



Universidade de Pernambuco
Escola Politécnica de Pernambuco
Departamento de Sistemas e Computação
Programa de Pós-Graduação em Engenharia da Computação

Rômulo da Silva Lima

**Estudos Empíricos em Engenharia de Software: Avaliando o
Benchmarking Framework Através de um *Survey* e Replicando
uma Avaliação de Desempenho**

Dissertação de Mestrado

Recife, Maio de 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE DE PERNAMBUCO
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

RÔMULO DA SILVA LIMA

**Estudos Empíricos em Engenharia de Software: Avaliando o
Benchmarking Framework Através de um *Survey* e Replicando
uma Avaliação de Desempenho**

Esta Dissertação é apresentada como requisito parcial para obtenção do título de Mestre em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Prof. Dr. Sérgio Castelo Branco Soares
Orientador

Recife, Maio de 2010

Dedico esse trabalho aos meus pais, exemplos de vida.

Agradecimentos

Primeiramente, quero agradecer ao Nosso Grandioso Pai, sem a Fé que tenho em Deus e sem suas bênçãos não teria chegado ao final dessa conquista. Depois, quero agradecer a minha família por toda força, motivação, dedicação, amor e incentivo que sempre me deram na minha vida e nas minhas batalhas. Ao meu pai, Edson, minha mãe, Guiomar, minha irmã, Daksuênia, muito obrigado por sempre terem acreditado em mim e me dado todo o apoio necessário para realização dos meus sonhos. A minha namorada Talitha pela cumplicidade, paciência e compreensão nos momentos difíceis, muito obrigado por está presente desde o início, me dando apoio, me orientando nas decisões.

Agradeço ao meu orientador, Sérgio Soares, que sempre esteve disponível, ajudando e orientando, transmitindo seu exemplar conhecimento, competência e presença em todas as horas do meu mestrado desempenhando um papel importante na minha formação acadêmica. Agradeço também aos professores e a Georgina do Departamento de Sistemas e Computação/UPE. Agradeço ao professor Alessandro Garcia, que mesmo distante, teve uma participação no desenvolvimento deste trabalho. Ao professor Cristiano Ferraz, muito obrigado por me ajudar na estatística.

A Pitang/CESAR – Centro de Estudos e Sistemas Avançados do Recife, pelo apoio técnico e pessoal. Obrigado por me entender no momento que precisei reduzir minha jornada de trabalho, a todos os gerentes de projetos que participei durante essa fase da minha vida, Marcela Guerra, Daniela Pompílio, Andrea Gomes.

Não poderia esquecer os meus colegas pelo apoio nos trabalhos, Nara Miranda, Nadilma Nunes, Danilo Costa, virando a noite fazendo trabalho de Engenharia de Software. Ao pessoal que participou do artigo submetido, Cristiane Queiroz, Julio Taveira, Juliana Saraiva, meu muito obrigado por tudo.

Meu muito obrigado a todos que contribuíram na construção da pesquisa de opinião que elaborei. Ao professor Fernando Castor por dar sugestões sempre que questionado, ao companheiro Marcelo Moura, ao pessoal do SPG - *Software Productivity Group*, e a todos professores das várias conferências que deram sua colaboração pra o sucesso deste trabalho.

Agradeço a banca desta defesa, composta pela professora Cristine Gusmão e pelo professor Fabio Queda, muito obrigado pela aceitação do convite feito pelo professor Sérgio Soares e pelos comentários que contribuíram para a melhoria da versão final deste documento.

A todos, o meu muito obrigado, saibam que esta conquista também é de vocês.

Resumo

Estudos Empíricos na Engenharia de Software exploram a experimentação para testar teorias e para a avaliação das tecnologias de software produzidas. Por exemplo, para responder uma pergunta como: "Este mecanismo é eficaz?" algumas medições deste mecanismo precisam ser levadas em consideração. Não há informação de valor ao dizer que uma técnica é "boa". Em vez disso, é essencial medir os vários atributos do mecanismo para que se possa avaliar de forma adequada, em geral comparando se a técnica é mais ou menos eficaz do que a outra e, eventualmente, em que situações isso acontece. Dependendo do propósito da avaliação e das condições para investigação, existem pelo menos três tipos de estratégias de investigação que podem ser utilizadas: experimentos controlados, estudos de caso e *surveys*. Os *surveys* se destacam pela popularidade e abrangência de utilização. Outro tipo de estratégia usada em Engenharia de Software são as revisões sistemáticas, ou seja, quando se deseja integrar os resultados provenientes de diversos estudos experimentais correlatos. O planejamento e o processo propostos por cada investigação são de fundamental importância para obter sucesso nos estudos experimentais. Devido à carência de fontes confiáveis que possibilitem responder de forma satisfatória a questões na Engenharia de Software, frequentemente decisões equivocadas são tomadas, e pessoas acabam investindo tempo e dinheiro em técnicas que falham. Entretanto, respostas para essas questões podem ser obtidas de forma razoável caso seja feito um uso intenso e sistemático de abordagens baseadas em evidências experimentais. Este trabalho busca evidências empíricas, por meio da execução de um *survey*, para avaliar uma proposta de classificação de aplicações e cenários de manutenção de software orientado a aspectos proposto por Moura [1]. Outro objetivo do trabalho é a replicação de uma avaliação de desempenho de um método de controle de concorrência, feito há quase 10 anos, onde se pretende apresentar as informações que devem ser fornecidas no relatório de uma replicação e as dificuldades que os pesquisadores enfrentam quando se pretende replicar um experimento. Além disso, esta dissertação apresenta passos e considerações para auxiliar na decisão de qual estratégia de investigação utilizar.

Palavras-chave: Estudos Empíricos, *Survey*, Desenvolvimento de Software Orientado a Aspectos, Manutenibilidade de Software, Replicação Empírica.

Abstract

Empirical Studies in Software Engineering explore experimentation to test theories and to assess produced software technologies. For instance, to answer a question like: "Is such mechanism effective?" some measurements of this mechanism must be considered. There is no information only saying that a technique is "good". Instead, it is essential to measure each attribute of the mechanism, so that an adequate assessment is performed, in general saying that a technique is more or less effective than another, and eventually in which contexts that happens. Depending on the evaluation purpose and the research conditions, there are at least three types of research strategies that can be used: controlled experiments, case studies and surveys. Surveys are highlighted by popularity and scope of use. Another type of strategy used in Software Engineering are systematic reviews, or when you wish to integrate results from several related experimental studies. The planning and the process proposed by each experimental research are of fundamental importance to achieve success in the experimental study. Due to the lack of reliable sources to enable satisfactorily answer to software engineering questions, often wrong decisions are made, and people end up investing effort and money on techniques that fail. However, answers to those questions might be obtained in a reasonable manner if an intensive and systematic approach based on experimental evidence is carried out. This work seeks empirical evidence, through the implementation of a survey, to evaluate a classification of aspect-oriented applications and maintenance scenarios proposed by Moura [1]. Another aim is the replication of a concurrency control method performance evaluation, done nearly 10 years ago, where the intention is to present the information to be provided in the report of a replication the difficulties that researchers face when trying to replicate an experiment. Besides, this dissertation presents the steps and considerations to support deciding which research strategy to use.

Keywords: Empirical Study support, Survey, Aspect-Oriented Software Development, Maintainability Studies, Empirical Replication.

Sumário

Lista de Símbolos e Siglas	ix
Índice de Figuras	x
Índice de Tabelas	xi
1 Introdução	12
1.1 Contexto do Problema e Motivação.....	12
1.2 Objetivos.....	14
1.3 Organização da dissertação.....	14
2 Fundamentação Teórica	15
2.1 Estudos Empíricos na Engenharia de Software.....	15
2.2 Seleção da Estratégia Empírica.....	17
2.2.1 Questionamentos.....	18
2.2.2 Posição Filosófica.....	20
2.2.3 Objetivo do Estudo.....	20
2.2.4 Estratégias Empíricas.....	21
2.3 Processo de Experimentação em Engenharia de Software.....	24
2.4 Replicação em Estudos Empíricos.....	25
2.4.1 Replicações Publicadas.....	26
2.4.2 Guia para Relatar Replicações [23].....	27
2.5 Conclusão.....	29
3 Pesquisa Survey	30
3.1 Seleção da Estratégia Empírica.....	30
3.1.1 O Benchmarking Framework.....	31
3.2 Survey (Pesquisa de Opinião).....	34
3.3 Survey na Internet.....	36
3.4 Planejamento de um Survey.....	37
3.4.1 Projeto do survey.....	38
3.4.2 Descrição do projeto.....	38
3.4.3 Projeto Experimental.....	38
3.4.4 Taxas de respostas.....	39
3.4.5 Seleção das questões.....	40
3.4.6 Formato do questionário.....	40
3.5 Processos de aplicação de Survey.....	41
3.5.1 Processo proposto por WOHLIN [9].....	42
3.5.2 Processo proposto por PFLEGEER e KITCHENHAM [24].....	42
3.5.3 Detalhamento do processo proposto por MENDONÇA [25].....	43
1. Definição dos objetivos.....	44
2. Planejamento da pesquisa de opinião.....	44
3. Projeto dos instrumentos da pesquisa de opinião.....	46
4. Validação da instrumentação.....	47
5. Execução da pesquisa de opinião.....	47
6. Análise dos dados.....	48
7. Empacotamento da pesquisa de opinião.....	48

3.6	Grau de relevância experimental.....	49
3.7	Conclusão.....	49
4	Uma Avaliação Empírica do <i>Benchmarking Framework</i>.....	51
4.1	Definição do Estudo Experimental.....	51
4.1.1	Objetivo.....	52
4.2	Planejamento.....	52
4.2.1	Seleção do contexto.....	52
4.2.2	Seleção dos Participantes.....	53
4.2.3	Descrição da Instrumentação.....	53
4.2.4	Definição da Avaliação.....	54
4.3	Ameaças a Validade.....	55
4.3.1	Validade de Face.....	56
4.3.2	Validade de Conclusão.....	56
4.3.3	Validade de Construção.....	56
4.3.4	Validade Externa.....	57
4.4	Operação do Estudo Experimental.....	57
4.4.1	Avaliação do Estudo Experimental.....	57
4.5	Lições Aprendidas.....	62
4.6	Conclusão.....	63
5	Replicação de uma Avaliação de Desempenho.....	64
5.1	Informações sobre o Estudo Original.....	64
5.1.1	Questionamentos.....	67
5.1.2	Participantes.....	67
5.1.3	Projeto.....	67
5.1.4	Artefatos.....	68
5.1.5	Variáveis de Contexto.....	68
5.1.6	Resumo dos Resultados.....	68
5.2	Informações sobre a Replicação.....	69
5.2.1	Motivação para a Realização da Replicação.....	69
5.2.2	Nível de Interação com o Pesquisador do Estudo Original.....	69
5.2.3	Mudanças no Estudo Original.....	69
5.3	Comparação dos Resultados.....	71
5.4	Lições Aprendidas.....	79
5.5	Conclusão.....	80
6	Conclusão e Trabalhos Futuros.....	82
6.1	Conclusões.....	82
6.2	Trabalhos Relacionados.....	84
6.3	Trabalhos Futuros.....	85
	Bibliografia.....	86
	Anexo 1 – Questionário Utilizado no <i>Survey</i>.....	90
	Anexo 2 – Resumo da Análise dos Dados do <i>Survey</i>.....	97

