra significativa não fosse considerada. O ideal nesse caso é que fosse utilizada uma ferramenta para automatizar e aumentar a precisão do processo. Essa ferramenta está sendo desenvolvida e poderá ser utilizada em trabalhos futuros.

6.3 Trabalhos Futuros

A partir dos resultados e experiências obtidas com essa dissertação, alguns trabalhos futuros podem ser sugeridos. Alguns deles já estão em andamento. A princípio, está sendo desenvolvida uma revisão sistemática onde questões de pesquisa mais específicas são exploradas. Por exemplo, “Quais as formas de obtenção de dados para a pesquisa

¹ <<http://hpdc.facom.ufu.br/sre>>

em SRE?”, “Eventos de falta/falha são considerados dependentes?”, “Quais os métodos de validação são utilizados?”, entre outras. Além disso, uma ferramenta está em fase de testes e tem como objetivo ajudar a minimizar as ameaças verificadas neste mapeamento. Essa ferramenta será utilizada para extrair os dados e as palavras-chaves dos artigos automaticamente. Dessa forma, tem-se condição de considerar todas as palavras-chave nos KGs e associá-los automaticamente aos artigos, conferindo mais opções de palavras para formarem os agrupamentos.

Sugere-se também como trabalhos futuros um estudo de utilização de técnicas de mineração de dados e classificadores para a realização de revisões sistemáticas. A abordagem geralmente utilizada nesses trabalhos, e a que foi utilizada neste, depende de análise humana. Explorar técnicas de mineração em texto poderá ampliar a capacidade de precisão dos resultados e permitir que um número maior de estudos seja utilizado.

Em trabalhos futuros pretende-se investigar as referências dos trabalhos selecionados em busca de artigos que sejam relevantes e que por algum motivo não estejam na amostra. Esses artigos podem ser, inclusive, estudos mais antigos. Se um artigo é citado pelos outros artigos que estão na amostra, isso pode significar que seja importante considerar aquele artigo. Dessa forma é possível garantir uma maior qualidade do estudo.

Apesar da crescente procura por mapeamentos e revisões sistemáticas em engenharia de software, nota-se poucas contribuições em relação a ferramentas para auxiliar no processo. O volume de dados a se trabalhar por vezes é muito grande, tornando sua execução muito dispendiosa, como foi o caso dessa pesquisa. Sendo assim, ferramentas para facilitar esse processo são contribuições valiosas. Nesse sentido, como já foi falado, será implementado uma ferramenta para apoiar o método de classificação proposto nesse mapeamento. Além disso, será modelado um banco de dados para catalogar os dados dos estudos. Além da ferramenta a ser desenvolvida, outras já existentes também podem ser utilizadas em conjunto para gerar relatórios.

Por fim, nota-se que algumas áreas dentro da confiabilidade de software necessitam de uma maior investigação, podendo ser inclusive, alvos de futuras revisões sistemáticas. “Testes de software”, por exemplo, pode ser considerada uma área essencial dentro de SRE e tem tido um recente crescimento, investigar a causa desse crescimento e o que esta área tem de promissora pode ser uma contribuição significativa. Outra proposta de revisão sistemática se aplica à investigação da pesquisa empírica em SRE. Nota-se que a pesquisa teórica está relativamente madura, a empírica e teórico-empírica, porém, estão começando a crescer. Investigar como tem sido tratadas e relatadas essas pesquisas pode auxiliar ainda mais o seu crescimento. Uma das contribuições, em menor nível de complexidade, que esta revisão sugerida pode fornecer é relatar, por exemplo, os repositórios públicos de dados e divulgação de dados empíricos.

Tabela 13 – Produção Bibliográfica.

Título	Conferência	Formato	Citação
A survey on research in software reliability engineering in the last decade	ACM SAC 2014	Pôster	(XAVIER et al., 2014b)
Engenharia de Confiabilidade de Software: Um mapeamento sistemático da última década	SBESC 2014	Artigo Completo	(XAVIER et al., 2014a)
Understanding the popularity of reporters and assignees in the github	SEKE 2014	Artigo Completo	(XAVIER et al., 2014c)

6.4 Contribuições em Produção Bibliográfica

Durante o desenvolvimento deste trabalho, dois artigos foram publicados com os resultados parciais obtidos. A Tabela 13 apresenta esses artigos com seus respectivos dados, a saber, título, o acrônimo da conferência em que foi publicado, o formato da publicação e a citação do mesmo. O terceiro artigo da tabela foi publicado como atividade ortogonal a esta pesquisa, porém foi desenvolvido dentro deste programa de mestrado.

Referências

- AMARI, S. V.; MCLAUGHLIN, L. Optimal design of a condition-based maintenance model. In: IEEE. **Reliability and Maintainability, 2004 Annual Symposium-RAMS**. [S.l.], 2004. p. 528–533.
- AVIZIENIS, A. et al. **Fundamental concepts of dependability**. [S.l.]: University of Newcastle upon Tyne, Computing Science, 2001.
- BARBETTA, P. A.; REIS, M. M.; BORNIA, A. C. **Estatística para Cursos de Engenharia e Informática**. [S.l.]: Editora Atlas, 2010. v. 3.
- BEELMANN, A.; PETTICREW, M.; ROBERTS, H. Systematic reviews in the social sciences. a practical guide. **European Psychologist**, Hogrefe & Huber, v. 11, n. 3, p. 244–245, 2006.
- BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. **Journal of Systems and Software**, Elsevier, v. 80, n. 4, p. 571–583, 2007.
- BUSSAB, W. d. O. **Introdução à análise de agrupamentos**. [S.l.]: ABE, 1990. 105 p., 1990.
- CHEN, L.; AVIZIENIS, A. On the implementation of n-version programming for software fault tolerance during program execution. In: **International Computer Software and Applications Conference (COMPSAC)**. [S.l.: s.n.], 1977.
- DEMO, P. **Pesquisa e construção de conhecimento: metodologia científica no caminho de Habermas**. [S.l.]: Tempo Brasileiro, 1994.
- DHARMASENA, L. S.; ZEEPHONGSEKUL, P.; JAYASINGHE, C. L. Software reliability growth models based on local polynomial modeling with kernel smoothing. In: IEEE. **Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on**. [S.l.], 2011. p. 220–229.
- DINGSØYR, T. et al. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, Elsevier, v. 85, n. 6, p. 1213–1221, 2012.
- DOGAN, S.; BETIN-CAN, A.; GAROUSHI, V. Web application testing: A systematic literature review. **Journal of Systems and Software**, v. 91, p. 174–201, 2014.

- DYBA, T.; KITCHENHAM, B. A.; JORGENSEN, M. Evidence-based software engineering for practitioners. **Software, IEEE, IEEE**, v. 22, n. 1, p. 58–65, 2005.
- ENGSTRÖM, E.; RUNESON, P. Software product line testing—a systematic mapping study. **Information and Software Technology**, Elsevier, v. 53, n. 1, p. 2–13, 2011.
- EVERITT, B. S. **Cluster Analysis**, Edward Arnold. [S.l.]: Halsted Press, London, 1992.
- FERNANDEZ, A.; INSFRAN, E.; ABRAHÃO, S. Usability evaluation methods for the web: A systematic mapping study. **Information and Software Technology**, Elsevier, v. 53, n. 8, p. 789–817, 2011.
- FERREIRA, D. F. **Estatística multivariada**. 2. ed. [S.l.]: UFLA, 2011.
- GOKHALE, S. S.; LYU, M.-T. A simulation approach to structure-based software reliability analysis. **Software Engineering, IEEE Transactions on**, IEEE, v. 31, n. 8, p. 643–656, 2005.
- GROTTKE, M.; MATIAS, R.; TRIVEDI, K. S. The fundamentals of software aging. In: IEEE. **Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on**. [S.l.], 2008. p. 1–6.
- HUANG, C.-Y.; KUO, S.-Y. Analysis of incorporating logistic testing-effort function into software reliability modeling. **Reliability, IEEE Transactions on**, IEEE, v. 51, n. 3, p. 261–270, 2002.
- HUANG, C.-Y.; LIN, C.-T. Software reliability analysis by considering fault dependency and debugging time lag. **Reliability, IEEE Transactions on**, IEEE, v. 55, n. 3, p. 436–450, 2006.
- HUANG, C.-Y.; LYU, M. R.; KUO, S.-Y. A unified scheme of some nonhomogenous poisson process models for software reliability estimation. **Software Engineering, IEEE Transactions on**, IEEE, v. 29, n. 3, p. 261–269, 2003.
- IEEE/ANSI. Ieee standard glossary of software engineering terminology. In: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. [S.l.], 1990. p. 1–84.
- ISO, N. 9000-4. normas de gestão da qualidade e garantia da qualidade—parte 4: Guia para gestão do programa de dependabilidade. **Rio de Janeiro: ABNT**, 1994.
- KAPUR, P. et al. **Software Reliability Assessment with OR Applications**. [S.l.]: Springer, 2013. (Springer Series in Reliability Engineering). ISBN 9780857292049.
- KEUPP, M. M.; PALMIÉ, M.; GASSMANN, O. The strategic management of innovation: a systematic review and paths for future research. **International Journal of Management Reviews**, Wiley Online Library, v. 14, n. 4, p. 367–390, 2012.
- KHANDELWAL, A. **Software Reliability Modeling using Fault Tree Analysis and Stochastic Petri Nets**. Tese (Doutorado), 2013.
- KITCHENHAM, B. **Procedures for Performing Systematic eviews**. [S.l.], 2004. v. 33, n. 2004, 1–26 p.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering—a systematic literature review. **Information and software technology**, Elsevier, v. 51, n. 1, p. 7–15, 2009.

_____. **Guidelines for performing Systematic Literature Reviews in Software Engineering**. [S.l.], 2007.

KITCHENHAM, B. A.; DYBA, T.; JORGENSEN, M. Evidence-based software engineering. In: IEEE COMPUTER SOCIETY. **Proceedings of the 26th international conference on software engineering**. [S.l.], 2004. p. 273–281.

LEVESON, N.; TURNER, C. An investigation of the therac-25 accidents. **Computer**, v. 26, n. 7, p. 18–41, July 1993.

LI, L.; VAIDYANATHAN, K.; TRIVEDI, K. S. An approach for estimation of software aging in a web server. In: IEEE. **Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium on**. [S.l.], 2002. p. 91–100.

LI, Z. et al. Have things changed now?: an empirical study of bug characteristics in modern open source software. In: ACM. **Proceedings of the 1st workshop on Architectural and system support for improving software dependability**. [S.l.], 2006. p. 25–33.

LIN, C.-T.; HUANG, C.-Y. Enhancing and measuring the predictive capabilities of testing-effort dependent software reliability models. **Journal of Systems and Software**, Elsevier, v. 81, n. 6, p. 1025–1038, 2008.

LIONS, J.-L. et al. **Ariane 5 flight 501 failure**. [S.l.]: Technical report, European Space Agency, July 1996. Disponível em <http://www.es-rin.esa.it/htdocs/tidc/Press/Press96/ariane5rep.html>, 1996.

LO, J.-H.; HUANG, C.-Y. An integration of fault detection and correction processes in software reliability analysis. **Journal of Systems and Software**, Elsevier, v. 79, n. 9, p. 1312–1323, 2006.

LYU, M. R. **Handbook of software reliability engineering**. [S.l.]: IEEE computer society press CA, 1996. v. 222.

MARCONI, M. d. A.; LAKATOS, E. M. Fundamentos de metodologia científica. In: **Fundamentos de metodologia científica**. [S.l.]: Atlas, 2010.

MARSHALL, C.; BRERETON, P.; KITCHENHAM, B. Tools to support systematic literature reviews in software engineering: A mapping study. In: IEEE. **Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on**. [S.l.], 2013. p. 296–299.

MATIAS, R. et al. Accelerated degradation tests applied to software aging experiments. **Reliability, IEEE Transactions on**, IEEE, v. 59, n. 1, p. 102–114, 2010.

MILLER, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. **Psychological review**, American Psychological Association, v. 63, n. 2, p. 81, 1956.