Lista de Símbolos e Siglas

DSOA – Desenvolvimento de Software Orientado a Aspectos

SGBD - Sistema de Gerenciamento de Banco de Dados

BF – *Benchmarking Framework*

OA - Orientação a Aspectos

GQM - *Goal/Question/Metric*

OO – Orientação a Objetos

AOSD - *Aspect-Oriented Software Development*

LA-WASP - *Latin American Workshop on Aspect-Oriented Software Development*

OOPSLA - *Object Oriented Programming, Systems, Languages and Applications*

ECOOP - *European Conference on Object-Oriented Programming*

ICSE - *International Conference on Software Engineering*

JDK - *Java Development Kit*

RAM - *Random Access Memory*

Índice de Figuras

FIGURA 2.1 PASSOS PARA SELECIONAR UMA ESTRATÉGIA EMPÍRICA	18
FIGURA 2.2 OS CONCEITOS DE UM EXPERIMENTO [9]	22
FIGURA 2.3 PROCESSO DE EXPERIMENTAÇÃO ADAPTADO [9].....	24
FIGURA 2.2 PROCESSO DEFINIDO PELO BENCHMARKING FRAMEWORK [1]	32
FIGURA 3.2 PROCESSO DE EXECUÇÃO DE EXPERIMENTOS [9].....	42
FIGURA 5.1 IMPACTO DE DIFERENTES TRATAMENTOS DE CONCORRÊNCIA [4].....	73
FIGURA 5.2 IMPACTO DE DIFERENTES TRATAMENTOS DE CONCORRÊNCIA NA REPLICAÇÃO.....	73
FIGURA 5.3 IMPACTO DE DIFERENTES TRATAMENTOS DE CONCORRÊNCIA COM MÉTODOS PESADOS [4].....	74
FIGURA 5.4 IMPACTO DE DIFERENTES TRATAMENTOS DE CONCORRÊNCIA COM MÉTODOS PESADOS NA REPLICAÇÃO.....	74
FIGURA 5.5 IMPACTO DO <i>TIMESTAMP</i> [4]	75
FIGURA 5.6 IMPACTO DO <i>TIMESTAMP</i> NA REPLICAÇÃO	76
FIGURA 5.7 IMPACTO DO <i>TIMESTAMP</i> COM MÉTODOS PESADOS [4]	76
FIGURA 5.8 IMPACTO DO <i>TIMESTAMP</i> COM MÉTODOS PESADOS NA REPLICAÇÃO	77
FIGURA 5.9 IMPACTO DO <i>SYNCRONIZED</i> EM RELAÇÃO AO GERENCIADOR DE CONCORRÊNCIA [4]	78
FIGURA 5.10 IMPACTO DO <i>SYNCRONIZED</i> EM RELAÇÃO AO GERENCIADOR DE CONCORRÊNCIA NA REPLICAÇÃO.....	78
FIGURA 5.11 IMPACTO DO <i>SYNCRONIZED</i> EM RELAÇÃO AO GERENCIADOR DE CONCORRÊNCIA COM MÉTODOS PESADOS [4].....	79
FIGURA 5.12 IMPACTO DO <i>SYNCRONIZED</i> EM RELAÇÃO AO GERENCIADOR DE CONCORRÊNCIA COM MÉTODOS PESADOS NA REPLICAÇÃO	79

Índice de Tabelas

TABELA 2.1. PERGUNTAS EXPLORATÓRIAS [12].....	18
TABELA 2.2. PERGUNTAS DE OCORRÊNCIAS [12]	19
TABELA 2.3. PERGUNTAS RELACIONAIS E CAUSAIS [12].....	19
TABELA 2.4. COMPARAÇÃO DAS ESTRATÉGIAS EMPÍRICAS	23
TABELA 4.1. MEDIDAS PARA CATEGÓRICAS A CONCORDÂNCIA DOS DADOS OBSERVADOS [43]	55
TABELA 4.2. POSIÇÃO DOS PARTICIPANTES.....	58
TABELA 4.3. TIPOS DE ATIVIDADES EM ESTUDOS EMPÍRICOS.....	58
TABELA 4.4. RESULTADO DA AVALIAÇÃO DOS ATRIBUTOS GERAIS.....	59
TABELA 4.5. RESULTADO DA AVALIAÇÃO DOS ATRIBUTOS ORIENTADO A ASPECTOS	60
TABELA 4.6. RESULTADO DA AVALIAÇÃO DOS CENÁRIOS DE MUDANÇA	60
TABELA 4.7. ATRIBUTOS GERAIS SUGERIDOS PELOS PARTICIPANTES	60
TABELA 4.8. ATRIBUTOS ORIENTADOS A ASPECTOS SUGERIDOS PELOS PARTICIPANTES.....	61
TABELA 4.9. CENÁRIOS DE MUDANÇA SUGERIDOS PELOS PARTICIPANTES.....	61
TABELA 4.10. RESULTADOS DA AVALIAÇÃO DE UM MECANISMO QUE GUIE ESTUDOS EMPÍRICOS OA	62
TABELA 5.1. COMPARAÇÃO DAS ESPECIFICAÇÕES GERAIS DOS TESTES.....	70
TABELA 5.2. COMPARAÇÃO ENTRE DIFERENTES TRATAMENTOS DE CONCORRÊNCIA	72
TABELA 5.3. COMPARAÇÃO DO IMPACTO DO MECANISMO DE <i>TIMESTAMP</i>	75
TABELA 5.4. COMPARAÇÃO DE <i>SYNCHRONIZED</i> VERSUS GERENCIADORCONCORRENCIA	77

Capítulo 1

Introdução

Este capítulo apresenta o contexto no qual este trabalho se encontra, assim como seus objetivos e os benefícios esperados. A Seção 1.1 contextualiza o problema e a motivação para este trabalho. O objetivo geral e os objetivos específicos são mostrados na Seção 1.2, enquanto a organização da dissertação é exposta na Seção 1.3.

1.1 Contexto do Problema e Motivação

Experimentação lê-se estudos empíricos, é utilizada para testar teorias e para a pesquisa, também é utilizada em casos onde a teoria e a análise dedutiva não funcionam. Experimentos provam a influência das hipóteses, eliminam alternativas de explicações para um fenômeno e descobrem novos fenômenos que precisam ser explorados. Desta forma, experimentos ajudam a indução: permitem derivar teorias a partir de observação [2].

Esta abordagem vem sendo usada na física, engenharia e medicina, e não deve ser diferente com software. A Engenharia de Software Experimental traduz a tentativa de se aplicar o método científico de experimentação ao software, buscando tornar a Engenharia de Software mais exata e precisa. Com isto, permite transferir tecnologias para a prática, adaptá-las e avaliá-las no contexto das organizações, e com base nestas experiências, melhorar e evoluir estas tecnologias para a melhoria de qualidade e produtividade ao longo do processo de desenvolvimento do software.

Empresas, organizações de softwares e a academia precisam de informações que apóiem suas decisões, como por exemplo, o uso ou não de uma determinada tecnologia para o desenvolvimento do software. Essas decisões visam atender os objetivos almejados, que podem

ser a melhora da qualidade do produto ou a redução dos custos associados ao seu desenvolvimento, por exemplo. Porém, na maioria dos casos da Engenharia de Software, a única maneira real de avaliar essas questões é colocando-as em prática, ou seja, fazendo com que pessoas utilizem a tecnologia proposta durante o desenvolvimento de um software real, para que se possa avaliá-la posteriormente. Por isso, apesar de custosas e demoradas, pesquisas científicas, através de experimentos e estudos empíricos, são de fundamental importância durante a avaliação de novas tecnologias utilizadas no desenvolvimento, e permitem que os resultados obtidos acelerem o crescimento e disseminação do conhecimento na área.

Para este trabalho é de importância fundamental apresentar os procedimentos para fazer estudos empíricos com qualidade em Engenharia de Software, ou seja, mostrar os passos a serem seguidos durante a elaboração de experimentos, *surveys* e estudos de caso de forma que fique evidente a complexidade que é encontrada quando se pretende ser rigoroso com os estudos.

A utilização de um processo com atividades bem definidas é fundamental para o sucesso de um estudo experimental. Por meio de sua estrutura pode-se acompanhar suas atividades visando o controle e melhoria das atividades que compõem o processo.

Dentre as estratégias de investigação utilizadas nos estudos empíricos, os *surveys* são apresentados com maior foco neste trabalho na avaliação de uma proposta feita por Moura [1] que teve avaliações simples no trabalho original. Os *surveys* se destacam pela popularidade e abrangência de utilização, sendo usados desde votações a definição de requisitos em software. O *survey* é um método para coletar informação de pessoas acerca de suas ideias, sentimentos, planos, bem como origem social, educacional e financeira. [3]. Entretanto, a aparente simplicidade do seu processo de aplicação ocasiona um grande número de resultados inválidos. As causas mais comuns deste tipo de problema são as deficiências no processo utilizado ou a não utilização de um processo.

Outra motivação para este trabalho é mostrar a replicação de uma avaliação de desempenho feito há 10 anos onde se pretende verificar se as conclusões permanecem as mesmas considerando novos hardwares e softwares.

1.2 Objetivos

Este trabalho propõe um estudo sobre estudos empíricos. Pretende-se com isso auxiliar pesquisadores na tomada de decisão sobre quais métodos de investigação experimental é mais adequado para o seu estudo. Para atingir o objetivo geral deste trabalho, serão apresentados com detalhes os procedimentos para a elaboração de um *survey* e a replicação de uma avaliação de desempenho, com todas as nuances e dificuldade, sendo estabelecidos os seguintes objetivos específicos:

- ✓ Discutir, planejar e executar estudos empíricos, apresentando exemplos de estudos, formas alternativas de avaliação, bem como lições aprendidas na condução de estudos empíricos na área de Engenharia de Software;
- ✓ Obter formas de avaliações para comparar a teoria apresentada por Moura [1] à realidade da experiência dos especialistas da área;
- ✓ Replicar a avaliação de desempenho feita por Soares [4], a fim de comparar os resultados há 10 anos com os de hoje;
- ✓ Oferecer um relato de experiência (lições aprendidas) sobre a execução de estudos empíricos sistemáticos.

1.3 Organização da dissertação

Esta dissertação está organizada em 6 capítulos. O Capítulo 2 expõe a fundamentação teórica sobre estudos empíricos na engenharia de software, passos para auxiliar a escolha da estratégia empírica a utilizar, o processo de experimentação e replicação em estudos empíricos. Já o Capítulo 3 executa os passos para decidir qual estratégia empírica utilizar para avaliar o Benchmarking Framework [1] e apresenta as definições e o processo para aplicação da estratégia empírica selecionada, o *survey*. O planejamento e avaliação do *survey* desenvolvido como parte deste trabalho, apresentando o estudo empírico (*survey*) utilizado para avaliar a definição inicial dos atributos de classificação e cenários de mudança conforme a literatura [1] é apresentado no Capítulo 4. Os conceitos principais da Avaliação de Desempenho e a replicação de um experimento para avaliação de desempenho de um método de controle de concorrência feito há 10 anos, são apresentadas no Capítulo 5. Por último, no Capítulo 6, são apresentadas as conclusões obtidas neste trabalho e trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica para este trabalho. Tal fundamentação é de crucial importância para o bom entendimento do restante da dissertação.

A Seção 2.1 mostra os principais conceitos de Estudos Empíricos na Engenharia de Software. Na Seção 2.2 passos para Seleção da Estratégia Empírica, o Processo de Experimentação em Engenharia de Software na Seção 2.3 e Replicação de Estudos Empíricos na Seção 2.4. Por fim na Seção 2.5 é feita uma conclusão do que foi apresentado no capítulo.

2.1 Estudos Empíricos na Engenharia de Software

Apesar da criação de uma área específica, a Engenharia de Software [5, 6, 7], que visa facilitar e controlar o desenvolvimento de sistemas de software, as atividades envolvidas nesse processo continuam fortemente dependentes da participação e criatividade humana. Esses fatores diminuem a confiabilidade de resultados científicos na área e impedem que ela se torne uma ciência exata. Por isso é necessário utilizar métodos empíricos que auxiliam a comprovação de tais resultados.

Analogamente à definição de Engenharia de Software [8], experimentações permitem de maneira sistemática, disciplinada, quantificável e controlada, avaliar conhecimentos emergentes [9]. Essa é uma das principais razões para as pesquisas empíricas serem tão comuns em área de ciência sociais e comportamentais [10], as quais estão intimamente interessadas e relacionadas ao comportamento humano. Tal característica dificulta a avaliação de resultados, pois não é possível encontrar leis da natureza, como na Física, que guiem e provem com exatidão a maioria das

hipóteses nessas áreas. Isto justifica a necessidade de estudos empíricos científicos para garantir a veracidade de tais resultados.

Estudos empíricos são importantes para pesquisadores de qualquer área, inclusive os de engenharia de software. Novos métodos, técnicas, processos, linguagens e ferramentas não devem ser apenas sugeridas, publicadas e comercializadas. Antes de qualquer adoção é extremamente necessário que novas invenções e sugestões sejam rigorosamente avaliadas e comparadas com outras já existentes, para que os riscos associados sejam bem conhecidos. A experimentação provê esta possibilidade, tornando a área pesquisada mais científica e confiável.