- MUSA, J.; IANNINO, A.; OKUMOTO, K. **Software reliability: measurement, prediction, application**. [S.l.]: McGraw-Hill, 1987. (McGraw-Hill series in software engineering and technology). ISBN 9780070440937.
- NAGAPPAN, N.; BALL, T. Use of relative code churn measures to predict system defect density. In: IEEE. **Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on**. [S.l.], 2005. p. 284–292.
- NAGAPPAN, N.; MURPHY, B.; BASILI, V. The influence of organizational structure on software quality: an empirical case study. In: ACM. **Proceedings of the 30th international conference on Software engineering**. [S.l.], 2008. p. 521–530.
- NICOL, D. M.; SANDERS, W. H.; TRIVEDI, K. S. Model-based evaluation: from dependability to security. **Dependable and Secure Computing, IEEE Transactions on**, IEEE, v. 1, n. 1, p. 48–65, 2004.
- NULL, L.; LOBUR, J. **Princípios Básicos de Arquitetura e Organização de Computadores**. [S.l.]: Bookman, 2011. ISBN 9788577807666.
- O’CONNOR, P.; KLEYNER, A. **Practical reliability engineering**. [S.l.]: John Wiley & Sons, 2011.
- PETERSEN, K. et al. Systematic mapping studies in software engineering. In: SN. **12th International Conference on Evaluation and Assessment in Software Engineering**. [S.l.], 2008. v. 17, n. 1.
- PHAM, H. **System software reliability**. [S.l.]: Springer Science & Business Media, 2007.
- PIETRANTUONO, R.; RUSSO, S.; TRIVEDI, K. S. Software reliability and testing time allocation: An architecture-based approach. **Software Engineering, IEEE Transactions on**, IEEE, v. 36, n. 3, p. 323–337, 2010.
- RADJENOVIĆ, D. et al. Software fault prediction metrics: A systematic literature review. **Information and Software Technology**, Elsevier, v. 55, n. 8, p. 1397–1418, 2013.
- RAMAMOORTHY, C.; BASTANI, F. B. Software reliability - status and perspectives. **IEEE Trans. Software Eng.**, v. 8, n. 4, p. 354–371, 1982.
- RANDELL, B. System structure for software fault tolerance. In: ACM. **ACM SIGPLAN Notices**. [S.l.], 1975. v. 10, n. 6, p. 437–449.
- ROTHER, E. T. Revisão sistemática x revisão narrativa. **Acta Paulista de Enfermagem**, scielo, v. 20, p. v – vi, 06 2007. ISSN 0103-2100.
- SAFF, D.; ERNST, M. D. An experimental evaluation of continuous testing during development. In: ACM. **ACM SIGSOFT Software Engineering Notes**. [S.l.], 2004. v. 29, n. 4, p. 76–85.
- SINGHAL, A.; SINGHAL, A. A systematic review of software reliability studies. **Softw. Eng.: Inte. J**, v. 1, n. 1, 2011.

SOMMERVILLE, I. **Software Engineering**. [S.l.]: Pearson, 2011. (International Computer Science Series).

SU, Y.-S.; HUANG, C.-Y. Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models. **Journal of Systems and Software**, Elsevier, v. 80, n. 4, p. 606–615, 2007.

SULLIVAN, M.; CHILLAREGE, R. Software defects and their impact on system availability: A study of field failures in operating systems. In: **FTCS**. [S.l.: s.n.], 1991. p. 2–9.

VAIDYANATHAN, K.; TRIVEDI, K. S. A comprehensive model for software rejuvenation. **Dependable and Secure Computing, IEEE Transactions on**, IEEE, v. 2, n. 2, p. 124–137, 2005.

WARD JR, J. H. Hierarchical grouping to optimize an objective function. **Journal of the American Statistical Association**, Taylor & Francis, v. 58, n. 301, p. 236–244, 1963.

WAZLAWICK, R. **Metodologia de pesquisa para ciência da computação**. [S.l.]: Elsevier Brasil, 2009.

XAVIER, J. et al. Engenharia de confiabilidade de software: um mapeamento sistemático da última década. In: **Proceedings of the Brazilian Symposium on Computing Systems Engineering**. [S.l.: s.n.], 2014. (SBESC '14).

_____. A survey on research in software reliability engineering in the last decade. In: **Proceedings of the 29th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2014. (SAC '14), p. 1190–1191.

_____. Understanding the popularity of reporters and assignees in the github. In: **Proceedings of the Twenty-Sixth International Conference on Software Engineering Knowledge Engineering**. [S.l.: s.n.], 2014. (SEKE '14), p. 484–489.

XIE, Y. et al. Reliability-aware co-synthesis for embedded systems. **The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology**, Springer, v. 49, n. 1, p. 87–99, 2007.

YAU, S.; CHEUNG, R. Design of self-checking software. In: ACM. **ACM SIGPLAN Notices**. [S.l.], 1975. v. 10, n. 6, p. 450–455.

YIN, Z. et al. How do fixes become bugs? In: ACM. **Proceedings of the 19th ACM SIGSOFT symposium and the 13th European conference on Foundations of software engineering**. [S.l.], 2011. p. 26–36.

Apêndices

Estudos da Amostra

Esta Seção apresenta a lista dos 300 estudos que compõem a amostra deste Mapeamento Sistemático. A lista de estudos desta Seção está organizada em ordem alfabética pelo sobrenome do primeiro autor. Alguns desses estudos também fizeram parte da Seção Referências.

- [1] AMENTE, M. P., AND GOMALIS, S. S. Error bugs: life cycle and resolution analysis. In *Quality Software, 2005. QSOFT'05. 15th Ninth International Conference on* (2005). IEEE, pp. 336–341.
- [2] AL, J., AND LEE, M. The analysis and modeling for the input space of real-time embedded software. In *Reliability, Maintainability and Safety, 2000. FORMS 2000. 6th International Conference on* (2000). IEEE, pp. 774–777.
- [3] ANDER, J., DEMIRKURT, B., ASARI, S. V., AND KIRAN, L. Analysis of multi-state systems using multi-valued decision diagrams. In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual* (2009). IEEE, pp. 337–343.
- [4] ASARI, S., AND FORTIN, W. Bounds on optimal replacement time of age replacement policy. In *Reliability and Maintainability Symposium, 2003. Annual* (2003). IEEE, pp. 117–122.
- [5] ASARI, S., MICHAELIDES, L., AND VARDARAKI, H. Optimal cost-effective design of parallel systems subject to imperfect fault-coverage. In *Reliability and Maintainability Symposium, 2003. Annual* (2003). IEEE, pp. 23–24.
- [6] ASARI, S. V. Adherence to Gersoni rules to evaluate system-failure frequency. *Reliability, IEEE Transactions on* 31, 3 (2002), 375–379.

- [7] ASHRAE, S. V. Bounds on mbbf of systems subjected to periodic maintenance. *Reliability, IEEE Transactions on* 55, 3 (2000), 169–174.
- [8] ASHRAE, S. V., AND AKER, J. H. Reliability analysis of large fault trees using the vesely failure rate. In *Reliability and Maintainability, 2004 Annual Symposium-RAIIS* (2004-01), IEEE, pp. 201–206.
- [9] ASHRAE, S. V., AND DECKMAN, R. Reliability analysis of k-out-of-n load-sharing systems. In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual* (2009), IEEE, pp. 440–445.
- [10] ASHRAE, S. V., AND HEDDER, V. New allocation methods for repairable systems. In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual* (2009), IEEE, pp. 290–295.
- [11] ASHRAE, S. V., AND McLAUGHLIN, L. Optimal design of a condition-based maintenance model. In *Reliability and Maintainability, 2004 Annual Symposium-RAIIS* (2004-01), IEEE, pp. 528–533.
- [12] ASHRAE, S. V., McLAUGHLIN, L., AND PHAM, H. Cost-effective condition-based maintenance using markov decision processes. In *Reliability and Maintainability Symposium, 2006. RAMS 06. Annual* (2006), IEEE, pp. 361–369.
- [13] ASHRAE, S. V., NIXON, A., AND RAJCV, A. An efficient algorithm to analyse new imperfect fault coverage models. In *Reliability and Maintainability Symposium, 2007. RAMS 07. Annual* (2007), IEEE, pp. 120–126.
- [14] ASHRAE, S. V., AND PHAM, H. Reliability analysis of dynamic fault tree models. In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual* (2009), IEEE, pp. 1–6.
- [15] ASHRAE, S. V., PHAM, H., AND DELL, G. Optimal design of k-out-of-G subsystems subjected to imperfect fault-coverage. *Reliability, IEEE Transactions on* 55, 4 (2004), 567–575.
- [16] ASHRAE, S. V., PHAM, H., CHOU, K., AND PHAM, S. K. A novel approach for sparse optimization of complex repairable systems. In *Reliability and Maintainability Symposium, 2005. Proceedings. Annual* (2005), IEEE, pp. 355–360.

- [17] AMADE, S. V., ZOU, M. J., AND DELL, G. A fast and robust reliability evaluation algorithm for generalized multi-state-out-of-systems. *Reliability, IEEE Transactions on* 58, 1 (2009), 88–97.
- [18] ANDO, T., OKAMURA, H., AND DOHI, T. Estimating markov model-related software reliability models via an algorithm. In *Dependable, Autonomous and Secure Computing, 2nd IEEE International Symposium on* (2006), IEEE, pp. 111–118.
- [19] AO, Q., AI, J., LI, M., AND ZHONG, F. Scenario-based software operational profile. In *Reliability, Maintainability and Safety (RIMS)*, 2011 15th International Conference on (2011), IEEE, pp. 200–204.
- [20] ARALDO, J., MAROS, R., MARINI, P., VIERKA, J., MAROS, R., AND THIRVEDI, K. S. Software rejuvenation in embedded cloud computing: industrial use: a method based on time series forecasting and multiple thresholds. In *Software Aging and Rejuvenation (ISAR)*, 2011 IEEE Third International Workshop on (2011), IEEE, pp. 38–43.
- [21] BAO, Y., SUN, X., AND THIRVEDI, K. S. A workload-based analysis of software aging and rejuvenation. *Reliability, IEEE Transactions on* 54, 3 (2005), 341–348.
- [22] BRAW, T., AND NAGARATHAN, N. Tempore: Towards early identification of failure-prone binaries. In *Dependable Systems and Networks (DSN), FCS and DCS, 2008. IEEE/ACM 2008. ACM International Conference on* (2008), IEEE, pp. 116–121.
- [23] BIRD, C., NAGAPAN, S., GALL, H., MURPHY, H., AND THIRVEDI, K. Putting it all together: Using socio-technical networks to predict failures. In *Software Reliability Engineering, 2008. FORSE'08. 2008 International Symposium on* (2008), IEEE, pp. 109–119.
- [24] CAI, X., AND LATT, M. R. An empirical study on reliability modeling for diverse software systems. In *Software Reliability Engineering, 2004. ISRE 2004. 15th International Symposium on* (2004), IEEE, pp. 125–130.
- [25] CAI, X., AND LATT, M. R. The effect of code coverage on fault detection under different testing profiles. *ICM SIGSOFT Software Engineering Notes* 26, 4 (2003), 1–7.

- [26] CAI, X., AND LYU, M. R. Software reliability modeling with test coverage: Experimentation and measurement with a fault-tolerant software project. In *Software Reliability 2005: ISSRA'05: The 18th Asian International Symposium on* (2005), HIKS, pp. 17–26.
- [27] CAI, X., LYU, M. R., VOOR, M., ET AL. An experimental evaluation on reliability features of π -version programming. In *Software Reliability Engineering, 2005. ISREW 2005. 16th Asian International Symposium on* (2005), HIKS, pp. 10–pp.
- [28] CHANG, Y.-R., ASARI, S. V., AND KOO, S.-Y. Reliability evaluation of multi-state systems subject to imperfect coverage using ohsel. In *Dependable Computing, 2003. Proceedings, 2003 Pacific Rim International Symposium on* (2003), IEEE, pp. 403–409.
- [29] CHANG, Y.-R., ASARI, S. V., AND KOO, S.-Y. Grid-based evaluation of reliability and importance measures for multistate systems subject to imperfect fault coverage. *Dependable and Secure Computing, IEEE Transactions on* 2, 4 (2005), 336–347.
- [30] CHANG, Y.-R., HUANG, C.-Y., AND KOO, S.-Y. Performance assessment and reliability analysis of dependable and distributed computing systems based on fdd and restrictive merge. *Applied Mathematics and Computation* 217, 1 (2010), 403–413.
- [31] CHEN, C.-C., FAN, C.-T., HUANG, H.-H., HUANG, S.-W., AND HUANG, C.-Y. Gpsel: a computer-aided reliability assessment tool for software based on object-oriented design. In *VACATION 2006, 2006 Asian Vacuum 20 Conference* (2006), HIKS, pp. 1–5.
- [32] CHEN, D., ENAKSHARAJA, S., CHEN, D., LI, F., THIRUVI, K. S., SONTI, H. R., AND NIKHILA, A. P. Reliability and availability analysis for the ipf remote expedition and experimentation system. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings, International Conference on* (2005), HIKS, pp. 337–342.
- [33] CHEN, C., KANDAMU, M., TOSUN, S., AND SCHER, U. Reliability-conscious process scheduling under performance constraints in ipso-based embedded systems. In *Parallel and Distributed Processing Symposium, 2005. Proceedings, 19th IEEE International* (2005), HIKS, pp. 162a–162a.

- [39] CHEN, C., KANDAMUR, M., VEERAKRISHNAN, N., AND IRWIN, M. J. Object duplication for improving reliability. In *Proceedings of the 2006 Asia and South Pacific Design Automation Conference* (2006), IEEE Press, pp. 140–145.
- [40] CHOUDHURY, S. R., ZHANG, D., VEMPER, H., AND GROSS, A. Web2000 Web application test repair. In *Proceedings of the First International Workshop on Web-to-Web Test Script Engineering* (2011), ACM, pp. 21–29.
- [41] CONNOR, J., XU, Y., KANDAMUR, M., LISK, G., AND DICK, R. Ed-logic: A hybrid heuristic/genetic algorithm hardware/software co-synthesis framework with fault detection. In *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific* (2005), vol. 2, IEEE, pp. 709–712.
- [42] CORMACK, J., DAS, R., NAVARATHNAN, N., TEJES, A., AND TEJES, A. Cramet: Failure prediction, change analysis and test prioritization in practice: experiences from windows. In *Software Testing, Verification and Validation (ICST'07), 2007 IEEE Fourth International Conference on* (2007), IEEE, pp. 357–365.
- [43] DAVIDSON, M., ZHANG, J., NAVARATHNAN, N., WILLIAMS, L., AND YOUNG, M. Gens: An empirical reliability estimation and testing feedback tool. In *Software Reliability Engineering, 2002. ISSRE 2002. 15th International Symposium on* (2002), IEEE, pp. 269–280.
- [44] DESTEFANO, S., LEBLOCK, P., AND FURBER, K. S. Investigating dynamic reliability and availability through state space models. *Computers & Mathematics with Applications* 65, 12 (2012), 3701–3716.
- [45] DOTT, T., OKAMURA, H., AND KANE, N. Optimal age-dependent checkpoint strategy with policy of rollback recovery. In *Autonomous Distributed Systems, 2003. The 2nd International Workshop on* (2003), IEEE, pp. 113–118.
- [46] DOTT, T., OKAMURA, H., AND TEJES, K. S. Optimizing software rejuvenation policies under interval reliability criteria. In *Computing Intelligence in Computing and 9th International Conference on Autonomous & Trusted Computing (CIC/ATC), 2012 9th International Conference on* (2012), IEEE, pp. 178–185.

- [42] Itoh, T., Osaki, S., AND Irie, K. S. An infinite server queueing approach for describing software reliability growth: unified modeling and estimation framework. In *Software Engineering Conference*, 2002, 33th Asia-Pacific (2004), IEEE, pp. 110–119.
- [43] Itoh, T., AND Uehara, T. An adaptive multi control algorithm of a scalable intrusion tolerant architecture. *Journal of Computer and System Sciences* 28, 6 (2012), 1751–1774.
- [44] Itoh, T., Yasui, K., AND Osaki, S. Software reliability assessment model based on cumulative bernoulli trial processes. *Mathematical and computer modelling* 38, 11 (2003), 1177–1184.
- [45] Iwata, M. H., Pearson, M., AND FUSALDI, S. Controlling factors in evaluating path-sensitive error detection techniques. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering* (2006), ACM, pp. 92–101.
- [46] FONDELLA, L., AND CORRALE, S. S. Importance measures for a modular software system (short paper). In *Quality Software, 2008. QSIC'08. 23rd English International Conference on* (2008), IEEE, pp. 338–343.
- [47] FONDELLA, L., AND CORRALE, S. S. Software reliability with architectural uncertainties. In *Parallel and Distributed Processing, 2008. PDP 2008. IEEE International Symposium on* (2008), IEEE, pp. 1–5.
- [48] FONDELLA, L., AND CORRALE, S. S. Software reliability model with bathtub-shaped fault detection rate. In *Reliability and Maintainability Symposium, 2011. Proceedings Annual* (2011), IEEE, pp. 1–6.
- [49] FONDELLA, L., AND CORRALE, S. S. Optimal allocation of testing effort considering software architecture. *Reliability, IEEE Transactions on* 61, 2 (2012), 340–350.
- [50] FONDELLA, L., RAJESKARAN, S., AND CORRALE, S. S. Efficient system reliability with correlated component failures. In *High-Assurance Systems Engineering (HASE), 2011 IEEE 13th International Symposium on* (2011), IEEE, pp. 269–276.
- [51] FORTES, H. M., AND IRIE, K. S. Importance analysis with markov chains. In *Reliability and Maintainability Symposium, 2003. Annual* (2003), IEEE, pp. 89–95.

- [52] FU, J., LU, M., AND AL, J. Test: An automation tool for software reliability test data generation. In *Advances in Computational Tools for Engineering Applications 2009, A.CTSE4-09. International Conference on (2009)*, IEEE, pp. 447–450.
- [53] FUJII, T., DEHN, T., AND FUJIMURA, T. Towards quantitative software reliability assessment in hierarchical development process. In *Proceedings of the 33rd International Conference on Software Engineering (2011)*, ACM, pp. 51–60.
- [54] FUJII, T., DEHN, T., OKAMURA, H., AND FUJIMURA, T. A software accelerated life testing model. In *Dependable Computing (FACED), 2010 DCC/ FACS Pacific Asia International Symposium on (2010)*, IEEE, pp. 88–92.
- [55] FUJIMURA, T., KATSUME, J. M., SATOH, Y., AND YAMADA, S. A calculation method for software safety integrity level. In *Proceedings of the 1st Workshop on Critical Infrastructure Applications: Resilience & Safety (2010)*, ACM, pp. 41–54.
- [56] FUJIMURA, T., KIMURA, M., SATOH, Y., AND YAMADA, S. A method of calculating safety integrity level for iso 61508 conformity software. In *Dependable Computing (FACED), 2011 DCC/ FACS Pacific Asia International Symposium on (2011)*, IEEE, pp. 295–301.
- [57] GAO, M., KUMAR, R., LU, M., AND YIN, Y. Research on test requirement modeling for software-intensive systems and its test implementation. In *Digital Automatic Systems Conference, 2007, DASIS'07, 2007/12/14-16 2007*, IEEE, pp. 9–13.
- [58] GIBSON, M., WILLIAMS, L., OSBORNE, J., AND VOOR, M. Enhancing software security fortification through role-based metrics. In *Proceedings of the 4th ACM workshop on Quality of protection (QIPRO), ACM*, pp. 31–38.
- [59] GIBSON, R., KIM, D., AND FRYER, K. S. System resilience quantification using risk sub-models and state-space analytic models. *Reliability Engineering & System Safety* 116 (2012), 109–125.
- [60] GORCHALE, S. S. Optimal software release time incorporating fault correction. In *Software Engineering Workshops, 2002. Proceedings, 20th Annual NAGSA-Goodland (2002)*, IEEE, pp. 175–184.

- [61] GOKHALE, S. S. Cost constrained reliability maximization of software systems. In *Reliability and Maintainability, 2004 Annual Symposium-W43467* (2004), IEEE, pp. 100–106.
- [62] GOKHALE, S. S. Quantifying the variance in application reliability. In *Dependable Computing 2003, Proceedings, 10th IEEE Pacific Rim International Symposium on* (2003), IEEE, pp. 113–121.
- [63] GOKHALE, S. S. Software application design based on architecture, reliability and cost. In *Computers and Communications, 2004, Proceedings, ISCC 2004, Ninth International Symposium on* (2004), vol. 2, IEEE, pp. 1098–1103.
- [64] GOKHALE, S. S. Software failure rate and reliability incorporating repair policies. In *Software Metrics, 2004, Proceedings, 16th International Symposium on* (2004), IEEE, pp. 394–401.
- [65] GOKHALE, S. S. Software reliability analysis with component-level fault tolerance. In *Reliability and Maintainability Symposium* (2005), pp. 24–27.
- [66] GOKHALE, S. S. Variance expressions for software reliability growth models. In *IEEE Software and Maintainability Symposium, Proceedings Annual* (2005), pp. 628–633.
- [67] GOKHALE, S. S. Architecture-based software reliability analysis: Overview and limitations. *IEEE Transactions on Dependable and Secure Computing*, 4 (2007), 32–40.
- [68] GOKHALE, S. S., LEE, M. H., AND ISHMAN, R. S. Incorporating fault debugging activities into software reliability models: A simulation approach. *Reliability, IEEE Transactions on* 55, 2 (2006), 281–292.
- [69] GOKHALE, S. S., AND LEE, M.-T. A simulation approach to structure-based software reliability analysis. *Software Engineering, IEEE Transactions on* 31, 8 (2005), 643–658.
- [70] GOKHALE, S. S., AND MENDURITA, V. D. Architecture-based assessment of software reliability. In *Quality Software, 2008, QSWC'08, The Ninth International Conference on* (2008), IEEE, pp. 444–444.

- [71] CORRALE, S. S., AND MULLEN, R. E. From test count to code coverage using the logarithmic failure rate. In *Software Reliability Engineering*, 2004. *ISSRE 2004. 15th International Symposium on* (2004). IEEE, pp. 235–249.
- [72] CORRALE, S. S., AND MULLEN, R. E. Dynamic code coverage metrics: A lognormal perspective. In *Software Metrics, 2005. 15th IEEE International Symposium (2005)*. IEEE, pp. 19 pp.
- [73] CORRALE, S. S., AND MULLEN, R. E. Updating models for field defect resolution process. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (2006). IEEE, pp. 353–362.
- [74] CORRALE, S. S., AND THURVED, K. S. Reliability prediction and sensitivity analysis based on software architectures. In *Software Reliability Engineering, 2002. IEEE 2002. Proceedings. 15th International Symposium on* (2002). IEEE, pp. 64–75.
- [75] CORRALE, S. S., AND THURVED, K. S. Analytical models for architecture-based software reliability prediction: A unification framework. *Reliability, IEEE Transactions on* 55, 4 (2006), 578–590.
- [76] CORRALE, S. S., VANDAL, P. J., AND LU, J. Performance and reliability analysis of web server architectures. In *Dependable Computing, 2006. DEPEND'06. 12th Pacific Rim International Symposium on* (2006). IEEE, pp. 331–358.
- [77] CORRALE, S. S., WONG, W. E., DEBBAN, J. R., AND THURVED, K. S. An analytical approach to architecture-based software performance and reliability prediction. *Performance Evaluation* 53, 4 (2004), 391–412.
- [78] GEORGIU, M., FU, L., MADEVANATHAN, R., AND THURVED, K. S. Analysis of software aging in a web server. *Reliability, 2006. VTSAC 2006. Analysis of Software Aging in a Web Server. Reliability, 2006. IEEE Symposium on* 55, 2 (2006), 411–420.
- [79] GEORGIU, M., MANN, R., AND THURVED, K. S. The fundamentals of software aging. In *Software Reliability Engineering Workshops, 2005. IEEE WSEAP 2005. IEEE International Conference on* (2005). IEEE, pp. 1–6.