As metodologias são necessárias para ajudar a estabelecer uma base de engenharia e de ciência para a Engenharia de Software. Existem quatro métodos relevantes para condução de estudos empíricos na área de Engenharia de Software: científico, de engenharia, experimental e analítico [9].

O método científico observa o mundo, sugere o modelo ou a teoria de comportamento, mede e analisa, verifica as hipóteses do modelo ou da teoria. Isto é um paradigma indutivo. Esse método pode ser utilizado quando se tenta entender, por exemplo, o processo, o produto de software, ambiente. Ele tenta extrair do mundo algum modelo que possa explicar um fenômeno, e avaliar se o modelo é realmente representativo para o fenômeno que está sob observação. Isto é uma abordagem para construção de modelos e teorias.

O método da engenharia observa as soluções existentes, sugere as soluções mais adequadas, desenvolve, mede e analisa, e repete até que nenhuma melhoria adicional seja possível. Isto é uma abordagem orientada à melhoria evolutiva que assume a existência de algum modelo do processo ou produto de software e modifica este modelo com propósito de melhorar os objetos do estudo.

O método experimental sugere o modelo, desenvolve o método qualitativo e/ou quantitativo, aplica um experimento, mede e analisa, avalia o modelo e repete o processo. Isto é uma abordagem orientada à melhoria revolucionária. O processo se inicia com o levantamento de um modelo novo, não necessariamente baseado em um modelo já existente, e tenta estudar o efeito do processo ou produto sugerido pelo modelo novo.

O método analítico (ou matemático) sugere uma teoria formal, desenvolve a teoria, deriva os resultados e se possível, compara-a com as observações empíricas. Isto é um método dedutivo que não precisa de um projeto experimental no sentido estatístico, mas oferece uma base analítica para o desenvolvimento de modelos.

Supõe-se que a abordagem mais apropriada para a experimentação na área de Engenharia de Software seja o método experimental que considera a proposição e avaliação do modelo com os estudos experimentais, entretanto, existe a possibilidade da utilização de outros métodos.

É importante notar que experimentos não provam nada. É verdade que nenhum experimento oferece prova com certeza absoluta. Os experimentos verificam a previsão teórica de encontro à realidade. A comunidade aceita uma teoria se todos os fatos conhecidos dentro de seu domínio possam ser deduzidos da teoria, possui uma verificação experimental extensa e prediz o novo fenômeno corretamente.

Uma informação mais detalhada a respeito do papel da experimentação na área de Engenharia de software pode ser encontrada em Basili [11].

2.2 Seleção da Estratégia Empírica

Selecionar uma estratégia empírica para a investigação na engenharia de software é uma atividade complexa, principalmente porque os benefícios e desafios para utilização de cada estratégia ainda não estão bem catalogados [12].

Apesar do grande interesse na engenharia de software empírica, há pouca orientação para saber quais estratégias são mais adequadas para um problema de pesquisa, e como escolher entre as estratégias existentes. Muitos pesquisadores selecionam estratégias inadequadas porque eles não conhecem claramente os objetivos do estudo ou possuem pouco conhecimento sobre as estratégias empíricas [12].

A seleção de estratégias empíricas depende de contingências locais, incluindo os recursos disponíveis, o acesso ao tema de pesquisa, o controle das variáveis de interesse, e, naturalmente, as habilidades do investigador.

Durante o restante desta seção serão apresentados passos [12] para a seleção da estratégia empírica que melhor se adéqüe ao estudo empírico.

Para selecionar a estratégia adequada, primeiramente é necessário definir o objeto de estudo, ou seja, o que se quer investigar, bem como qual a teoria envolvida para então seguir os próximos passos.

Depois é apresentada uma análise de questionamentos e quais destes podem ser considerados válidos. Para verificar a validade das questões, serão apresentadas as posições filosóficas que podem apoiar a pesquisa empírica. Além dos questionamentos e da posição filosófica também é necessário observar qual será o objetivo do estudo empírico. Tendo esses

pontos definidos é possível, conforme mostra a Figura 2.1, selecionar a estratégia empírica mais adequada para o estudo dentre as disponíveis como, por exemplo: *surveys*, estudos de caso e experimentos controlados.

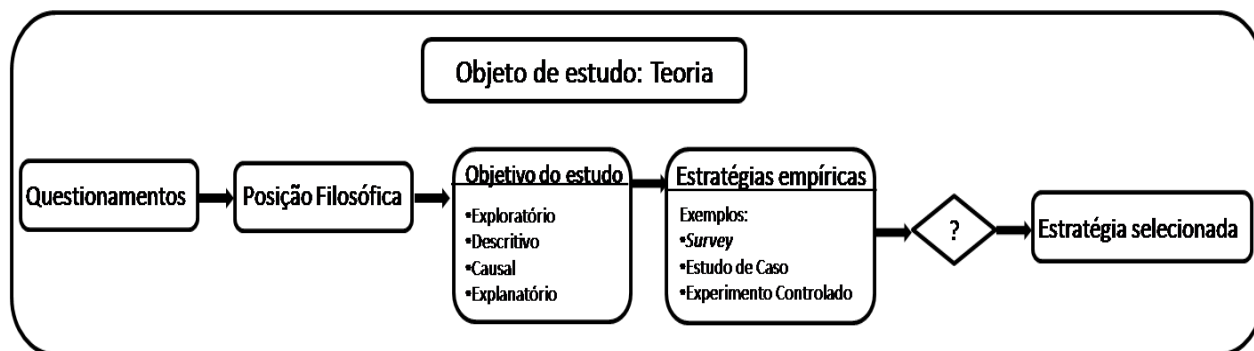


Figura 2.1 Passos para selecionar uma estratégia empírica

2.2.1 Questionamentos

Um dos primeiros passos na escolha de uma estratégia empírica adequada é esclarecer as perguntas da pesquisa. Tendo identificado o problema, que é o objeto de estudo, é importante esclarecer as perguntas que envolvem a decisão e orientação de qual a estratégia mais adequada a ser utilizada. A classificação das questões que serão utilizadas é uma adaptação de Meltzoff [13].

Nos estágios iniciais de uma investigação geralmente é necessário perguntas exploratórias, com a tentativa de compreender o fenômeno e identificar distinções para esclarecer nosso entendimento. A Tabela 2.1 fornece uma lista de perguntas exploratórias que auxiliam nestes estágios iniciais.

Tabela 2.1. Perguntas Exploratórias [12]

Existência	Será que a ferramenta X vai ajudar os desenvolvedores?
Descrição	Como funciona a tecnologia X? Quais suas propriedades? Como pode ser classificado? Como podemos medir? Qual a sua finalidade? Quais seus componentes?
Descritivo-comparativo	Como a ferramenta X difere da ferramenta Y?

As respostas as perguntas da Tabela 2.1 resultam em uma melhor compreensão dos fenômenos, incluindo definições mais precisas dos termos teóricos, provas de que podemos medi-los, e prova de que as medidas são válidas.

Uma vez que temos uma compreensão mais clara do nosso estudo, talvez sejam necessárias algumas questões sobre a ocorrência do fenômeno. Exemplo dessas perguntas são apresentados na Tabela 2.2.

Tabela 2.2. Perguntas de Ocorrências [12]

Frequência de distribuição	Quantas vezes a ferramenta X apresentou problema?
Descrição do processo	Como os desenvolvedores reportam suas atividades sem o uso da ferramenta X? Qual é o processo que a ferramenta X está inserida? Quais são as etapas que a ferramenta X é usada? Como é que a ferramenta X atinge seu propósito?

Outro conjunto de perguntas se aplica se o interesse do estudo estiver associado à relação entre dois fenômenos diferentes, e especificamente se a ocorrência de um está relacionada à ocorrência de outro. Uma vez que se tenha demonstrado que existe uma relação entre dois fenômenos, é natural tentar explicar por que a relação acontece tentando identificar a causa e o efeito. A Tabela 2.3 apresenta os tipos de perguntas para esses casos.

Tabela 2.3. Perguntas Relacionais e Causais [12]

Relacionamento	A ferramenta X e Y estão relacionadas? A ocorrência do uso da ferramenta X se relaciona com a ocorrência do uso da ferramenta Y?
Causalidade	A ferramenta X causa o uso da ferramenta Y? Quais são os fatores que causam o uso da ferramenta Y? Qual efeito na produtividade do desenvolvimento de software quando usa a ferramenta X comparada com o uso da ferramenta Y?
Causalidade comparativa	Será que o uso da ferramenta X causa maior produtividade no desenvolvimento de software do que a ferramenta Y e Z?

2.2.2 Posição Filosófica

Tendo especificado as questões da pesquisa, é preciso considerar o que aceitar como respostas válidas. Diferentes pessoas fazem diferentes suposições sobre a verdade científica, gerando o conflito e resultando em diferentes posturas filosóficas [12].

Para entender as diferentes posturas, é necessário conhecer o que os filósofos fazem para distinguir entre epistemologia (a natureza do conhecimento humano, e como obtê-lo) e ontologia (a natureza do mundo, independentemente das nossas tentativas de entender). Esta separação nos ajuda a discutir o que aceitamos como conhecimento científico [14]

Os pesquisadores podem não serem capazes de convencer outras pessoas sobre suas posições filosóficas, mas serão capazes de argumentar de forma convincente o porquê escolheram as estratégias empíricas. Conforme Creswell [15] quatro posições filosóficas podem ser usadas, são elas: positivista, construtivista, teoria crítica e o pragmatismo.

Para o positivista, a ciência é o processo de verificação de teorias por meio de testes de hipóteses deles derivados. Para os construtivistas, a ciência é o processo de buscar as teorias que emergem e explicar os dados. Para o teórico crítico, as teorias são afirmações de conhecimento a serem criticadas. Para o pragmatista, as teorias são o produto de um consenso entre a comunidade de pesquisadores, para ser julgado por sua utilidade prática.

A escolha de uma posição filosófica é dada a partir do tipo da investigação que os pesquisadores queiram dar ao objeto de estudo, conforme as quatro posições citadas acima, ajudando a justificar as respostas dos questionamentos da seção anterior como válidas e guiando os pesquisadores em mais uma etapa para a escolha da estratégia empírica.

É importante compreender que em qualquer estudo empírico as teorias têm um forte impacto sobre como as coisas são observadas e interpretadas. A teoria torna-se uma "lente" através do qual o mundo é observado.

2.2.3 Objetivo do Estudo

Além dos questionamentos e a posição filosófica, também é necessário observar qual será a classificação do estudo experimental, ou seja [12]:

- Exploratório: usado para construir novas teorias a partir das existentes. Por exemplo: Quais são as experiências dos desenvolvedores que usaram a tecnologia X?
- Descritivo: usado para descrever uma sequência de eventos. Por exemplo: Como o uso da tecnologia X realmente funciona?

- Causal: usado para determinar se há relação causal entre fenômenos. Por exemplo: Uma ferramenta X de rastreamento de requisitos ajuda os programadores a encontrar as informações mais rápidas?
- Explanatório: usado para explicar relações existentes na teoria. Por exemplo: Por que a ferramenta X deve ser considerada a melhor para rastrear requisitos?

2.2.4 Estratégias Empíricas

A estratégia empírica mais apropriada em uma situação concreta vai depender, por exemplo, da teoria apresentada para o objeto de estudo, dos objetivos do estudo, das propriedades do processo de software usado durante o estudo, ou dos resultados finais esperados.