- [80] CHARTRE, M., AND THIRYON, K. S. On a method for modeling time to failure distributions. In *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on* (2005), IEEECS, pp. 500–509.
- [81] CHU, P., AND LYO, M. R. A pseudorecursive learning algorithm for feedforward neural networks with stacked generalization applications to software reliability growth data. *Neurocomputing* 56 (2004), 101–121.
- [82] HARRINGTON, M., AND THIRYON, K. Availability monitor for a software based system. In *1999 IEEE High Assurance Systems Engineering Symposium, 2000. HASES'00. (2000)*, IEEECS, pp. 321–328.
- [83] HARRISON, W. G., AND ORESO, A. Automated identification of parameter mismatches in web applications. In *Proceedings of the 15th ACM SIGSOFT International Symposium on Foundations of software engineering* (2008), ACM, pp. 181–191.
- [84] HOFFMANN, G. A., THIRYON, K. S., AND MATHUR, M. A best practice guide to resource forecasting for the apollo webserver. In *Dependable Computing, 2000. DEPEND'00, 19th Pacific Rim International Symposium on* (2000), IEEECS, pp. 183–190.
- [85] HONG, Y., THIRYON, K. S., HAY, A., AND PRINIA, S. Software performance analysis using a language heuristic. In *Proceedings of the American Nuclear Conference* (2003), vol. 3, pp. 2258–2263.
- [86] HSU, C.-J., AND HUANG, C.-Y. Integration path testing with software reliability estimation using control flow graph. In *Proceedings of International and Technology, 2008. ICTAIT'08, 4th IEEE International Conference on* (2008), IEEECS, pp. 1234–1239.
- [87] HSU, C.-J., AND HUANG, C.-Y. Reliability analysis using weighted combinatorial models for web-based software. In *Proceedings of the 18th international conference on Model driven arch.* (2009), ACM, pp. 1131–1137.
- [88] HSU, C.-J., AND HUANG, C.-Y. A study on the applicability of modified genetic algorithm for the parameter estimation of software reliability modeling. In *Computer Software and Applications Conference (COMPSAC), 2009 IEEE High Assurance (2009)*, IEEECS, pp. 531–540.

- [80] HSU, C.-J., AND HUANG, C.-Y. An adaptive reliability analysis using path testing for complex component-based software systems. *Reliability, IEEE Transactions on* 69, 1 (2011), 158–170.
- [81] HSU, C.-J., HUANG, C.-Y., AND CHEN, T.-Y. A modified genetic algorithm for parameter estimation of software reliability growth models. In *IEEE International Symposium on Software Reliability Engineering* (2008), IEEE, pp. 281–282.
- [82] HU, J. S., LI, P., EDEKATI, V., KANDAMBA, M., VIKARAKIS, N., AND LEWIS, M. J. Compiler-directed instruction duplication for soft error detection. In *Proceedings of the conference on Design, Automation and Test in Europe—Volume 2* (2005), IEEE Computer Society, pp. 1056–1057.
- [83] HU, X., LIU, H., AND WANG, Y. Research on code pattern automatic-based code error pattern automatic detection technique. In *Reliability, Maintainability and Safety, 2009. RIMS 2009. 8th International Conference on* (2009), IEEE, pp. 754–760.
- [84] HUANG, D., ZHANG, H., AND LU, M. Software fault approach based on failure modes database. In *Reliability, Maintainability and Safety, 2009. RIMS 2009. 8th International Conference on* (2009), IEEE, pp. 749–753.
- [85] HUANG, H., ZHANG, H., LU, M., AND MINN, L. A perturbation model for software fault. In *Reliability, Maintainability and Safety (RIMS), 2011 8th International Conference on* (2011), IEEE, pp. 724–729.
- [86] HUANG, C.-Y. Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency. *Journal of Systems and Software* 77, 2 (2005), 129–155.
- [87] HUANG, C.-Y. Performance analysis of software reliability growth models with testing-effort and change-point. *Journal of Systems and Software* 76, 2 (2005), 181–191.
- [88] HUANG, C.-Y., AND CHANG, Y.-H. An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees. *Reliability Engineering & System Safety* 95, 10 (2007), 1403–1412.

- [198] HUANG, C.-Y., AND HUANG, W.-C. Software reliability analysis and measurement using finite and infinite server queuing models. *Reliability, IEEE Transactions on* 57, 1 (2006), 192–203.
- [199] HUANG, C.-Y., AND HUNG, T.-Y. Software reliability analysis and assessment using queuing models with multiple change-points. *Coveragers & Mathematics with Applications* 69, 7 (2016), 2013–2030.
- [200] HUANG, C.-Y., HUNG, T.-Y., AND HEO, C.-J. Software reliability prediction and analysis using queuing models with multiple change-points. In *Secure Software Integration and Reliability Engineering*, 2009. *SSIRE 2009, Third Asian International Conference on* (2009), IEEE, pp. 212–221.
- [201] HUANG, C.-Y., AND KOO, S.-Y. Analysis of incorporating logistic testing-effort function into software reliability modeling. *Reliability, IEEE Transactions on* 51, 3 (2002), 261–276.
- [202] HUANG, C.-Y., KOO, S.-Y., AND LYO, M. B. An assessment of testing-effort dependent software reliability growth models. *Reliability, IEEE Transactions on* 56, 2 (2007), 198–211.
- [203] HUANG, C.-Y., AND LIN, C.-T. Software reliability analysis by considering fault dependency and debugging time lag. *Reliability, IEEE Transactions on* 55, 3 (2006), 436–450.
- [204] HUANG, C.-Y., AND LIN, C.-T. Analysis of software reliability modeling considering testing compression factor and failure-to-fault relationship. *Computers, IEEE Transactions on* 55, 2 (2006), 282–288.
- [205] HUANG, C.-Y., LIN, C.-T., KOO, S.-Y., LYO, M. B., AND SHEN, C.-C. Software reliability growth models incorporating fault dependency with various debugging time lags. In *Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 2004 Annual International* (2004), IEEE, pp. 180–191.
- [206] HUANG, C.-Y., LIN, C.-T., LO, J.-H., AND SHEN, C.-C. Effect of fault dependency and debugging time lag on software error models. In *TRUST 2004, 2004 IEEE Region 10 Conference* (2004), IEEE, pp. 232–246.
- [207] HUANG, C.-Y., LIN, C.-T., AND SHEN, C.-C. Considering fault dependency and debugging time lag in reliability growth modeling during

- software testing. In *Test Symposium, 2000. 15th Asian (2000)*. IEEE, pp. 378–383.
- [108] HUANG, C.-Y., LIN, C.-F., AND SUN, C.-C. Software reliability prediction and analysis during operational use. In *Information Technology Research and Education, 2005. ITRE 2005. 2nd International Conference on (2005)*. IEEE, pp. 317–321.
 - [109] HUANG, C.-Y., AND LO, J.-H. Optimal resource allocation for cost and reliability of modular software systems in the testing phase. *Journal of Systems and Software* 79, 5 (2005), 633–654.
 - [110] HUANG, C.-Y., LO, J.-H., KOO, S.-Y., AND LYO, M. R. Optimal allocation of testing resources for modular software systems. In *Software Reliability Engineering, 2002. ASSAT 2002. Proceedings. 13th International Symposium on (2002)*. IEEE, pp. 129–138.
 - [111] HUANG, C.-Y., LO, J.-H., KOO, S.-Y., SUN, C.-C., AND LIN, C.-F. Optimal resource allocation and sensitivity analysis for modular software testing. In *Multimedia Software Engineering, 2005. Proceedings. FOMA International Symposium on (2005)*. IEEE, pp. 231–238.
 - [112] HUANG, C.-Y., AND LYO, M. R. Optimal release time for software systems considering cost, testing-effort, and test-efficiency. *Reliability, IEEE Transactions on* 54, 4 (2005), 588–593.
 - [113] HUANG, C.-Y., AND LYO, M. R. Optimal testing resource allocation, and sensitivity analysis in software development. *Reliability, IEEE Transactions on* 54, 4 (2005), 593–603.
 - [114] HUANG, C.-Y., AND LYO, M. R. Estimation and analysis of some generalized multiple change-point software reliability models. *Reliability, IEEE Transactions on* 59, 2 (2011), 498–514.
 - [115] HUANG, C.-Y., LYO, M. R., AND KOO, S.-Y. A unified scheme of some multichange-point poisson process models for software reliability estimation. *Software Engineering, IEEE Transactions on* 29, 3 (2003), 261–269.
 - [116] HUANG, W.-C., HUANG, C.-Y., AND SUN, C.-C. Software reliability prediction and assessment using both finite and infinite server queueing approaches. In *Dependable Computing, 2006. DEPC'06. 13th Pacific Rim International Symposium on (2006)*. IEEE, pp. 191–201.

- [117] HUANG, Y.-C., PENG, K.-L., AND HUANG, C.-Y. A history-based cost-equivalent test case prioritization technique in regression testing. *Journal of Systems and Software* 85, 3 (2012), 626–637.
- [118] HUPPMAN, D., REICHMAN, R., ASARI, S. V., AND ZUO, M. J. Assailability analysis of systems with suspended animation. In *Reliability and Maintainability Symposium*, 2009, RAMS 2009, January (2009), IEEE, pp. 283–288.
- [119] INOUE, S., ITOH, T., AND OKAMURA, H. Estimating software reliability via pseudo maximum likelihood method. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (2012), ACM, pp. 1171–1176.
- [120] INOUE, S., AND YAMADA, S. Discrete software reliability assessment with discretised clapp models. *Computers & Mathematics with Applications* 51, 2 (2006), 161–170.
- [121] INOUE, S., AND YAMADA, S. Generalized discrete software reliability modeling with effect of program size. *Systems, Man and Cybernetics, Part A: Systems and Humans*, IEEE Transactions on 37, 2 (2007), 170–179.
- [122] INOUE, S., AND YAMADA, S. Optimal software release policy with change-point. In *Industrial Engineering and Engineering Management*, 2008, IEEM 2008, IEEM International Conference on (2008), IEEE, pp. 521–525.
- [123] INOUE, S., AND YAMADA, S. Two-dimensional software reliability assessment with testing coverage. In *Software Systems Integration and Reliability Improvement*, 2008, SSRI 08, Second International Conference on (2008), IEEE, pp. 150–157.
- [124] INOUE, S., AND YAMADA, S. Two-dimensional software reliability measurement techniques. In *Industrial Engineering and Engineering Management*, 2009, IEEM 2009, IEEM International Conference on (2009), IEEE, pp. 224–227.
- [125] INOUE, S., AND YAMADA, S. Change-point modeling for software reliability assessment depending on two-types of reliability growth factors. In *Industrial Engineering and Engineering Management (IEEM)*, 2010, IEEM International Conference on (2010), IEEE, pp. 626–629.

- [126] Imai, S., AND YAMADA, S. A bootstrap method for software reliability assessment based on a discretized nhpp model. In *Dependable Transportation Systems/Recent Advances in Software Dependability (RTSDS-2012 Workshop on)* (2012). IEEE, pp. 25–27.
- [127] Imai, T., AND DORI, T. Two-dimensional software reliability models and their application. In *Dependable Computing, 2000. Dependable Computing, 2000. Pacific Asia International Symposium on* (2000). IEEE, pp. 3–10.
- [128] Imai, T., AND DORI, T. A new paradigm for software reliability modeling from nhpp to nhgp. In *Dependable Computing, 2000. Dependable Computing, 2000. Pacific Asia International Symposium on* (2000). IEEE, pp. 221–231.
- [129] Imai, T., DORI, T., AND OKAMURA, H. Discrete software fault-detection models. In *wdi* (2007). IEEE, pp. 522–528.
- [130] IWASAKI, K., DORI, T., OKAMURA, H., AND KATO, N. Discrete-time cost analysis for a telecommunication billing application with rejuvenation. *Computers & Mathematics with Applications* 51, 2 (2006), 885–894.
- [131] IWAMOTO, K., OKAMURA, H., AND NAKANO, K. Discrete availability models to represent a telecommunication billing application. *IEEE Transactions on communications* 56, 10 (2008), 2931–2939.
- [132] JACOB, P., ET AL. Analysis of complex dependable systems. 189–199.
- [133] JAYEN, C., LIO, H., YIN, Y., AND LIO, C. Study on real-time test script in automated test equipment. In *Reliability, Maintainability and Safety, 2009. RAMS 2009 20th International Conference on* (2009). IEEE, pp. 768–772.
- [134] JIN, L.-Z., AND IMAI, T. Non-homogeneous inverse gaussian software reliability models. In *Quality Reliability, Risk, Maintenance, and Safety Engineering (QR2010/R2), 2011 International Conference on* (2011). IEEE, pp. 444–451.
- [135] JIN, L.-Z., DORI, T., AND OSAKI, S. Continuous software reliability models. In *Quality Reliability, Risk, Maintenance, and Safety Engineering (QR2010/R2), 2011 International Conference on* (2011). IEEE, pp. 405–410.