Em síntese, a literatura descreve três principais estratégias empíricas:

- *Survey* (Pesquisa de opinião): Uma pesquisa de opinião é, em geral, uma investigação baseada em retrospecto. Ela pode ser realizada, por exemplo, após a utilização de uma ferramenta ou técnica. Um questionário é construído para obter a informação necessária para a pesquisa, que deve ser realizada através de uma amostra representativa da população a ser estudada. Os questionários são respondidos pela amostra selecionada e os resultados da pesquisa são coletados e analisados para derivar uma conclusão descritiva, explanatória, exploratória ou relacional. O resultado é então generalizado para a população representada. As vantagens dos *surveys* são a facilidade da repetição e baixo custo, principalmente para os *surveys* aplicados na internet, e a desvantagem é o falta de controle na execução quando não é usada uma ferramenta que possa auxiliar na gestão da pesquisa.
- Estudo de caso: Os estudos de caso são utilizados para monitorar projetos, atividade ou tarefas. Os dados são coletados para um propósito específico do estudo. É normalmente dirigido para estudar um atributo específico ou estabelecer relacionamentos entre diferentes atributos. O nível de controle é mais baixo em um estudo de caso do que em um experimento. A vantagem dos estudos de caso é que são fáceis de planejar, e a desvantagem é a dificuldade na generalização dos resultados, isto é, é possível mostrar os efeitos em uma situação específica, mas esses efeitos não podem ser generalizados para uma situação qualquer.
- Experimento Controlado: Um experimento representa uma investigação formal, rigorosa e controlada. Experimentos são normalmente realizados em laboratórios, provêm um nível maior de controle e são indicados quando se quer controlar toda a

situação e manipular comportamentos de forma precisa, direta e sistemática. Além disso, experimentos envolvem mais de um tratamento para comparar os resultados. Os elementos principais do experimento são as variáveis, os objetos, os participantes, o contexto do experimento, hipóteses e o tipo de projeto do experimento. A vantagem dos experimentos controlados está relacionada à sistematização do controle para se obter resultados sem viés e com qualidade, devido ao controle extremo tem suas desvantagens pois a repetição e o custo se tornam altos.

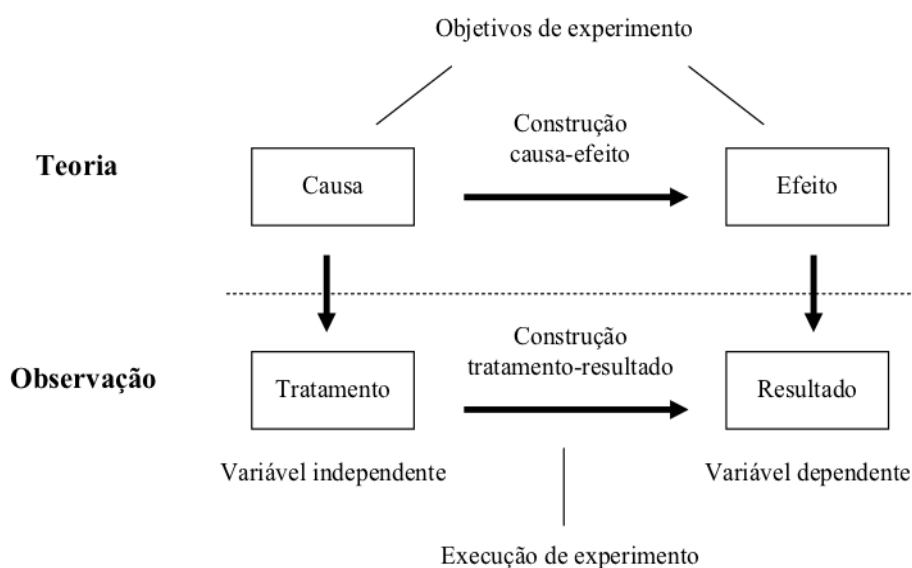


Figura 2.2 Os conceitos de um experimento [9]

Há dois tipos de variáveis do experimento: dependentes e independentes. As variáveis independentes referem-se à entrada do processo de experimentação. Essas variáveis também se chamam "fatores" e apresentam a causa que afeta o resultado do processo de experimentação. O valor de um fator se chama "tratamento". As variáveis dependentes referem-se à saída do processo de experimentação. Essas variáveis apresentam o efeito que é causado pelos fatores do experimento. O valor de uma variável dependente se chama "resultado" [9]. A Figura 2.2 apresenta os relacionamentos entre os conceitos descritos acima.

Tabela 2.4. Comparação das Estratégias Empíricas

Fator	Survey	Estudo de caso	Experimento
O controle da execução	Baixo	Baixo	Alto
O controle de medição	Baixo	Médio	Alto
O controle de investigação	Baixo	Médio	Alto
Facilidade da repetição	Alta	Baixa	Alta
Custo	Baixo	Médio	Alto

Cada uma das três técnicas apresentadas na Tabela 2.4 conduz estudos empíricos — pesquisa de opinião, estudo de caso e experimento controlado — apresenta vantagens e desvantagens distintas. Tais vantagens estão ligadas à qualidade e à utilização dos dados obtidos, a serem consideradas pelo pesquisador quando escolher a mais apropriada para seu objetivo de pesquisa.

Segundo Travassos e outros [16], para um estudo empírico ter uma importância científica ou industrial, seus resultados devem ser obrigatoriamente válidos. Isso significa que os resultados de um estudo devem ser considerados segundo quatro tipos de validade: de conclusão, interna, externa e de construção, que apresentam respectivamente, os aspectos a respeito da análise estatística, o relacionamento tratamento-resultado, a generalização dos resultados a uma população maior, e a relação entre a teoria e a observação [17].

Uma observação importante, é que os diferentes métodos de investigação não são ortogonais entre si. Um estudo empírico completo pode ser realizado considerando todos os métodos descritos. Segundo Wohlin e outros [9], um estudo experimental completo poderia, por exemplo, adquirir a informação inicial utilizando um *survey*, elaborar uma teoria através de um experimento controlado e verificar a teoria proposta em condições reais por meio de um estudo de caso.

Outro tipo de estratégia usada em Engenharia de Software são as revisões sistemáticas, ou seja, quando se deseja integrar os resultados provenientes de diversos estudos experimentais correlatos. Uma revisão sistemática atua como um meio para identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse. Além disso, uma revisão sistemática deve obrigatoriamente conter o protocolo de busca executado de forma a permitir que a revisão seja repetida por outros pesquisadores interessados [18].

2.3 Processo de Experimentação em Engenharia de Software

No que diz respeito à condução de estudos experimentais, Wohlin e outros [9] definiram um processo composto por cinco principais etapas: Definição, Planejamento, Execução, Análise e Interpretação, e Apresentação e Empacotamento. A etapa de Definição é a primeira atividade, onde o estudo experimental é expresso em termos dos problemas e objetivos. Durante a etapa de Planejamento, o projeto do estudo é determinado, a instrumentação a ser utilizada é definida, e os aspectos da validade dos resultados são analisados, resultando na elaboração do Plano de Estudo Experimental.

O Plano de Estudo Experimental desempenha um papel crucial no contexto de um estudo. A elaboração do plano deve contemplar aspectos que venham a minimizar a probabilidade de ocorrência de riscos durante a etapa de Execução. Dependendo da magnitude dos riscos, há uma possibilidade considerável do estudo ser invalidado.

Dessa forma, o processo de experimentação contém um ponto de controle (visto na Figura 2.3) onde o plano do estudo deve ser avaliado e, dependendo da decisão tomada, opta-se pelo replanejamento ou pelo prosseguimento do estudo.

Em alguns casos, o plano do estudo deve ser revisado, de preferência por pesquisadores que não possuam interesse direto nos resultados do estudo, para minimizar a presença de viés, em

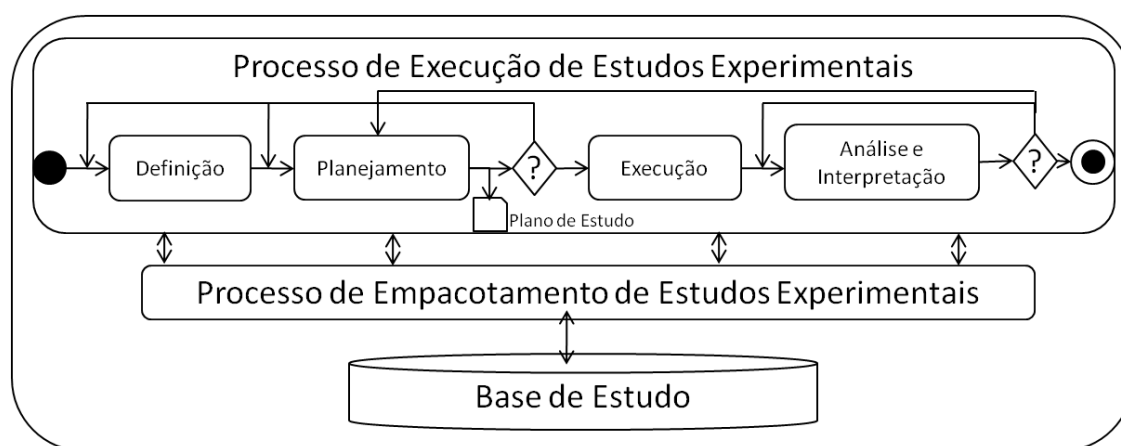


Figura 2.3 Processo de Experimentação Adaptado [9]

outras situações, onde a consequência da ocorrência de riscos é mais significativa, a condução de um estudo piloto é recomendada.

Uma vez aprovado o Plano do Estudo Experimental, e decidida pela condução do estudo, a etapa de Execução segue conforme o planejado. Durante essa etapa, os dados experimentais são coletados, de forma a serem analisados e avaliados durante a etapa de Análise e Interpretação. A etapa de Análise e Interpretação pode ser re-executada caso haja necessidade de se avaliar os dados coletados durante a etapa de Execução de uma forma não prevista inicialmente durante o Planejamento.

2.4 Replicação em Estudos Empíricos

A replicação de estudos empíricos é freqüentemente defendida, mas raramente praticada. Por exemplo, Basili e outros argumentam que replicação sistemática de estudos empíricos é fundamental para a construção do conhecimento [19], enquanto Kitchenham e outros identificaram a falta de incentivo para a realização das replicações, como um dos entraves da Engenharia de Software baseada em evidências [20]. Em uma pesquisa na literatura sobre Engenharia de Software Empírica, Sjberg e outros [21] encontraram apenas vinte casos de replicações publicadas, onde apenas nove foram realizados por outros pesquisadores que não fosse à equipe original. O problema não é exclusivo para Engenharia de Software, replicações para outras áreas também são muito raras.

Muito se tem especulado sobre o motivo das replicações serem raras. Entre as razões está à falta de informações publicadas corretamente e mesmo que os materiais estejam disponíveis, a replicação de estudos empíricos requer um conhecimento tácito que nunca pode ser capturado em relatórios publicados [22]. Além disso, as replicações são vistas como menos interessante do que a novas pesquisas, e há uma percepção da comunidade de Engenharia de Software Experimental que as repetições são difíceis de publicar [20].

As replicações podem ampliar a base de dados e a validade estatística dos resultados. Replicação em diferentes contextos é uma característica importante para qualquer tipo de estudo em Engenharia de Software. A replicação e os estudos complementares são encorajados pela disponibilidade dos pacotes que documentam um experimento corretamente e a experiência do pesquisador. Os pacotes descrevem um experimento em termos específicos permitindo replicação, oportunidades de variação e um contexto para combinação de resultados em diferentes tipos de tratamentos experimentais.

As replicações podem ser internas ou externas. As replicações internas são usadas quando os pesquisadores repetem seus próprios experimentos. Em muitas áreas da ciência, repetições

internas são realizadas como parte de um programa de pesquisa, ou quando se pretende melhorar os resultados já existentes. Este tipo de replicação é bastante comum.

As replicações externas são utilizadas quando os pesquisadores que participam são independentes daqueles que inicialmente realizaram o estudo original. Para esse tipo de replicação há um maior poder de confirmação e raramente as repetições externas serão exatas como as originais, por isso é importante analisar e categorizar as diferenças.

As replicações também podem ser classificadas em literal ou teórica. As replicações literais são destinadas a produzirem os mesmos resultados, ou seja, simular os resultados do estudo original, enquanto que as replicações teóricas são destinadas a produzirem resultados contrastantes por razões previsíveis.

As seguintes seções irão discutir um trabalho realizado por Carver [23] mostrando como as replicações são relatadas na literatura e mostrar a contribuição da proposta definida por Carver [23] com algumas diretrizes para o relato de replicações, respectivamente.

2.4.1 Replicações Publicadas

Para relatar uma replicação é importante ter informações sobre o estudo original, o contexto da replicação, as alterações feitas, e os resultados. Carver [23] realizou uma revisão sistemática com algumas replicações coletadas e em geral as replicações não foram relatadas de forma consistente. Cada autor usou sua própria organização para publicar sua replicação.

As replicações coletadas por Carver são discutidas apenas para ilustrar a incoerência com que replicações estão sendo publicadas e identificar os tipos de informações publicadas por vários autores, com o intuito de obter informações para relatar algumas diretrizes que podem guiar os pesquisadores em suas replicações.

Descrição do estudo original

Para fornecer o contexto para o estudo, a replicação deve discutir o papel do estudo original sobre a qual se baseou a replicação. Na análise feita por Carver foi visto algumas abordagens, primeiro alguns autores descrevem totalmente o estudo original em uma seção a parte, outros só apresentam um breve resumo do estudo original e outros não fornecem nenhuma discussão detalhada sobre o estudo original, ou seja, apenas citam durante a replicação. Outro fato também observado por Carver é que mesmo os autores seguindo a mesma abordagem eles não incluem o mesmo nível de detalhes. Carver notou que a falta de coerência na forma como autores relatam as informações do estudo original leva a necessidade de uma padronização.

Comparação dos resultados da replicação com os resultados do estudo original

Um dos principais benefícios de uma replicação experimental é que ela proporciona aos pesquisadores a capacidade de confirmar, refutar, ou aprofundar as conclusões de um estudo anterior. Para tirar essas conclusões, os resultados da replicação devem ser comparados com os resultados do estudo original.

Nas revisões publicadas, Carver pode notar diversas formas de relatar a comparação dos resultados da replicação com os resultados do estudo original, por exemplo: um autor integrou a comparação dos resultados ao longo da replicação, outro criou uma seção separada para comparação dos resultados e já outro usou tabelas para deixar mais evidente as comparações.

2.4.2 Guia para Relatar Replicações [23]

Com base na análise da replicações publicadas na literatura como descrito na seção anterior Carver propos os seguintes itens para compor um relatório de uma replicação.

Informações sobre o estudo original

Para ajudar o leitor a compreender a replicação, um documento de replicação deve discutir algumas informações sobre o estudo original. Os autores devem fornecer informações suficientes sobre o estudo original para permitir o leitor interpretar corretamente e compreender a reprodução, sem fornecer tantos detalhes [23].

- Questionamentos - uma descrição dos questionamentos que foi a base para o estudo original;
- Participantes - o número de participantes e as características relevantes dos participantes;
- Projeto - uma interface gráfica (ou textual) da descrição do projeto experimental;
- Artefatos - uma descrição e / ou links para os artefatos utilizados;
- Variáveis de contexto - as variáveis de contexto importantes que afetaram o projeto do estudo e interpretação dos resultados;
- Resumo dos resultados - um breve panorama das principais conclusões.

Informações sobre a replicação

Como acontece com qualquer experimento, as informações básicas sobre o estudo deve ser relatado. Esta atividade incide sobre as informações específicas que precisam ser informada sobre

uma replicação. Um relatório de replicação deve conter as seguintes informações (no mínimo)[23]:

- Motivação para a realização da replicação - a descrição do motivo pelo qual a replicação foi realizado (por exemplo, para validar os resultados, para ampliar os resultados, alterando o pool de participante ou artefatos);
- Nível de interação com pesquisadores originais - Se a replicação é externa (isto é, os pesquisadores originais não estão envolvidos), o nível de interação que os replicadores tinha com o pesquisador do estudo original deve ser comunicado. Essa interação pode variar de nenhuma (ou seja, basta ler o estudo original) para ter interação (ou seja, pesquisadores do estudo original atuaram como consultores). Se um pacote é usado, então o seu uso deve ser descrito;
- Mudanças no estudo original - Todas as alterações feitas no projeto, os participantes, artefatos, procedimentos, dados coletados e / ou técnicas de análise deve ser discutido juntamente com a motivação para a mudança.

Comparação dos resultados

Um dos principais valores de uma replicação é a comparação de seus resultados com os resultados do estudo original. Para fazer a comparação é importante ter uma secção especificamente dedicada a comparar os resultados da replicação com os do resultados do estudo original. Esta secção deve destacar as seguintes informações:

- Resultados consistentes - Resultados da replicação que apoiaram os resultados do estudo original;
- As diferenças nos resultados - os resultados da replicação que não coincide com os resultados do estudo original. Os autores devem também discutir como as mudanças feitas no projeto experimental pode ter causado essas diferenças.

Conclusões

Finalmente, reunindo informações sobre o estudo original, as alterações feitas para a replicação e a comparação dos resultados, os autores deverão apresentar uma discussão sobre o estado atual dos conhecimentos. Ao combinar as conclusões do estudo original com as conclusões a partir da replicação, os autores devem ser capazes de fornecer informações que não teria sido evidente a partir de qualquer estudo individual. Nesta secção os autores devem destacar qualquer conclusões do estudo original que foram reforçadas. Esta secção é também o lugar de propor hipóteses sobre

novas variáveis de contexto que podem ter se tornado evidente através da análise de vários estudos.

2.5 Conclusão

Este capítulo discutiu a importância e necessidade da realização de estudos empíricos na área de engenharia de software, assim como, os principais passos para chegar à seleção da estratégia empírica adequada ao estudo, o processo de experimentação na engenharia de software e as replicações de estudos empíricos.

Foram apresentados também os quatro métodos relevantes para condução de experimentos na área de Engenharia de Software: científico, de engenharia, experimental e analítico [9].

Para as estratégias empíricas (*survey*, estudo de caso e experimento) foi descrito com detalhes cada um a fim de mostrar a dependência que pode existir entre eles e assim mostrar que um estudo empírico completo pode, por exemplo, adquirir a informação inicial utilizando um *survey*, elaborar uma teoria através de um experimento controlado e verificar a teoria proposta em condições reais por meio de um estudo de caso.

No que diz respeito à condução de estudos empíricos, foi apresentado, alguns passos para chegar à conclusão de qual estratégia empírica selecionar para um estudo empírico, o processo de experimentação composto por cinco principais etapas: Definição, Planejamento, Execução, Análise e Interpretação, e Apresentação e Empacotamento e por fim a importância das replicações dos estudos empíricos.

Também foi apresentada a importância de replicações de estudos empíricos na Engenharia de Software e algumas classificações que podem ajudar o pesquisador. Foi apresentado com detalhes o estudo e as diretrizes propostas por Carver [23] com o intuito de mostrar a não padronização quando se trata dos relatos e publicações de replicações de estudos empíricos que existem na literatura e também a proposta definida por Carver onde ele mostrar passos importantes para serem seguidos para documentar as replicações.

Capítulo 3

Pesquisa *Survey*

Este capítulo apresenta os passos para seleção da estratégia empírica *Survey* para a avaliação do Benchmarking Framework [1] e uma abordagem teórica para essa estratégia como investigação empírica. Será dada uma ênfase nas definições e em todo o processo para aplicação de um *Survey*, deixando evidente a necessidade de seguir os passos necessários para se obter um estudo experimental com qualidade.

A Seção 3.1 mostra os passos para a seleção da estratégia empírica que melhor se adéqüe ao objeto de estudo em questão, na Seção 3.2 são apresentados os conceitos e classificações principais de um *Survey*. Na Seção 3.3 é dada uma ênfase nos *Surveys* aplicados na Internet, mostrando vantagens e desvantagens. Alguns conceitos necessários no Planejamento de um *Survey* são apresentados na Seção 3.4 e alguns Processos de Aplicação de um *Survey* definidos na literatura [9] [24] na Seção 3.5. A Seção 3.6 apresenta com detalhes o processo apresentado por Mendonça [25], mostrando os procedimentos para executá-lo. A Seção 3.7 apresenta o Grau de Relevância Experimental que o *Survey* possui. Por fim na Seção 3.8 é feita uma conclusão do que foi apresentado no capítulo.

3.1 Seleção da Estratégia Empírica

Serão apresentados os passos utilizados para chegar à conclusão de qual estratégia empírica selecionar para o nosso estudo empírico conforme apresentado na Figura 2.1.

Para ilustrar as etapas envolvidas para decidir qual estratégia(s) selecionar, será apresentado o **objeto de estudo** que utilizamos para investigação neste trabalho.

3.1.1 O Benchmarking Framework

O Benchmarking Framework (BF) [1] visa reduzir as dificuldades e desafios associados à geração de evidências científicas na Engenharia de Software, mais especificamente para o Desenvolvimento de Software Orientado a Aspectos (DSOA). Em especial o mesmo atua na área de manutenção de software, uma das atividades mais custosas do ciclo de vida de um software. Um dos principais motivos para a dificuldade na obtenção de evidências é a ausência de um conjunto de aplicações comuns, exemplos benchmarks (aplicações e cenários suficientemente representativos), que permitam pesquisadores escolherem aplicações mais adequadas ao estudo, bem como, comparar e correlacionar resultados de diversos estudos que utilizam uma mesma aplicação.

Como o foco é estudos empíricos sobre manutenção de software, precisamos, além das aplicações, de cenários de manutenção comuns (mudanças de requisitos, adaptações, correções de erros, melhorias e etc.), permitindo tais comparações, correlações e estudos adicionais e complementares. Da necessidade de organizar tais aplicações e cenários de mudança, o BF propõe um conjunto de atributos para classificar aplicações e cenários de mudança para serem utilizados por estudos empíricos sobre manutenibilidade no DSOA. Essa classificação deve apoiar a seleção de qual aplicação e cenários são mais adequados para um experimento específico, a partir de um conjunto de aplicativos e cenários disponíveis. Além disso, este conjunto de atributos de classificação, executa um outro papel de auxiliar pesquisadores a considerar outras questões em seus estudos empíricos, por expor diversas características, das aplicações e dos cenários de mudança. Através da utilização do BF também se pretende acelerar a transferência e disseminação das tecnologias utilizadas. O BF pode ser definido como [1]:

“Conjuntos de atributos capazes de identificar características específicas de diferentes aplicações (framework), de modo que seja possível avaliá-las e compará-las (benchmarking) sob um determinado contexto (manutenibilidade de software OA).” (MOURA, 2008, p. 30)

Portanto, como indicado na definição, o BF é destinado a auxiliar a avaliação da manutenibilidade de software orientado a aspectos. O framework define uma estrutura, para que aplicações sejam avaliadas e comparadas com relação aos atributos de manutenibilidade.

Dessa forma, o framework é capaz de guiar pesquisadores e profissionais da área durante os processos de seleção, elaboração ou adaptação de aplicações e seus respectivos cenários de manutenção, tornando-os mais adequados para atender os objetivos específicos de determinados experimentos ou estudos de caso sobre manutenibilidade no DSOA. Além disso, por utilizar o

BF, pesquisadores são guiados a considerar novas variáveis na elaboração, replicação e avaliação de seus estudos empíricos.

O BF é composto por três componentes principais: Processo, Produto e Cenários de Manutenção. Cada um deles possui propriedades específicas que têm o objetivo de avaliar pontos relevantes sobre conceitos e técnicas de manutenibilidade de OA.

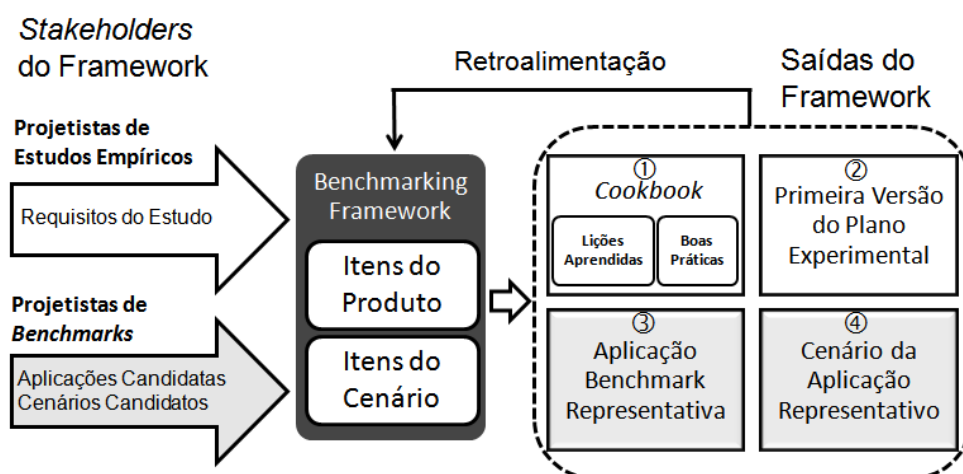


Figura 2.2 Processo definido pelo Benchmarking Framework [1]

A Figura 2.2 ilustra detalhadamente a representação gráfica do processo do *Benchmarking Framework*, contendo seus atores, componentes, entradas e saídas.