- [136] KANEHSHI, T., AND DOHI, T. Parametric bootstrapping for assessing software reliability measures. In *Dependable Computing (PRDC)*, 2012. *IEEE*, 2012. Pacific Rim International Symposium on (2011), 815-817, pp. 1-9.
- [137] KLEIN, A., CHANG, Y., ANDER, S., GUN, P. J., HODENBACH, P., AND FISHER, M. D. Barend: A solver for word equations over strings, regular expressions, and context-free grammars. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 21, 4 (2012), 25.
- [138] KIM, D. S., CHOSI, R., AND THIVIAZ, K. S. A hierarchical model for reliability analysis of sensor networks. In *Dependable Computing (PRDC)*, 2010. *IEEE*, 2010. Pacific Rim International Symposium on (2010), IEEE, pp. 247-248.
- [139] KIM, D. S., MARTIN, P., AND THIVIAZ, K. S. Availability modeling and analysis of a virtualized system. In *Dependable Computing, 2009. PRDC'09, 2009. IEEE Pacific Rim International Symposium on* (2009), IEEE, pp. 363-371.
- [140] KIM, S., ZIMMERMAN, T., AND NAGAPPAN, N. Crash graphs: An aggregated view of multiple crashes to improve crash usage. In *Dependable Systems & Networks (DSN)*, 2013. *IEEE/ACM*, 2013. International Conference on (2013), IEEE, pp. 486-490.
- [141] KRISHNA, N. S. H., SOK, S. W., KANISUNG, M., AND LI, P. Using loop invariants to fight soft errors in data caches. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference (2005)*, ACM, pp. 1317-1320.
- [142] KUDRAVITS, G., NAGAPPAN, N., AND RALE, T. Assessing the relationship between software assertions and faults: An empirical investigation. In *ISSTA* (2009), vol. 6, pp. 201-212.
- [143] Kuo, C.-S., AND HUANG, C.-Y. A study of applying the bounded generalized pareto distribution to the analysis of software fault distribution. In *Industrial Engineering and Engineering Management (IEEM)*, 2010. *IEEE International Conference on* (2010), IEEE, pp. 611-615.
- [144] LEE, K. H., ZHENG, Y., SUNGER, N., AND ZHANG, X. Toward improving reliable copy ops. *ACM SIGPLAN Notices* 46, 6 (2011), 946-957.

- [145] Li, Q., AND LU, M. A framework for online software reliability evaluation. In *World Automation Congress 2012* (2012).
- [146] Li, Q., LU, M., AND LIU, H. Sba-based safety analysis for bridge pattern. In *Reliability, Information and Safety, 2009. RICIS 2009. 30th International Conference on* (2009), IEEE, pp. 522–526.
- [147] Li, H., Li, Q., AND LU, M. Software reliability modeling with logistic test coverage function. In *Software Reliability Engineering, 2009. RSSN 2009. 30th International Symposium on* (2009), IEEE, pp. 319–323.
- [148] Li, H., LU, M., AND LI, Q. Software reliability metrics selecting method based on analytic hierarchy process. In *Quality Software, 2009. QSW 2009. Sixth International Conference on* (2009), IEEE, pp. 337–345.
- [149] Li, L., VALLANATHAN, K., AND THIRUCH, S. S. An approach for estimation of software aging in a web server. In *Computer Software Engineering, 2003. Proceedings. 2003. International Symposium on* (2003), IEEE, pp. 91–100.
- [150] Li, M.-L., RANAKANTHAN, P., SANDHU, S. K., ARORA, S. V., ARORA, V. S., AND ZHOU, Y. Understanding the propagation of hard errors to software and implications for resilient system design. In *ACM SIGARCH Computer Architecture News* (2008), vol. 36, ACM, pp. 265–275.
- [151] Li, Q., LU, L., AND AN, J. The determination method for software reliability qualitative indices. In *Software Security and Reliability (SRSS), 2013. IEEE 3th International Conference on* (2013), IEEE, pp. 207–215.
- [152] Li, Q.-Y., JIANG, M.-C., LI, H.-P., AND LU, M.-Y. Software reliability qualitative evaluation method based on bayesian networks. In *Reliability, Information and Computer (RICOI), 2010. 2nd International Conference on* (2010), vol. 1, IEEE, pp. 74–78.
- [153] Li, S., XIE, Q., CHU, P., AND LIU, M. H. A hierarchical mixture model for software reliability prediction. *Applied mathematics and computation* 188, 2 (2007), 1120–1130.

- [159] LI, Z., LIU, D., MA, N., AND YIN, Y. Formal testing applied in embedded software. In *Reliability, Maintainability and Safety, 2005. RAMS 2005. 2005 20th International Conference on* (2005), IEEE, pp. 687–702.
- [160] LI, X., FAN, L., WANG, X., FAN, S., XIESE, Y., AND ZHAI, C. Have things changed now?: an empirical study of bug characteristics in modern open source software. In *Proceedings of the 1st workshop on Architectural and system support for managing software dependency* (2000), ACM, pp. 25–33.
- [161] LIN, C.-T., AND HUANG, C.-Y. Software reliability modeling with multiple testing-effect and multiple change-points. In *IEEE/ACM 2005 2005 JSSAC Workshop* 20 (2005), IEEE, pp. 1–6.
- [162] LIN, C.-T., AND HUANG, C.-Y. Software release time management: how to use reliability growth models to make better decisions. In *Management of Innovation and Technology, 2006 IEEE International Conference on* (2006), vol. 2, IEEE, pp. 658–662.
- [163] LIN, C.-T., AND HUANG, C.-Y. Estimating and measuring the predictive capabilities of testing-effect dependent software reliability models. *Journal of Systems and Software* 82, 6 (2008), 1025–1038.
- [164] LIN, C.-T., AND HUANG, C.-Y. Modeling the software failure correlations when test automation is adopted during the software development. In *Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on* (2008), IEEE, pp. 307–308.
- [165] LIN, C.-T., AND HUANG, C.-Y. Scaling level and cost analysis for software debugging activities through rate-based simulation approaches. *Reliability, 2008. 2008 Symposium on* 58, 1 (2008), 711–724.
- [166] LIN, C.-T., HUANG, C.-Y., AND SHU, C.-C. Measuring and assessing software reliability growth through simulation-based approaches. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International* (2007), vol. 1, IEEE, pp. 439–448.
- [167] LIN, C.-T., TANG, K.-W., CHANG, J.-H., AND HUANG, C.-Y. An investigation into whether the mhp framework is suitable for software reliability prediction and estimation. In *Software Engineering and Knowledge Management / SEKM 2010, 2010 IEEE International Conference on* (2010), IEEE, pp. 620–630.

- [168] LEE, C., LEE, D., AND RUAN, L. A reliability model based on heterogeneous software architectures. In *Reliability, Maintainability and Safety, 2000. RAMS 2000. 20th International Conference on* (2000). IEEE, pp. 728–732.
- [169] LEE, Y., MA, Y., HAN, J. J., LEEVERMAN, H., AND THIRVUM, K. S. A proposed approach towards always-on availability in broadband cable networks. *Computer Communications* 28, 1 (2005), 51–64.
- [170] LEE, Y., NIRMALAKUTTA, V. H., AND THIRVUM, K. S. Survivability analysis of telephone access network. In *Software Reliability Engineering, 2004. ISSRE 2004. 20th International Symposium on* (2004). IEEE, pp. 367–377.
- [171] LEE, Y., THIRVUM, K. S., MA, Y., HAN, J. J., AND LEEVERMAN, H. Modeling and analysis of software rejuvenation in cable modem termination systems. In *Software Reliability Engineering, 2005. ISSRE 2005. Proceedings. 20th International Symposium on* (2005). IEEE, pp. 159–170.
- [172] LO, H.-H., HUANG, C.-Y., CHANG, Y.-R., HUANG, W.-C., AND CHANG, F.-R. Reliability and sensitivity analysis of embedded systems with modular dynamic fault trees. In *TENCON 2005 2005 IEEE Region 10 Conference* (2005). IEEE, pp. 1–6.
- [173] LO, J.-H., AND HUANG, C.-Y. Incorporating imperfect debugging into software fault processes. In *TENCON 2004 2004 IEEE Region 10 Conference* (2004). IEEE, pp. 328–329.
- [174] LO, J.-H., AND HUANG, C.-Y. An integration of fault detection and correction processes in software reliability analysis. *Journal of Systems and Software* 78, 9 (2006), 1512–1523.
- [175] LO, J.-H., HUANG, C.-Y., CHAN, Y., KOO, S.-Y., AND LO, M.-R. Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure. *Journal of Systems and Software* 78, 1 (2005), 3–13.
- [176] LO, J.-H., KOO, S.-Y., LO, M. R., AND HUANG, C.-Y. Optimal resource allocation and reliability analysis for component-based software applications. In *Computer Software and Applications Conference, 2002. COMSAF 2002. Proceedings. 20th Annual International* (2002). IEEE, pp. 7–12.

- [172] LYO, M. R., HUANG, Z., SEE, S. K., AND CAI, X. An empirical study on testing and fault tolerance for software reliability engineering. In *Software Reliability Engineering, 2002. ISSRE 2002. 12th International Symposium on* (2002), IEEE, pp. 119–129.
- [173] LYO, M. R., RASCHAUER, S., AND VAN MEUSEL, A. Optimal allocation of test resources for software reliability growth modeling in software development. *Reliability, IEEE Transactions on* 51, 2 (2002), 181–192.
- [174] MACQUEEN, P., KIM, D. S., AND THIRVEDI, K. S. Modeling and analysis of software rejuvenation in a server virtualized system with live-vms migration. *Performance Evaluation* 70, 4 (2011), 232–249.
- [175] MATHIAS, JR., R., HANSEN, R. A., THIRVEDI, K. S., AND PAVLO PETER, J. E. Accelerated degradation tests applied to software aging experiments. *Reliability, IEEE Transactions on* 50, 1 (2001), 112–114.
- [176] MATHIAS, JR., R., THIRVEDI, K. S., AND MATHI, P. R. Using accelerated life tests to estimate time to software aging failure. In *Software Reliability Engineering (ISSRE), 2010 IEEE 22nd International Symposium on* (2010), IEEE, pp. 241–249.
- [177] MISRA, K., AND THIRVEDI, K. S. Uncertainty propagation through software dependability models. In *Software Reliability Engineering (ISSRE), 2011 IEEE 23rd International Symposium on* (2011), IEEE, pp. 83–92.
- [178] MULLER, R. E., AND CORRALE, S. S. Software defect collection over a discrete lognormal model. In *Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on* (2005), IEEE, pp. 10–19.
- [179] NAGARAJAN, N. Toward a software testing and reliability early warning metric suite. In *Proceedings of the 20th international conference on software engineering* (2004), IEEE Computer Society, pp. 60–62.
- [180] NAGARAJAN, N., AND HALL, T. Use of relative code change measures to predict system defect density. In *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference on* (2005), IEEE, pp. 285–292.

- [181] NAGARAJAN, N., AND DODD, T. Using software dependencies and chain metrics to predict field failures: An empirical case study. In *Computer Software Engineering and Assessment*, 2007. *CSSE&A 2007. First International Symposium on* (2007), IEEE, pp. 364–370.
- [182] NAGARAJAN, N., DODD, T., AND MURPHY, D. Using historical in-process and product metrics for early estimation of software failures. In *ISSNAN* (2006), vol. 6, Chicago, pp. 62–74.
- [183] NAGARAJAN, N., MURPHY, D., AND BASSETT, V. The influence of organizational structure on software quality: an empirical case study. In *Proceedings of the 2006 international conference on Software engineering* (2006), ACM, pp. 521–530.
- [184] NAGARAJAN, N., WILLIAMS, L., HENDERSON, J., SANCHEZ, W., AND VORON, S. Preliminary results on using static analysis tools for software inspection. In *Software Reliability Engineering*, 2004. *ISSRE 2004. 15th International Symposium on* (2004), IEEE, pp. 429–439.
- [185] NAGARAJAN, N., WILLIAMS, L., VORON, S., AND CLARKE, J. Using in-process testing metrics to estimate post-release field quality. In *Software Reliability, 2007. ISSRE '07. The 18th IEEE International Symposium on* (2007), IEEE, pp. 209–214.
- [186] NAGARAJAN, N., ZIEGLER, A., ZIMMERMAN, T., HERRIG, K., AND MURPHY, D. Change bursts as defect predictors. In *Software Reliability Engineering (ISSRE)*, 2010. *IEEE 21st International Symposium on* (2010), IEEE, pp. 309–318.
- [187] NICO, D. M., SANCHEZ, W. H., AND TRIVIA, K. S. Model-based evaluation: from dependability to security. *Dependable and Secure Computing, IJCCW Transactions on* 1, 4 (2003), 48–65.
- [188] OCHS, K., OKAMURA, H., AND DODD, T. Comparing software reliability model and its application. In *Computer Software and Applications Conference, 2005. COMPSAC 2005. 29th Annual International* (2005), vol. 3, IEEE, pp. 345–440.
- [189] OCHS, K., OKAMURA, H., AND DODD, T. Comparing software reliability model: Estimation algorithm and empirical validation. *Journal of Systems and Software* 82, 3 (2009), 535–545.