O DSOA tem a finalidade de alcançar uma melhor separação de interesses, e com isso aumentar a modularidade dos interesses transversais de um sistema, ou seja, requisitos que apresentam características semelhantes de espalhamento e entrelaçamento por todo o código fonte, característica essa que não pode ser atendida satisfatoriamente apenas com a utilização de orientação a objetos (OO) e padrões de projeto [26]. Com a separação dos interesses transversais, DSOA garante que o código destinado a implementação de cada requisito não irá se misturar e espalhar com o código destinado a outros requisitos do sistema. Os interesses transversais mais comuns em softwares são persistência, distribuição, controle de concorrência, tratamento de exceções e auditoria.

Embora o DSOA vise melhorar a manutenção de software através do encapsulamento de interesses transversais, são raras as evidências empíricas que suportem tal teoria [27].

A realização de estudos empíricos sólidos e controlados na área de engenharia de software no contexto de manutenibilidade no DSOA ainda são bastantes escassos devido à dificuldade e complexidade de padronizar e encapsular todas as informações necessárias aos estudos de forma

que os mesmos se tornem representativos ao que se pretende analisar. Com o intuito de reduzir as dificuldades e desafios associados à realização de tais estudos a contribuição do trabalho de Moura [1] pode ser definida como um conjunto de atributos e cenários de mudanças capazes de identificar características específicas de diferentes aplicações, de modo que seja possível avaliá-las e compará-las sob um determinado contexto (manutenibilidade no DSOA) apoiando a seleção de aplicações adequadas para um estudo específico.

Tendo o BF como objeto de estudo, durante o restante desta seção serão explorados os passos conforme a Figura 2.1 para a seleção da estratégia empírica que melhor se adéque a avaliação do BF.

Questionamentos

Muitas vezes, a pergunta mais óbvia não é a melhor escolha para o ponto de partida. Para avaliar o BF poderíamos ter a seguinte pergunta: A utilização do BF traria vantagens para guiar estudos empíricos sob seu contexto? A mesma é um pouco vaga, pois faz suposição ao fenômeno a ser estudado. Por exemplo, essa questão faz sentido se soubéssemos as pessoas que iriam utilizar o BF, em que circunstâncias, quais seriam as vantagens, como medir, o quanto isso é relevante, qual o contexto entre outras.

Para a avaliação do BF perguntamos sobre sua existência:

- O BF seria útil para guiar os pesquisadores?
- É realmente um problema conduzir estudos sob o contexto de manutenibilidade de softwares orientados a aspectos?

Já com relação a sua descrição e classificação:

- Como podemos medir a importância dos atributos do BF?

Em relação ao descritivo-comparativo que investiga similaridade e diferenças entre dois ou mais fenômenos:

- Como o BF difere dos métodos convencionais para elaborar e executar estudos sobre manutenibilidade de softwares orientados a aspectos?

Ao explorar essas perguntas para o BF [1] podemos refinar as idéias sobre a natureza dos fenômenos em estudo, e possivelmente as respostas já existem na literatura publicada.

Como perguntas de descrição do processo para o BF podemos ter:

- Como é que os pesquisadores elaborarão seus estudos empíricos no contexto de manutenibilidade de softwares orientados a aspectos sem um processo que possa guiá-los?

Podemos perguntar para o BF:

- Será que fazer estudos empíricos sob o contexto de manutenibilidade de softwares orientados a aspectos, está relacionado com a experiência do pesquisador na área?

Posição Filosófica

Positivista, pois queremos mostrar e analisar a concordância dos especialistas em relação aos atributos e cenário de mudança definidos por Moura [1].

Objetivo do Estudo

Explanatório: Por que os atributos e cenários de mudança do BF devem ser considerados como os definidos por Moura?

Estratégia Selecionada

De acordo com o a teoria do BF apresentada por Moura [1] juntamente com os questionamentos apresentados, a aceitação das respostas como verdade de acordo com o posicionamento filosófico construtivista, onde queremos buscar a concordância dos especialistas em relação aos atributos e cenário de mudança definidos por Moura [1], e o objetivo do estudo experimental ser explanatório, é chegado à conclusão da utilização de um *Survey* como estratégia adequada para avaliar os atributos definidos no BF, pois poderemos obter informações de especialistas sob suas opiniões, visto que estudos de casos já foram realizados anteriormente para avaliar o BF com o intuito de avaliar de que forma o mesmo atinge os seus objetivos propostos. Essas avaliações foram executadas de forma a comparar estudos que não faziam o uso do BF com estudos que utilizaram o BF de modo que fosse percebido que algumas decisões sobre os estudos (que não fez o uso do BF) poderiam ter sido consideradas se o BF fosse utilizado.

3.2 *Survey* (Pesquisa de Opinião)

A pesquisa de opinião é provavelmente o método de pesquisa mais popular e utilizado no mundo. Uma pesquisa de opinião está visível não apenas porque encontramos vários exemplos em pesquisas de Engenharia de Software, mas também porque usualmente somos convidados a participar de pesquisas como eleitores, consumidores ou usuário de serviços [24].

Um *survey* não é apenas um instrumento para se obter informação. É uma forma de se coletar informação para descrever, comparar ou explicar conhecimento, atitudes e

comportamentos [24]. Parte da população representativa do estudo é selecionada, e questionários são aplicados para coleta de dados. As pesquisas de opinião podem envolver grandes e pequenas populações (ou universos) através da seleção e estudo de amostras escolhidas na população para se descobrir a incidência relativa.

Com relação ao apoio aos participantes, e dependendo dos objetivos e recursos disponíveis, as pesquisas de opinião podem ser classificadas em 3 tipos [24]:

- Supervisionadas: um pesquisador deve auxiliar o respondente para garantir que todas as questões foram entendidas e respondidas. Entrevistas pelo telefone em geral são deste tipo, onde um entrevistador trabalha em conjunto com o respondente na extração das respostas;
- Semi-supervisionadas: um pesquisador explica os objetivos e o formato da pesquisa de opinião, deixando que o respondente obtenha informação para resposta por conta própria;
- Não supervisionada: não há nenhum auxílio pessoal aos participantes, apenas instruções descritas na pesquisa de opinião. É o tipo mais utilizado, incluem pesquisas por e-mail ou publicadas na Web.

Dependendo do objetivo da pesquisa e da população-alvo (especialmente do seu tamanho e da sua acessibilidade) podemos ter diferentes tipos de amostra. O tamanho da amostra é determinado pelos recursos (tempo, dinheiro e recursos humanos) disponíveis. Esta disponibilidade de recursos influencia o planejamento da administração do instrumento, bem como a codificação das respostas, seu processamento e, eventualmente, as possíveis análises [3]. Com relação à seleção da amostra, as pesquisas de opinião podem ser classificadas em [28]:

- Probabilísticas: as pesquisas são projetadas visando obter uma amostragem representativa e todos os elementos da população têm a mesma chance de serem escolhidos. Isso implica utilizar seleção aleatória dos respondentes, eliminando a subjetividade da amostra. Os métodos baseados em probabilidade incluem usuários pré-recrutados, amostras de toda a população ou amostras baseadas em listas pré definidas. O resultado encontrado para uma amostra pode ser generalizado para a população utilizando-se inferência estatística;
- Não probabilísticas: não é selecionada uma amostra representativa e não produz conclusões que possam ser generalizadas para uma população. A amostra é obtida a partir de algum tipo de critério, e nem todos os elementos da população têm a mesma chance de serem selecionados. Guardando suas limitações, esse tipo de amostra pode

ser conveniente quando os respondentes são pessoas difíceis de identificar, pertencentes a grupos específicos, ou quando existe restrição no orçamento da pesquisa. Os tipos de pesquisa não probabilísticas incluem, por exemplo, votações.

Com relação ao objetivo, uma pesquisa de opinião pode ser classificada como [29]:

- Explanatória: tem como objetivo testar uma teoria e as relações causais, estabelecendo a existência de relações causais, mas também questiona por que a relação existe;
- Exploratória: o objetivo é familiarizar-se com o tópico ou identificar os conceitos iniciais sobre um tópico, dar ênfase na determinação de quais conceitos devem ser medidos e como devem ser medidos, buscar descobrir novas possibilidades e dimensões da população de interesse;
- Descritiva: busca identificar situações, eventos, atitudes ou opiniões de uma população; descreve a distribuição de algum fenômeno na população ou entre os subgrupos da população ou, ainda, faz uma comparação entre essas distribuições. Neste tipo de pesquisa de opinião a hipótese não é causal, mas tem o propósito de verificar se a percepção dos fatos está ou não de acordo com a realidade.

O instrumento utilizado na pesquisa de opinião, o questionário, pode ser administrado em interação pessoal, em forma de entrevista individual ou por telefone, ou pode ser auto-aplicado, através de envio por correio, por e-mail ou publicado na Internet. A forma de distribuição mais adequada vai depender do tipo de pesquisa a ser aplicada. No nosso estudo utilizaremos a publicação na internet.

3.3 *Survey* na Internet

Com a grande evolução tecnológica, a Internet está cada vez mais interativa e as facilidades em se comunicar cada vez mais fácil. O interesse em pesquisas de opinião baseadas na Web não é nenhuma surpresa devido às vantagens sobre os métodos tradicionais de pesquisa, como correio e telefone [30]. Dentre essas vantagens, podemos destacar:

- Redução do tempo;
- Acesso a pessoas nas mais diversas áreas geográficas;
- Menor custo;
- Qualidade dos dados;
- Questionários mais atrativos;
- Possibilidade de amostras maiores;

- Iteratividade.

Entretanto, embora as pesquisas de opinião baseadas na Web pareçam muito atrativas, devem ser usadas com cautela. Ainda existe uma série de restrições sobre a sua aplicação, podemos citar:

- Representatividade;
- Segurança e controle de múltiplas submissões;
- Baixa taxa de resposta;
- Problemas de hardware e software.

Cook e outros [31] conduziram uma meta-análise dos fatores que influenciam a taxa de resposta em pesquisas de opinião na Web. Eles observaram que insistência no contato com participantes que não responderam a pesquisa de opinião, contatos personalizados, e contatos antes da publicação da pesquisa de opinião foram fatores importantes para se obter taxas de resposta mais elevadas. Kittleson [32] em um estudo sobre pesquisas de opinião enviadas por correio, verificou que este tipo de abordagem não funciona, a taxa de resposta não aumentou e houve algumas reclamações de participantes.

O estudo desenvolvido por Burkey e outros [28] identificou os principais problemas na publicação de pesquisas de opinião na Web, as etapas em que estes problemas ocorrem e as possíveis soluções em nível de projeto e implementação. O estudo dá ênfase ao problema da baixa taxa de resposta e a influência do projeto nas respostas. Segundo os pesquisadores a não resposta de uma pergunta pode revelar problemas relacionados ao formato, conteúdo ou tamanho da pergunta, enquanto que a não resposta de uma pesquisa pode estar relacionada a problemas com anonimato, navegador ou servidor. Logo, o projeto do questionário deve ser feito com atenção, visando minimizar qualquer influência nos resultados. Diferentes interfaces podem produzir diferentes efeitos e resultados nas pesquisas de opinião.

3.4 Planejamento de um *Survey*

O planejamento começa revendo os objetivos do estudo, examinando a população alvo identificadas pelos objetivos e decidindo a melhor forma de obter as informações necessárias para chegar naquele objetivo, no entanto temos também que considerar fatores, tais como o tamanho da amostra adequada para assegurar a maior taxa de respostas [33].

3.4.1 Projeto do *survey*

Projetar um *survey* é muito semelhante a elaborar um experimento em que o projeto tem que corresponder aos objetivos, para o que o levantamento dos dados e análise respondam a finalidade do estudo. Normalmente, um *survey* tem um dos dois objetivos fundamentais. São eles: tornar mais concreto um fato nebuloso ou tentar descrever dentre um conjunto de fatores os quais são mais populosos [33]. Como acontece nos demais métodos experimentais, um *survey* pode variar do mais simples ao mais complexo, a decisão será feita dependendo da situação em questão, ou seja, queremos selecionar um projeto que forneça meios eficazes de obter as informações necessárias para chegar ao objetivo. Abaixo está descrito o que seria eficaz para o projeto de um *survey* [33]:

1. Evitar qualquer tipo de viés: Um projeto eficaz não é susceptível de ser indevidamente influenciado por determinadas ações, ou seja, queremos que os resultados do *survey* reflitam representativamente a realidade da situação;
2. Estar adequado ao problema: Um modelo que faça sentido no contexto da população. Deve ser complexo o suficiente para abordar as questões levantadas pelo objetivo do estudo;
3. Ter um custo compatível com sua eficácia: Um projeto cuja administração e análise estejam no meio dos recursos atribuídos a pesquisa. Isto também se aplica a relação custo-benefício dos participantes de um *survey* onde os resultados devem ser úteis para os mesmos, assim motivando os entrevistados a concluir o *survey*.

3.4.2 Descrição do projeto

Tendo os objetivos e recursos para um *survey*, existem três tipos de concepção [33]:

1. Transversal (*Cross sectional*): os participantes são convidados a dar as informações em um determinado ponto fixo no tempo.
2. Corte (*Cohort*): este tipo de estudo é voltado ao futuro, oferecendo informações sobre mudanças de uma população específica.
3. Controle de caso (*Case control*): este estudo é retrospectivo, são perguntas sobre suas circunstâncias anteriores para ajudar a explicar um fenômeno corrente.

3.4.3 Projeto Experimental

Abaixo serão descritas cinco principais opções de projeto do qual pode haver escolha, cada modelo permite testar uma hipótese formal [33]:

1. Estudo de controle concorrente nos quais os participantes são divididos aleatoriamente em grupos: nesse estudo os participantes são submetidos a grupos experimentais aleatoriamente, garantindo assim uma diversidade representativa dos participantes dependendo do contexto da pesquisa;
2. Estudo de controle concorrente nos quais os participantes não são divididos aleatoriamente em grupos: Há ocasiões em que os grupos experimentais existem naturalmente, não permitindo os participantes serem atribuídos ao acaso;
3. Estudos auto-controlados: Esses estudos são baseados em pré e pós medidas, ou seja, quando é necessária uma avaliação antes de um determinado fato e após o acontecimento do mesmo;
4. Controle histórico do estudo: esse tipo de projeto pode ser utilizado quando se pretendem fazer comparações entre *surveys* anteriores;
5. Estudo utilizando combinações de técnicas: o projeto pode necessitar da combinação de diferentes tipos de técnicas.

3.4.4 Taxas de respostas

Qualquer estudo confiável deve medir a taxa de respostas, ou seja, a proporção de participantes que responderam em comparação ao número que foram contatados. A validade dos resultados do *survey* é gravemente comprometida se houver um nível significativo de não respostas. Se for obtido um grande nível de não respostas, mas for compreendida a razão e ainda o grupo de entrevistados que responderam for representativo, podemos prosseguir com a análise. Existem maneiras de planejar o estudo para aumentar as taxas de respostas, por exemplo, mostrando benefícios para os entrevistados para assim valer a pena o tempo para responder. Outra forma também é envio de lembretes aos participantes ao aproxima-se dos indivíduos pessoalmente, quando possível. A seguir, são apresentadas medidas que podem melhorar as taxa de respostas, buscando assegurar que as pessoas envolvidas são [33]:

1. Capazes de responder as perguntas: validar se o questionário está simples e de fácil entendimento;
2. Estão dispostos a responder as perguntas;
3. Estão motivados a responder as perguntas: benefícios aos entrevistados.

3.4.5 Seleção das questões

Um questionário solicita aos entrevistados a responder perguntas de forma que nosso ponto de partida para construir o questionário deve sempre ser os propósitos e objetivos do estudo. Assim definimos o contexto da pesquisa e das questões para cobrir as áreas de interesse. O objetivo diz o quanto é bem vindo os resultados e, portanto, restringe o contexto das questões [35].

Compreender os entrevistados: As perguntas devem ser formuladas de forma que os entrevistados possam respondê-las com facilidade e precisão. É importante certificar se os entrevistados têm suficientemente conhecimentos para responder as perguntas. É essencial que especifique a partir de que período no tempo se pretende obter as respostas extraídas no questionário, pois respostas imprecisas podem ser obtidas quando isso não é levado em consideração [35].

Perguntar um número apropriado de questões: Receber um questionário com muitas questões pode ser assustador a primeira vista. Assim é importante manter em mente que um número suficiente deve ser considerado. Em um questionário via web é preciso ter a certeza que o mesmo não irá sobrecarregar os entrevistados, mantendo as perguntas mais simples de forma que o entrevistado não precise passar muito tempo pensando sobre a questão, a não ser que o tema da questão seja de importância primordial para os entrevistados. Perguntas podem ser agrupadas por temas e assim manter um equilíbrio entre o que se pretende realizar e o que os entrevistados vão responder [35]. É importante deixar claro que a duração do tempo de respostas não está sempre relacionada com o número absoluto de perguntas.

Padronizar os formatos das respostas: Uma forma de reduzir o tempo necessário para os entrevistados responderem o questionário é ter um formato padronizado para as respostas. Por exemplo, para atitudes as respostas geralmente são padronizadas em uma escala:

Concordo plenamente, Concordo, Discordo, Discordo plenamente

É importante deixar claro o que significa cada escala, explicando a diferença entre cada uma dessas respostas de forma a não confundir o entendimento do entrevistado.

Se todas as respostas são padronizadas, os entrevistados conhecem as opções para cada tipo de questão e não têm que “perder” tempo para ler as respostas cuidadosamente. Assim, os entrevistados podem responder mais rápido quando existe um formato padrão.

3.4.6 Formato do questionário

Para questionários via web é importante considerar as instruções do questionário, ou seja, reservar uma seção no início para informar os procedimentos necessários para responder o

questionário, tentando assim eliminar as futuras dúvidas que o entrevistado possa ter durante a leitura do questionário.

Para as informações e instruções fazer a seguinte lista [35]:

1. Explicar a finalidade do estudo;
2. Descrever quem está participando do estudo;
3. Incluir data, fornecer nome de contato e número de telefone;
4. Personalizar a saudação;
5. Explicar porque o entrevistado foi escolhido;
6. Explicar o modo de devolução do questionário;
7. Fornecer uma estimativa realista do tempo necessário para completar o questionário.

Tais passos são essenciais para evitar qualquer dúvida do entrevistado durante suas respostas, evitando respostas indesejadas e questionários incompletos.

3.5 Processos de aplicação de *Survey*

A grande popularidade das pesquisas de opinião ocasiona problemas inevitáveis, resultando em um número elevado de pesquisas ruins. As causas de fracasso na aplicação de pesquisas de opinião são as mais diversas, como: a não utilização de metodologia ou a aplicação de pesquisas de opinião quando outro método de pesquisa seria mais apropriado. A difusão do uso das pesquisas de opinião pode passar a impressão de que é um tipo de pesquisa simples, uma opção fácil para os pesquisadores obterem informações sobre produtos, contextos ou processos. Entretanto, como qualquer outra estratégia de investigação, as pesquisas de opinião requerem a utilização de um processo em sua aplicação.

O objetivo de utilizar um processo para aplicação de pesquisas de opinião é prover um mecanismo de apoio para o acompanhamento e a condução da pesquisa. Um processo é composto por um conjunto de etapas ordenadas que envolvam uma série de atividades, restrições e recursos com o intuito de produzir uma saída.

Os processos são importantes, pois impõem consistência e estrutura a um conjunto de atividades. A estrutura do processo guia as ações permitindo inspecionar, entender, controlar e melhorar as atividades que compõem o processo [36].

Serão apresentadas três abordagens de processo de aplicação de *survey* encontrada na literatura, um proposto por Wohlin e outros [9] que engloba as estratégias de investigação aplicáveis em Engenharia de Software, Pfleeger e outros [24] que descreve as etapas de um

processo de *survey* genérico, e suas atividades mais críticas e por fim Mendonça [25] define um processo baseado nos dois anteriores para aplicação de *survey*.

3.5.1 Processo proposto por WOHLIN [9]

O processo (Figura 3.2) proposto por Wohlin e outros [9] é válido para os três tipos de estratégias de investigação: pesquisas de opinião, experimentos e estudos de caso. Entretanto, sua descrição é focada no experimento, apesar de todas as atividades do processo ser comuns a qualquer estudo experimental. A primeira etapa do processo é a definição do experimento, onde define-se o problema a ser estudado, os objetivos e as metas. O planejamento vem após, onde o projeto do experimento é determinado, a instrumentação é considerada e os tratamentos do experimento são avaliados. A operação do experimento é realizada após o projeto. Na etapa de operação os dados são coletados, para após serem analisados e avaliados na etapa de análise e interpretação. Por fim, os resultados são empacotados e apresentados na etapa de apresentação e empacotamento.

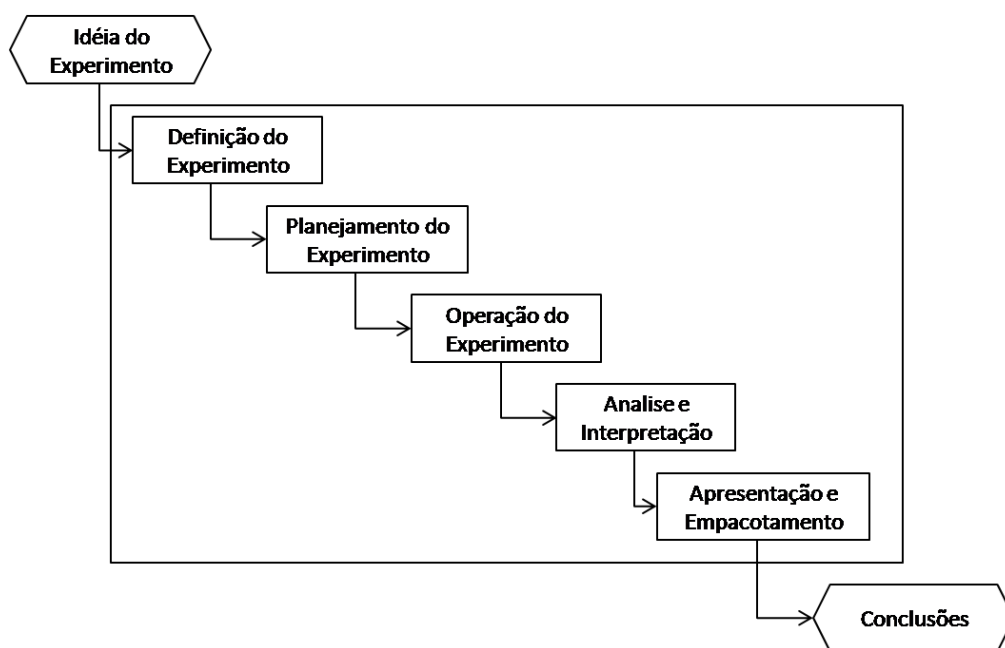


Figura 3.2 Processo de Execução de Experimentos [9]

3.5.2 Processo proposto por PFLEGEER e KITCHENHAM [24]

Ao verificar que os resultados obtidos pela comunidade científica para apoiar a transferência de tecnologia não se enquadravam ao tipo de resultado procurado pelos engenheiros de software, Pfleeger e outros [24] decidiram aplicar uma pesquisa de opinião para definir os critérios de avaliação e o tipo de evidência que os engenheiros de software utilizam para tomada de decisão

sobre tecnologia. Entretanto, ao analisar as pesquisas de opinião já aplicadas foram detectados erros no projeto, na construção, na administração e na análise das pesquisas de opinião.