- [1180] OKAMURA, H., AND DOHI, T. Building phase-type software reliability models. In *Software Reliability Engineering, 2006. ISSREW. 17th International Symposium on* (2006), IEEE, pp. 289–298.
- [1181] OKAMURA, H., AND DOHI, T. Availability optimization in open-ended software system with aperiodic time-based software rejuvenation scheme. In *International Conference on Software Reliability Engineering Workshops, 2008. ISSREW Workshops (2008)*, IEEE.
- [1182] OKAMURA, H., AND DOHI, T. Hyper-exact software reliability model. In *Dependable Computing, 2008. DEPEND'08. 14th IEEE Pacific Rim International Symposium on* (2008), IEEE, pp. 232–239.
- [1183] OKAMURA, H., AND DOHI, T. Comprehensive evaluation of aperiodic checkpointing and rejuvenation schemes in operational software system. *Journal of Systems and Software* 82, 9 (2009), 1599–1611.
- [1184] OKAMURA, H., AND DOHI, T. Application of reinforcement learning to software rejuvenation. In *Automations Dependable Systems (ASDAAS), 2011. 10th International Symposium on* (2011), IEEE, pp. 647–652.
- [1185] OKAMURA, H., AND DOHI, T. Unification of software reliability models using markovian arrival processes. In *Dependable Computing (DEPOT), 2011 IEEE 17th Pacific Rim International Symposium on* (2011), IEEE, pp. 26–27.
- [1186] OKAMURA, H., AND DOHI, T. Dynamic software rejuvenation policies in a transaction-based system under markovian arrival processes. *Performance Evaluation* 70, 3 (2013), 197–211.
- [1187] OKAMURA, H., DOHI, T., AND OZAKI, S. Software reliability growth model with normal failure time distributions and its parameter estimation. In *Quality, Reliability, Risk, Maintenance and Safety Engineering (QR2MS&E), 2011 International Conference on* (2011), IEEE, pp. 411–416.
- [1188] OKAMURA, H., DOHI, T., AND OZAKI, S. Software reliability growth models with normal failure time distributions. *Reliability Engineering & System Safety* 126 (2013), 138–144.
- [1189] OKAMURA, H., DOHI, T., SUMIYOSHI, S., AND ARAI, M. Composite dependability modelling for in-vehicle networks. In *Dependable*

- Systems and Networks Workshops (IASN-W)*, 2011. *IEEE/IFIP 13th International Conference on* (2011). IEEE, pp. 240–245.
- [200] OKAMURA, H., OKANE, Y., AND DOHI, T. A multi-factor software reliability model based on logistic regression. In *Software Reliability Engineering (ISSRE)*, 2010. *IEEE 13th International Symposium on* (2010). IEEE, pp. 31–39.
 - [201] OKAMURA, H., OKANE, Y., AND DOHI, T. Quantifying the effectiveness of testing efforts on software fault detection with a logit software reliability growth model. In *Software Assessment, 2011. Joint Conference of the 22nd Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSS&APPC&SPR&P)* (2011). IEEE, pp. 62–68.
 - [202] OKAMURA, H., KIMURA, H., AND DOHI, T. On the effect of fault removal in software testing-logistic reliability estimation approach. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (2006). IEEE, pp. 247–255.
 - [203] OKAMURA, H., BROTTKE, M., DOHI, T., AND HEDBERG, K. S. Variational bayesian approach for interval estimation of alpha-based software reliability models. In *Dependable Systems and Networks, 2007. DSN'07. 7th Annual IEEE/IFIP International Conference on* (2007). IEEE, pp. 698–707.
 - [204] OKAMURA, H., KAWAYASHI, H., AND DOHI, T. Markovian modeling and analysis of internet worm propagation. In *Software Reliability Engineering, 2005. ISSRE 2005. 16th IEEE International Symposium on* (2005). IEEE, pp. 10–pp.
 - [205] OKAMURA, H., MITSUHASHI, S., AND DOHI, T. Dependability analysis of a cloud-server software system with rejuvenation. In *Software Reliability Engineering, 2005. ISSRE 2005. Proceedings. 16th International Symposium on* (2005). IEEE, pp. 171–180.
 - [206] OKAMURA, H., MITSUHASHI, S., AND DOHI, T. Effect of preventive rejuvenation in communication network system with burst arrival. In *Automotive Distributed Systems, 2005. ISALDS 2005. Proceedings (2005)*. IEEE, pp. 151–158.
 - [207] OKAMURA, H., MITSUHASHI, A., AND DOHI, T. Ekm algorithm for discrete software reliability models: a unified parameter estimation

- method. In *High Assurance Systems Engineering, 2002. Proceedings. Eighth IEEE International Symposium on* (2004-1), IEEE, pp. 219–228.
- [208] OKAMURA, H., TATEISHI, K., AND ICHII, T. Statistical inference of computer virus propagation using non-homogeneous poisson processes. In *Software Reliability, 2007. ISSRE'07. The 18th IEEE International Symposium on* (2007), IEEE, pp. 149–158.
- [209] OKAMURA, H., TATEISHI, K., AND ICHII, T. Optimal security patch release timing under non-homogeneous vulnerability-discovery processes. In *Software Reliability Engineering, 2008. ISSRE'08. 20th International Symposium on* (Nov 2008), pp. 120–128.
- [210] OKAMURA, H., WATANABE, Y., AND ICHII, T. Reliability robust software reliability models based on the em algorithm. In *Computer Software Engineering, 2009. Proceedings. 2009 International Symposium on* (2009), IEEE, pp. 69–78.
- [211] OKAMURA, H., WATANABE, Y., AND ICHII, T. An iterative scheme for maximum likelihood estimation in software reliability modeling. In *Software Reliability Engineering, 2009. ISSRE 2009. 19th International Symposium on* (2009), IEEE, pp. 240–250.
- [212] OKAMURA, H., OKAMURA, H., AND ICHII, T. On the effect of the order of test cases in the modified exponential software reliability growth model. In *2013 Second International Conference on Networking and Computing* (2013), IEEE, pp. 294–298.
- [213] OZDEREK, O., CHEN, G., KANDEMIR, M., AND KANARSKY, M. Compiler-directed variable latency queue size management to cope with timing problems. In *Proceedings of the International Symposium on Code Generation and Optimization* (2007), IEEE Computer Society, pp. 252–263.
- [214] PAPER, J., KIM, H.-J., SEIN, J.-H., AND HANG, J. An embedded software reliability model with consideration of hardware related software failures. In *Software Security and Reliability (SSRE), 2012 IEEE Sixth International Conference on* (2012), IEEE, pp. 207–214.
- [215] PIETRASZCZAK, R., RUSSEL, S., AND TRIVEDI, K. S. Software reliability and testing time allocation: An architecture-based approach. *Software Engineering, ACM Transactions on* 36, 3 (2010), 300–337.

- [416] PARSHLOVSKAYA, K., C. J. HILL, M., AND WANG, Z. Accuracy, scalability, and uncertainty of architecture-based software reliability: lessons learned from large empirical case studies. In *Software Reliability Engineering*, 2006. *ISREW'06: 22th International Symposium on* (2006), IEEE, pp. 197–203.
- [417] RUSSELL, K., AND DODI, T. Who solved the optimal software release problem based on markovian software reliability model? In *Circuits and Systems, 2004. ISCAS 2004*. The 2004 47th Annual Symposium on (2004), vol. 3, IEEE, pp. 11–175.
- [418] RUSSELL, K., AND DODI, T. Non-parametric predictive inference of generative rejuvenation schedule in operational software systems. In *Software Reliability, 2007. ISREW'07: The 20th Annual International Symposium on* (2007), IEEE, pp. 2–17 256.
- [419] RUSSELL, K., AND DODI, T. Non-parametric predictive inference of adaptive software rejuvenation schedule. In *Software Reliability Engineering Workshops, 2008. ISREW Workshops 2008. 20th International Conference on* (2008), IEEE, pp. 1–6.
- [420] RUSSELL, K., SHIMADA, K., AND DODI, T. Proportional intensity-based software reliability modeling with time-dependent metrics. In *Computer Software and Applications Conference, 2006. COMAPL, S4C'06. 20th Annual International* (2006), vol. 1, IEEE, pp. 308–370.
- [421] RUSSENDAL, R., AND MÅLSTAD, S. Refining reliability estimation of mobile software systems. In *Proceedings of the 1st international workshop on Software architectures and modeling* (2008), ACM, pp. 39–41.
- [422] SAMP, D., AND FROST, M. D. An experimental evaluation of concurrent testing during development. In *ACM SOFTWARE Engineering Notes* (2004), vol. 29, ACM, pp. 76–89.
- [423] SÄRNIT, C., JONASSEN, A., SUN, S., AND NACHTHAGEN, S. Profiling the operational behavior of os device drivers. *Accepted Software Engineering* 15, 1 (2010), 110–122.
- [424] SAIO, S., AND THIRIAN, K. S. Accurate and efficient stochastic reliability analysis of composite services using their compact markov reward model representations. In *Service Computing, 2007. SSCS 2007. IEEE International Conference on* (2007), IEEE, pp. 113–121.

- [225] SCHRIYER, A., ZEMKEDEBANK, T., AND ZELBER, A. Predicting component failures at design time. In *Proceedings of the 2005 ACM/IEEE International Symposium on Integrated software engineering* (2005), ACM, pp. 18–27.
- [226] SHARMA, V. S., AND TALWAR, P. Stabilisation under quality metrics for software products. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (2006), IEEE, pp. 45–51.
- [227] SHARMA, V. S., AND THIRYANI, K. S. Architecture based analysis of performance, reliability and security of software systems. In *Proceedings of the 26th international workshop on Software and performance* (2005), ACM, pp. 217–227.
- [228] SHARMA, V. S., AND THIRYANI, K. S. Reliability and performance of component based software systems with restarts, retries, retries and exits. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (2006), IEEE, pp. 299–310.
- [229] SHARMA, V. S., AND THIRYANI, K. S. Quantifying software performance, reliability and security: An architecture-based approach. *Journal of Systems and Software* 80, 4 (2007), 498–519.
- [230] SHIRAWA, K., RISHARA, K., AND DORA, T. Metrics-based software reliability models using non-homogeneous poisson processes. In *Software Reliability Engineering, 2006. ISSRE'06. 17th International Symposium on* (2006), IEEE, pp. 52–61.
- [231] SHIRAWA, K., RISHARA, K., AND DORA, T. Piess: Proportional intensity-based software reliability assessment tool. In *Dependable Computing, 2007. DEPEC'2007. 10th Pacific Asia International Symposium on* (2007), IEEE, pp. 43–52.
- [232] SU, Y.-S., AND HONG, C.-Y. Neural-network-based approaches for software reliability estimation using dynamic weighted combination models. *Journal of Systems and Software* 80, 4 (2007), 606–615.
- [233] SU, Y.-S., HONG, C.-Y., CHEN, Y.-S., AND CHEN, J.-K. An artificial neural-network-based approach to software reliability assessment. In *TECHNICAL 2005 2005 IEEE Region 10* (2005), IEEE, pp. 1–8.
- [234] STRICK, H., DORA, T., KATA, N., AND THIRYANI, K. S. Maximizing internal reliability in operational software systems with rejuvenation.

- In *Software Reliability Engineering, 2003. ISSRE 2003. 14th International Symposium on* (2003), IEEE, pp. 479–490.
- [237] TAKAHARA, Y., AND YAMADA, S. Comparison of software reliability assessment methods for open source software. In *Frontier and Distributed Systems, 2005. Proceedings. 11th International Conference on* (2005), vol. 2, IEEE, pp. 488–492.
 - [238] TAKAHARA, Y., AND YAMADA, S. A flexible stochastic differential equation model in distributed development environment. *European Journal of Operational Research* 168, 1 (2006), 143–152.
 - [239] TAKAHARA, Y., AND YAMADA, S. A method of user-oriented reliability assessment for open source software and its applications. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on* (2006), vol. 3, IEEE, pp. 2185–2190.
 - [240] TAKAHARA, Y., AND YAMADA, S. Software reliability assessment and optimal testing-upgrade problems for open source software. In *Systems, Man and Cybernetics, 2007. SMC'07. IEEE International Conference on* (2007), IEEE, pp. 1842–1848.
 - [241] TAKAHARA, Y., AND YAMADA, S. Software reliability growth model based on stochastic differential equations for open source software. In *Microsystems, 2007/2007 4th IEEE International Conference on* (2007), IEEE, pp. 1–6.
 - [242] TAKAHARA, Y., AND YAMADA, S. Comparison of software reliability assessment methods based on deterministic chaos theory for an open source software. In *Systems, Man and Cybernetics, 2008. SMC'08. IEEE International Conference on* (2008), IEEE, pp. 3606–3611.
 - [243] TAKAHARA, Y., AND YAMADA, S. A method of reliability assessment based on deterministic chaos theory for an open source software. In *System Science Information and Reliability Improvement, 2008. SSI/IRI'08. Second International Conference on* (2008), IEEE, pp. 60–65.
 - [244] TAKAHARA, Y., AND YAMADA, S. Software reliability analysis with optimal release problems based on hazard rate model for an embedded system. In *Systems, Man and Cybernetics (SMC), 2010 IEEE International Conference on* (2010), IEEE, pp. 720–725.

- [242] TAMURA, Y., AND YAMADA, S. Dependability analysis and optimal maintenance problem for open source cloud computing. In *Systems, Man, and Cybernetics (SMC)*, 2012. *2012 IEEE International Conference on* (2012). IEEE, pp. 1592–1597.
- [243] TAMURA, Y., YAMADA, S., AND KIMURA, M. A reliability assessment tool for distributed software development environment based on java and j/Trak. *European journal of operational research* 175, 1 (2006), 435–445.
- [244] THOMAS, N., DU HERRON, J., AND XIE, H. Parameterized unit testing: Theory and practice. In *Software Engineering, 2019. ICSE/FASE Joint International Conference on* (2019), vol. 2, IEEE, pp. 428–441.
- [245] TOMIYAMA, S., AND DORN, F. Risk-based intelligent software release planning. In *Wayside: Artificial Intelligence in Computing and 9th International Conference on Autonomous & Trusted Computing (AITC/AITC)*, 2012 9th International Conference on (2012). IEEE, pp. 240–247.
- [246] TORIENO, K., AND YAMADA, S. Markovian software-reliability measurement with a geometrically decreasing percent debugging rate. *Mathematical and computer modelling* 38, 11 (2003), 1443–1451.
- [247] TORIENO, K., AND YAMADA, S. Dynamic performance analysis for software system considering real-time property in case of nhpp task arrival. In *Secure Systems Integration and Reliability Engineering, 2008. SSIRE/ISRE. Second International Conference on* (2008). IEEE, pp. 73–80.
- [248] TORIENO, K., AND YAMADA, S. User-perceived software service availability modeling with reliability growth. In *Service Availability*, Springer, 2008, pp. 75–83.
- [249] TORIENO, K., AND YAMADA, S. Markovian model for user-perceived software service availability measurement with operation-oriented restoration. In *Service Systems and Service Management (ICSSSM)*, 2019 9th International Conference on (2019). IEEE, pp. 1–6.
- [250] TORIENO, K., AND YAMADA, S. Customer-oriented performance reliability modeling for hardware-software systems. *Reliability, IEEE Transactions on* 60, 1 (2011), 171–176.

- [252] TOSUN, S., MASSOUDI, N., ARVAS, E., KANDAMUR, M., KIM, Y., AND HUNG, W.-L. Reliability-centric hardware/software co-design. In *Guiding of Electronic Design*, 2005, 25Q&D 2005, *25Q&D International Symposium on* (2005), IEEE, pp. 375–380.
- [253] TRIVEDI, K., CLAUDIO, C., DASARATHY, B., GUERIN, M., MAYRAS, R., RUSCH, A., AND VASUDEV, B. Achieving and asserting high availability. In *Service Availability*, Springer, 2008, pp. 20–25.
- [254] TRIVEDI, K., WANG, D., THOR, D. J., RUSCH, A., SHERR, W. B., AND VASUDEV, B. Availability modeling of stp protocol on ibm® webSphere®. In *Dependable Computing*, 2008, *dependable computing Pacific Asia International Symposium on* (2008), IEEE, pp. 328–330.
- [255] TRIVEDI, K. S. Sharp 2002: Symbolic hierarchical automated reliability and performance evaluator. In *Dependable Systems and Networks*, 2002, *DSN 2002. Proceedings. International Conference on* (2002), IEEE, p. 541.
- [256] TRIVEDI, K. S. Srept: A tool for software reliability estimation and prediction. In *Dependable Systems and Networks*, 2002, *DSN 2002. Proceedings. International Conference on* (2002), IEEE, p. 546.
- [257] TRIVEDI, K. S., KIM, D. S., AND CHUGH, R. Resilience in computer systems and networks. In *Proceedings of the 2002 International Conference on Computer-Aided Design* (2002), ACM, pp. 74–77.
- [258] TRIVEDI, K. S., MANNABATHI, R., KIM, D. S., GUERIN, M., AND SAMBAS, M. Recovery from failures due to misbehaviors in it systems. In *Dependable Computing (Depend), 2007, DSN 2007 Pacific Asia International Symposium on* (2007), IEEE, pp. 224–233.
- [259] TRIVEDI, K. S., RAMANI, S., AND PRADH, R. Recent advances in modeling response-time distributions in real-time systems. *Proceedings of the IACW 94*, 7 (2003), 1023–1037.
- [260] TRIVEDI, K. S., WANG, D., AND HUNG, J. Computing the number of calls dropped due to failures. In *Software Availability Engineering (SAENG), 2009, 25Q&D 2009 International Symposium on* (2009), IEEE, pp. 11–20.
- [261] TSAI, T.-H., HUANG, C.-Y., AND CHANG, J.-H. A study of applying extended pte technique to software testability analysis. In *Computer*

- Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International (2009)*, vol. 1. IEEE, pp. 89–98.
- [262] UEMOTO, S., OZEE, T., AND OKAMURA, H. Software testing-resource allocation with operational profile. In *Proceedings of the 37th Annual ACM Symposium on Applied Computing (2012)*, ACM, pp. 1203–1206.
 - [263] VAIDYANATHAN, K., SELVASUBBU, D., AND THIRUVAL, K. S. Analysis of inspection-based preventive maintenance on operational software systems. In *Reliable Distributed Systems, 2002. Proceedings. 21st Annual Symposium on (2002)*, IEEE, pp. 280–295.
 - [264] VAIDYANATHAN, K., AND THIRUVAL, K. S. A comprehensive model for software rejuvenation. *Dependable and Secure Computing, 2003. Proceedings on 2, 2 (2003)*, 123–137.
 - [265] WANG, D., XIE, W., AND THIRUVAL, K. S. Performance analysis of clustered systems with rejuvenation under varying workload. *Performance Evaluation 64, 4 (2007)*, 247–265.
 - [266] WANG, L., XU, P., AND LI, B. The design of 1553b bus fault injection device and its implementation for bit verification. In *Aviation Control and Electronics Engineering (ACEEE), 2012 International Conference on (2012)*, IEEE, pp. 1525–1528.
 - [267] WANG, S., WU, Y., LI, M., AND LI, H. Software reliability accelerated testing method based on test coverage. In *Reliability and Maintainability Symposium (RAMS), 2011 Proceedings Annual (2011)*, IEEE, pp. 1–7.
 - [268] WANG, S., WU, Y., LI, M., AND LI, H. Software reliability modeling method based on test coverage. In *Reliability, Maintainability and Safety (ICRAMS), 2011 9th International Conference on (2011)*, IEEE, pp. 665–671.
 - [269] WILLIAMS, L., KIRKLANDS, G., AND NAGARAJ, N. On the effectiveness of unit test automation at microsoft. In *Software Reliability Engineering, 2009. ISSRE'09. 2009 International Symposium on (2009)*, IEEE, pp. 81–89.
 - [270] WU, Y., ZHANG, Y., AND LI, M. Software reliability accelerated testing method based on ordered testing. In *Reliability and Maintain-*

- reliability Specifications (RELIRES)*. 2010. *Proceedings* (2010). IEEE, pp. 1–6.
- [271] XIANG, X., AND DORE, T. On equilibrium-distribution properties in software reliability modeling. In *Acoustatics, Reliability and Security*, 2009. *ARES'09. International Conference on* (2009). IEEE, pp. 158–169.
 - [272] XIANG, X., AND DORE, T. Wavelet-based approach for estimating software reliability. In *Software Reliability Engineering*, 2009. *ISSRE'09, 20th International Symposium on* (2009). IEEE, pp. 11–20.
 - [273] XIANG, X., AND DORE, T. Estimating software intensity function via multiscale analysis and its application to reliability assessment. In *Dependable Computing (DEPOT)*, 2012. *2012 IEEE 72th Pacific Rim International Symposium on* (2012). IEEE, pp. 10–19.
 - [274] XIANG, X., AND DORE, T. Robust wavelet shrinkage estimation without data transform for software reliability assessment. In *Software Security and Reliability (SSR'12)*, 2012. *IEEE Sixth International Conference on* (2012). IEEE, pp. 215–224.
 - [275] XIANG, X., XIE, T., TILLMANN, N., AND DE HALLACK, J. Process identification of problems for structural test generation. In *Proceedings of the 33rd International Conference on Software Engineering* (2011). ACM, pp. 611–620.
 - [276] XIE, T., TILLMANN, N., DE HALLACK, J., AND SCHWARTZ, W. Future of developer testing: Building quality in code. In *Proceedings of the 2006/SPL workshop on Future of software engineering research* (2006). ACM, pp. 415–420.
 - [277] XIE, T., AND ZHANG, J. Perspectives on automated testing of aspect-oriented programs. In *Proceedings of the 3rd workshop on Testing aspect-oriented programs* (2007). ACM, pp. 7–12.
 - [278] XIE, Y., LI, L., KANUMURI, M., VIJAYAKRISHNAN, N., AND DAVIS, M. J. Reliability-aware co-synthesizers for embedded systems. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* 49, 1 (2007), 87–99.
 - [279] XING, F., CHEN, P., AND FORT, M. H. A novel method for early software quality prediction based on support vector machine. In *Software*

- Reliability Engineering*, 2005. *ISSRE 2005. 16th IEEE International Symposium on* (2005). IEEE, pp. 10–19.
- [280] YANG, S., LIU, M., LIU, B., AND HAO, B. A fault diagnosis model for embedded software based on fuzzy/lin and bayesian network. In *Reliability, Maintainability and Safety, 2009. FORMS 2009. 9th International Conference on* (2009). IEEE, pp. 779–782.
 - [281] YIN, Y., LIU, B., AND NI, H. Real-time embedded software testing method based on extended finite state machine. *Systems Engineering and Electronics, Journal of*, 22, 2 (2012), 276–280.
 - [282] YIN, Y., LIU, B., AND ZHANG, C. On hardware oriented embedded software testing development environment. In *Reliability, Maintainability and Safety, 2009. FORMS 2009. 9th International Conference on* (2009). IEEE, pp. 708–712.
 - [283] YIN, X., YUAN, D., XIE, Y., PASARIC, S., AND BASKAKI-STEFANOVA, L. How do firms become happy? In *Proceedings of the 10th ACM SIGSOFT symposium and the 10th European conference on Foundations of software engineering* (2011). ACM, pp. 28–36.
 - [284] YUAN, D., XIE, Y., FANG, S., XIE, Y., AND SAVAN, S. Improving software debuggability via log enhancement. *ACM Transactions on Computer Systems (TOCS)*, no. 1 (2012), 1.
 - [285] XIE, Y., AND LI, M. Application research of software reliability testing. In *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on* (2012), vol. 2. IEEE, pp. 507–509.
 - [286] ZHANG, H., NELSON, A., AND MENZIES, T. On the value of learning from defect-dense components for software defect prediction. In *Proceedings of the 9th International Conference on Predictive Models in Software Engineering* (2010). ACM, p. 14.
 - [287] ZHANG, Y., LIU, B., AND XIE, Q. A dynamic software binary fault injection system for real-time embedded software. In *Reliability, Maintainability and Safety (FORMS), 2013 9th International Conference on* (2013). IEEE, pp. 678–680.
 - [288] ZHANG, Y., XIE, X., AND LI, M. R. Effortful: A dynamic fault-tolerance framework for voluntary-resource cloud computing. In

Closed Computing (CLOUD), 2011 IEEE International Conference on (2011), IEEE, pp. 444–451.

- [288] ZHANG, J., JIN, Y., TAYLOR, K. S., AND MATIAS JR., H. Injecting memory leaks to accelerate software failures. In *Software Reliability Engineering (ISSRE)*, 2011 IEEE 22nd International Symposium on (2011), IEEE, pp. 269–280.
- [289] ZHANG, Y., YIN, X., KASSI, R., AND TAYLOR, K. S. A review of the research on quantitative reliability prediction and assessment for electronic components. In *Prognostics and Systems Health Management Conference (PHM-Schenectady)*, 2011 (2011), IEEE, pp. 1–7.
- [290] ZHANG, J., WILLIAMS, J., AND HENNESSY, H. Pathfind automation to support regression test selection for cost-based applications. In *Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering (2007)*, ACM, pp. 241–253.
- [291] ZHANG, X., AND LYU, M. R. A distributed replication strategy evaluation and selection framework for fault-tolerant web services. In *Web Services, 2008. ICWS'08. IEEE International Conference on* (2008), IEEE, pp. 145–152.
- [292] ZHANG, Z., AND LYU, M. R. A qos-aware middleware for fault-tolerant web service simulation approach to structure-based software reliability analysis. In *Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on* (2008), IEEE, pp. 97–106.
- [293] ZHANG, Z., AND LYU, M. R. Webstream: A distributed reliability assessment mechanism for web services. In *Dependable Systems and Networks With FTCS and FACS, 2008. DSN 2008. IEEE International Conference on* (2008), IEEE, pp. 302–307.
- [294] ZHANG, X., AND LYU, M. R. Collaborative reliability prediction of service-oriented systems. In *Proceedings of the 32nd ACM/ANZIS International Conference on Software Engineering-Volume 1* (2010), ACM, pp. 35–43.
- [295] ZHANG, X., AND LYU, M. R. Personalized reliability prediction of web services. *ACM Transactions on Software Engineering and Life Sciences (TOSELS)*, vol. 2 (2013), 12.

- [497] ZHANG, Z., ZHOU, T.-C., LYU, M. R., AND KISS, I. Picking: A component picking framework for fault-tolerant cloud applications. In *Software Reliability Engineering (ISSRE)*, 2009 *IEEE 23rd International Symposium on* (2009), IEEE, pp. 398–407.
- [498] ZHOU, D., LIU, D., RAN, L., AND WU, Y. The domain analysis and design of system-testing equipment for software-intensive systems. In *Antarescon, 2006 JAVAS (2006)*, IEEE, pp. 523–529.
- [499] KUMARAN, T., NAGARAJ, N., HANIG, K., PRASAD, R., AND WILKINS, L. An empirical study on the relation between dependency neighborhoods and failures. In *Software Testing, Verification and Validation (ICST)*, 2017 *IEEE 36th International Conference on* (2017), IEEE, pp. 347–358.
- [500] KUMARAN, T., AND NAGARAJ, N. Predicting subsystem failures using dependency graph complexities. In *Software Reliability, 2007. ISSRE'07. The 18th IEEE International Symposium on* (2007), IEEE, pp. 227–236.

Lista de *Keyword Groups*

- ❑ KG1 = {analytical models};
- ❑ KG2 = {application software, software packages, software tools};
- ❑ KG3 = {optimization methods, optimization algorithms};
- ❑ KG4 = {computational modeling};
- ❑ KG5 = {computer architecture};
- ❑ KG6 = {debugging, imperfect debugging, program debugging};
- ❑ KG7 = {software rejuvenation, aging, degradation, software life-cycle};
- ❑ KG8 = {fault detection, fault minimization, fault correction, fault tolerance};
- ❑ KG9 = {statistical distributions};
- ❑ KG10 = {software architecture};
- ❑ KG11 = {EM algorithm};
- ❑ KG12 = {Markov processes};
- ❑ KG13 = {modeling frameworks};
- ❑ KG14 = {component-based software system, modular software systems};
- ❑ KG15 = {NHPP, non-homogeneous Poisson process, two-dimensional NHPP, two-dimensional stochastic processes, stochastic process};
- ❑ KG16 = {resource allocation, optimal testing resource allocation};
- ❑ KG17 = {Poisson process};
- ❑ KG18 = {preventive maintenance};

- ❑ KG19 = {testing effort, software testing, integration testing, program testing, goodness-of-fit test, software reliability accelerated testing};
- ❑ KG20 = {queuing model, queuing analysis};
- ❑ KG21 = {regression testing, regression tree analysis};
- ❑ KG22 = {change point};
- ❑ KG23 = {software maintenance};
- ❑ KG24 = {estimation methods, parameter estimation};
- ❑ KG25 = {software measurement, software reliability methods, software reliability assessment, software reliability model};
- ❑ KG26 = {failure mode, failure modes and effects analysis};
- ❑ KG27 = {testing coverage};
- ❑ KG28 = {fault injection};
- ❑ KG29 = {software reliability growth model};
- ❑ KG30 = {web services, web applications, web server};
- ❑ KG31 = {failure analysis};
- ❑ KG32 = {fault trees, fault tree analysis};
- ❑ KG33 = {availability};
- ❑ KG34 = {predictive models};
- ❑ KG35 = {software performance, performance evaluation};
- ❑ KG36 = {embedded systems};
- ❑ KG37 = {software quality};
- ❑ KG38 = {software metrics};
- ❑ KG39 = {fault correlation, fault dependency};
- ❑ KG40 = {Bayesian approach, variational Bayes, Bayes}.

Acrônimos das Instituições

- ❑ ASUS (ASUSTeK Computer Inc);
- ❑ BA (Bank of America);
- ❑ BCP (Business Cube Partners);
- ❑ BELL (Bell Labs);
- ❑ BNU (Beijing Normal University);
- ❑ BUAA (Beihang University);
- ❑ CHTTL (Telecommunication Laboratories Chungwha Telecom);
- ❑ CISCO (Cisco Systems);
- ❑ CUHK (Chinese University of Hong Kong);
- ❑ DUKE (Duke University);
- ❑ HRBEU (Harbin Engineering University);
- ❑ HKUST (Hong Kong University of Science and Technology);
- ❑ HSU (Hiroshima Shudo University);
- ❑ HU (Hiroshima University);
- ❑ HUB (Humboldt University of Berlin);
- ❑ IBIS (Ibis Therapeutics);
- ❑ IBM (IBM Research);
- ❑ IITK (Indian Institute of Technology Kanpur);
- ❑ UI (Illinois University);

- ❑ INER (Institute of Nuclear Energy Research);
- ❑ LYIT (Lan Yang Institute of Technology);
- ❑ MSC (Microsoft Corporation);
- ❑ MSR (Microsoft Research);
- ❑ NCSU (North Carolina State University);
- ❑ NCTU (National Chiao Tung University);
- ❑ NEC (NEC Corporation);
- ❑ NTHU (National Tsing Hua University);
- ❑ NTUT (National Taipei University of Technology);
- ❑ NZU (Nanzan University);
- ❑ PDM (Politecnico di Milano);
- ❑ PSU (Pennsylvania State University);
- ❑ RSC (Relex Software Corporation);
- ❑ SLU (Saarland University);
- ❑ SU (Stanford University);
- ❑ TRL (TR Labs);
- ❑ TT (Telcordia Technologies);
- ❑ TU (Tottori University);
- ❑ TUD (Technische Universität Darmstadt);
- ❑ UConn (University of Connecticut);
- ❑ UC (University of California);
- ❑ UFPE (Federal University of Pernambuco);
- ❑ UFSC (Federal University of Santa Catarina);
- ❑ UNIME (University of Messina);
- ❑ UNF (University of Naples Federico II);
- ❑ UT (University of Twente);

- ❑ UZH (University of Zurich);
- ❑ VU (VU University Amsterdam);
- ❑ WSTI (WebSphere Technology Institute)

Veículos de Publicação

Esta seção apresenta as listas dos veículos onde foram publicados os estudos utilizados neste trabalho. Esses veículos foram divididos em duas listas: a lista de conferências e a de periódicos. Cada uma dessas listas está organizada conforme a classificação obtida pelos veículos no extrato Qualis de 2014.

D.1 Lista de Conferências

Conferências Qualis A1:

- CCS ACM Conference on Computer and Communications Security
- PLDI ACM SIGPLAN conference on Programming Language Design and Implementation
- FSE ACM SIGSOFT Conference on the Foundations of Software Engineering
- ICSE ACM/IEEE International Conference on Software Engineering
- ACC American Control Conference
- DATE Design, Automation and Test in Europe Conference and Exhibition
- ICWS IEEE International Conference on Web Services
- ASE IEEE/ACM International Conference on Automated Software Engineering
- ASPLOS International Conference on Architectural Support for Programming Languages and Operating Systems
- AOSD International Conference on Aspect-Oriented Software Development
- ICCAD International Conference on Computer-Aided Design
- DSN International Conference on Dependable Systems and Networks
- ICSE International Conference on Software Engineering
- WWW International World Wide Web Conference

Conferências Qualis A2:

ASP-DAC Asia and South Pacific Conference on Design Automation

COMPSAC International Computer Software and Applications Conference

ISCC International Symposium on Computers and Communications

ISSRE International Symposium on Software Reliability Engineering

Conferências Qualis B1:

SCC IEEE International Conference on Services Computing

HASE IEEE International Symposium on High Assurance Systems Engineering

ICPADS International Conference on Parallel and Distributed Systems

Conferências Qualis B2:

CLOUD IEEE International Conference on Cloud Computing

ITRE International Conference on Information Technology: Research and Education

ICST International Conference on Software Testing, Verification and Validation

UIC International Conference on Ubiquitous Intelligence and Computing

Conferência Qualis B3:

DASC IEEE International Symposium on Dependable, Autonomic and Secure Computing

Conferência Qualis B4:

ICETC International Conference on Education Technology and Computer

QSIC International Conference on Quality Software

SSIRI International Conference on Secure Software Integration and Reliability Improvement

Conferências sem Qualis:

DASC Digital Avionics Systems Conference

AUTORESTCON IEEE Autotestcon

ICMIT IEEE International Conference on Management of Innovation and Technology

ASAP IEEE International Conference on Application-Specific Systems, Architectures and Processors

CSAE IEEE International Conference on Computer Science and Automation Engineering

ICM IEEE International Conference on Mechatronics

TENCON IEEE Tencon

ACTEA International Conference on Advances in Computational Tools for Engineering Applications

ATC International Conference on Autonomic Trusted Computing

ARES International Conference on Availability, Reliability and Security

ICICEE International Conference on Industrial Control and Electronics Engineering

ICNC International Conference on Networking and Computing

ICPE International Conference on Performance Engineering

ICQR2MSE International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering

ICRMS International Conference on Reliability, Maintainability and Safety

ISAS International conference on Service availability

ICSSSM International Conference on Service Systems and Service Management

SERE International Conference on Software Security and Reliability

SMC International Conference on Systems, Man and Cybernetics

PHM-Shenzhen Prognostics and System Health Management Conference

WAC World Automation Congress

D.2 Lista de Periódicos**Periódicos A1:**

0377-2217 European Journal of Operational Research

0018-9340 IEEE Transactions on Computers
0098-5589 IEEE Transactions on Software Engineering
1083-4427 IEEE Transactions on Systems, Man and Cybernetics
0925-2312 Journal Neurocomputing
0018-9219 Proceedings of the IEEE

Periódicos A2:

1049-331X ACM Transactions on Software Engineering and Methodology
0898-1221 Computers Mathematics with Applications
0018-9529 IEEE Transactions on Reliability
0377-0427 Journal Applied Mathematics and Computation
0895-7177 Journal Mathematical and Computer Modelling
1090-2724 Journal of Computer and System Sciences
0166-5316 Journal Performance Evaluation
0164-1212 The Journal of Systems and Software

Periódico B1:

1545-5971 IEEE Transactions on Dependable and Secure Computing

Periódico B2:

0951-8320 Reliability Engineering System Safety

Periódico B4:

090-7141 Journal of Computer Networks and Communications

Periódico sem Qualis:

1004-4132 Journal Systems Engineering and Electronics

—