Com o objetivo de obter resultados mais significativos na aplicação de pesquisas de opinião em Engenharia de Software, as pesquisadoras realizaram um trabalho investigativo visando definir diretrizes de organização, administração e análise de pesquisas de opinião. Neste trabalho foram utilizadas pesquisas de opinião já publicadas para ilustrar procedimentos incorretos e erros mais comuns. Segundo [24], o instrumento da pesquisa de opinião é apenas parte de um longo processo, que define as seguintes etapas:

- I. Definição de objetivos, o que deve ser medido;
- II. Planejamento e agendamento da pesquisa de opinião;
- III. Garantir que recursos necessários estarão disponíveis;
- IV. Projeto da pesquisa de opinião;
- V. Preparação do instrumento de coleta de dados;
- VI. Validação da instrumentação;
- VII. Seleção dos participantes;
- VIII. Execução da pesquisa de opinião;
- IX. Análise dos dados;
- X. Relato dos resultados.

O trabalho apresentado por Pfleeger e outros [24] não foca no processo em si, e sim nas atividades mais relevantes como: definição do tipo de projeto da pesquisa de opinião, definição do tamanho da amostra, métodos de amostragem, procedimentos para obter uma taxa de resposta mais alta, construção das questões, formatação das questões, avaliação da instrumentação e análise dos dados. Algumas destas atividades foram utilizadas como base para o *survey* que será proposto nesse capítulo.

Os processos anteriores possuem muitas atividades em comum, onde parte das diferenças está no nível de detalhamento das atividades e no foco do processo.

3.5.3 Detalhamento do processo proposto por MENDONÇA [25]

Com o propósito de definir um processo para aplicação de pesquisas de opinião as abordagens apresentadas anterior foram analisadas e Mendonça [25] propôs um processo composto pelas seguintes etapas:

- I. Definição de objetivos;
- II. Planejamento da pesquisa de opinião;
- III. Projeto dos instrumentos da pesquisa de opinião;
- IV. Validação da instrumentação;
- V. Execução da pesquisa de opinião;
- VI. Análise dos dados;
- VII. Empacotamento da pesquisa de opinião.

A seguir, as etapas do processo de aplicação de pesquisas de opinião serão descritas com mais detalhes.

1. Definição dos objetivos

Um fator importante é descrever os objetivos, deixando claro o objeto de estudo, o foco da qualidade, o ponto de vista e o contexto. Como resultado, a fase de definição fornece à direção geral do experimento, o seu escopo, a base para formulação das hipóteses e as notações preliminares para avaliação da validade [9]. A fim de capturar estas informações pode-se utilizar a abordagem GQM *Goal/Question/Metric* definida por Basili [34], cujo template básico será descrito a seguir:

- Analisar** <objeto(s) do estudo>
- Com o propósito de** <objetivo>
- Com respeito a** <foco de qualidade>
- Do ponto de vista** <perspectiva>
- No contexto de** <contexto>.

Nessa etapa deve-se verificar se o *survey* é o método de pesquisa mais apropriado para atingir aos objetivos. Também é recomendável fazer uma pesquisa na literatura, visando identificar estudos anteriores.

2. Planejamento da pesquisa de opinião

E preciso definir como e quando a pesquisa será aplicada. A etapa de planejamento envolve a seleção do contexto, a formulação das hipóteses, a seleção das variáveis, a seleção dos participantes, o projeto da pesquisa e a definição da infra-estrutura necessária para execução da pesquisa.

A **seleção do contexto** envolve a definição das condições em que a pesquisa será realizada, e será determinada a partir dos recursos disponíveis e da facilidade de alocação destes recursos.

A **formulação das hipóteses** é a base para a análise estatística de um experimento. A definição da pesquisa é, então, formalizada através da formulação das hipóteses. Basicamente dois tipos de hipóteses devem ser formulados: a hipótese nula e a(s) hipótese(s) alternativa(s). A hipótese nula (H_0) assume que não há nenhum relacionamento estatisticamente significativo entre a causa e o efeito, e de forma geral indica uma igualdade a ser contestada. A(s) hipótese(s) alternativa(s) (H_1, H_2, \dots) sugere(m) que existe um relacionamento entre a causa e efeito. O objetivo principal do experimento é, então, rejeitar a hipótese nula em favor de uma ou algumas hipóteses alternativas. A probabilidade de ocorrer um erro durante a verificação das hipóteses depende da potência do teste estatístico. Mais detalhes sobre testes estatísticos podem ser encontrados em Wohlin e outros [9].

A **seleção das variáveis** consiste na seleção das variáveis independentes e dependentes. As variáveis independentes são as variáveis que serão controladas ou alteradas no estudo experimental. As variáveis independentes devem causar algum efeito nas variáveis dependentes e devem ser controláveis [9].

É importante ressaltar que a formulação de hipóteses e a seleção de variáveis dependem da natureza do *survey* elaborado, ou seja, estudos que são causais e que possuem um tratamento são aptos a possuírem as suas hipóteses e suas variáveis.

A **seleção dos participantes** define a população alvo, a seleção da amostra e a definição de procedimentos para obter uma alta taxa de respostas (envio de lembretes por e-mail, abordagem pessoal, etc). Deve-se conhecer bem o objeto de estudo para determinação do tamanho ideal da amostra para a mesma ser representativa. Para selecionar uma amostra pode se utilizar métodos probabilísticos ou não probabilísticos [24].

O **projeto da pesquisa** determina a forma como a pesquisa será conduzida, como os testes serão organizados e executados. A escolha do projeto de uma pesquisa está diretamente relacionada à análise dos dados. Para projetar uma pesquisa é necessário considerar as hipóteses para verificar que tipo de análise estatística deve ser executado para rejeitar a hipótese nula [9].

A **definição da infra-estrutura** depende principalmente da complexidade da pesquisa, do número de participantes e dos recursos disponíveis para execução da pesquisa. Os recursos incluem desde laboratórios a softwares e recursos financeiros.

3. Projeto dos instrumentos da pesquisa de opinião

Um instrumento de um *survey* (questionário) deve ser projetado cuidadosamente para evitar informações e perguntas mal formuladas deixando o entrevistado confuso. O instrumento será composto por um conjunto de questões, que podem ser expressas de duas formas: abertas ou fechadas. Quando as questões são abertas os entrevistados são questionados para escrever sua própria resposta. Inversamente as questões fechadas os entrevistados são questionados a selecionar uma resposta de uma lista pré-definida. Existem vantagens e desvantagens em cada tipo de questão. Questões abertas evitam restrições aos respondentes, porém podem dificultar a codificação e análise da resposta. As questões fechadas são mais fáceis de analisar, pois há restrições para as respostas.

Como em qualquer método experimental, um projeto de um instrumento de uma pesquisa de opinião pode estender-se de simples a complexo, sendo que o objetivo principal é ter um projeto que possibilite o meio mais efetivo para se obter a informação necessária. Esta efetividade pode ser obtida a partir de três fatores [33]: evitar qualquer tipo de viés, estar adequado ao problema, e ter um custo compatível com sua eficácia. Uma vez que se tenha a idéia do que irá ser perguntado, é importante seguir algumas diretrizes que auxiliam a formulação das perguntas, tais como [35] [37]:

- Deixar claro o relacionamento entre a intenção da questão e os objetivos do estudo;
- Formular questões claras e sem ambigüidade;
- Utilizar linguagem adequada;
- Disponibilizar número apropriado de questões;
- Formular questões de acordo com o conhecimento dos participantes.

Algumas dicas de organização das informações do questionário por Kitchenham e outros [38]:

- Deixar espaço para o participante fazer comentários sobre o questionário;
- Manter espaço em branco entre as questões;
- Usar o formato de folheto;
- Usar tamanho de fonte entre 10 e 12 pt.;
- Usar formato de fonte legível;
- Evitar formatação em itálico;
- Usar formatação em negrito, sublinhado ou letras maiúsculas de forma sensata para dar ênfase ou para representar instruções;
- Não separar entre as páginas instruções, questões e respostas associadas.

4. Validação da instrumentação

A validação é um procedimento fundamental para ter a certeza que o *survey* vai medir o que se pretende medir. A validação da instrumentação busca [37]: validar que as questões são de fácil compreensão, avaliar os procedimentos para se obter uma taxa de resposta alta e efetividade do processo de distribuição, avaliar a validade e confiabilidade da instrumentação, e garantir que as técnicas para análise dos dados estão compatíveis com as respostas esperadas. Uma importante consideração na construção de questionários é o impacto da influência do próprio pesquisador na pesquisa de opinião.

De forma geral, quando o questionário é construído se tem a idéia do que está sendo procurado, e a forma como o instrumento da pesquisa de opinião é construído pode inadvertidamente revelar essa influência. De uma forma geral as pessoas estão mais dispostas a respostas mais completas e precisas quando têm a percepção de que os resultados do estudo serão proveitosos. Por esta razão, deve-se ter certeza que o instrumento da pesquisa de opinião contém todas as informações necessárias aos participantes, tais como: a finalidade do estudo, a relevância do estudo, a importância da participação de cada indivíduo, como e quais critérios foram utilizados na seleção dos participantes e como a confidencialidade será preservada [37]. A forma mais comum de avaliação é a utilização de grupos de discussão ou estudo piloto.

Os grupos de discussão devem ser formados por um grupo de pessoas que represente as pessoas que utilizarão os resultados da pesquisa de opinião e as pessoas que irão responder a pesquisa. Deverão ser discutidos os problemas relacionados a questões e instruções ambíguas, desnecessárias ou ausentes. O estudo piloto da pesquisa de opinião deve ser aplicado a uma pequena amostra, e visa identificar os problemas relacionados a taxa de resposta, distribuição e confiabilidade do processo.

5. Execução da pesquisa de opinião

A etapa de execução da pesquisa de opinião é iniciada com a atividade de convite aos participantes. Esta atividade não é obrigatória, e irá variar de acordo com o planejamento da pesquisa. É importante que tenham sido definidos procedimentos para motivar os respondentes a participarem da pesquisa. O instrumento da pesquisa de opinião deverá então ser distribuído e respondido pelos participantes. Caso a taxa de resposta da pesquisa de opinião esteja baixa e a extensão do prazo não vá influenciar significativamente os resultados, o período de tempo em que a pesquisa de opinião estará disponibilizada pode ser estendido.

Dependendo do tipo de pesquisa também podem ser utilizados procedimentos para monitoramento do progresso da pesquisa, podendo ser feitos ajustes para melhoria do processo, tendo cuidado para não introduzir nenhum tipo de influência. O aspecto mais importante da fase de execução é que a parte humana entra em jogo neste momento. Os participantes devem estar preparados para a experimentação do ponto de vista moral e metodológica para evitar resultados errôneos devido a mal-entendido ou falta de interesse [16].

6. Análise dos dados

A etapa de análise dos dados é um ponto fundamental, pois é nela que será apresentado o resultado de toda a pesquisa. A análise é iniciada com a validação das respostas com relação à consistência e completude. Se a maioria dos questionários for respondida por completo, os questionários incompletos podem ser inutilizados. Entretanto, é importante investigar as causas da não completude, assim como as não respostas, para garantir que não foi introduzido nenhum viés. A seguir, deve ser feita a codificação dos dados.

Geralmente os dados devem ser convertidos para escalas numéricas, pois muitos pacotes estatísticos não tratam categorias representadas por caracteres. A codificação se torna mais difícil para questões abertas. Neste caso, as categorias de resposta têm que ser definidas após o retorno do questionário, e requer a avaliação das respostas por um especialista, havendo a possibilidade de introdução de algum viés [39].

A análise dos dados vai depender do projeto de pesquisa definido anteriormente e do tipo de escala a ser utilizado (nominal, ordinal, razão, intervalar). Mais detalhes sobre como analisar cada tipo de escala podem ser vistos em Pfleeger e outros [39]. De forma geral são utilizados pacotes estatísticos para a análise dos dados. Após a análise dos dados deve ser feito um documento com relato e interpretação dos resultados encontrados. É importante que neste documento seja relatada a taxa de resposta da pesquisa de opinião, isto é, o percentual de pessoas que responderam a pesquisa comparada com o total de pessoas abordadas [24].

7. Empacotamento da pesquisa de opinião

O armazenamento dos artefatos é importante para a repetição da pesquisa no futuro. Uma das características mais importantes de um experimento é a necessidade de repetição. Com a repetição os pesquisadores adquirem conhecimento adicional a respeito dos conceitos estudados, e recebem os resultados que podem ser iguais ou diferentes dos resultados do experimento original. O empacotamento vai descrever a ordem na qual as atividades ocorreram, as pessoas

envolvidas no estudo, o relacionamento entre essas pessoas e as atividades e os documentos utilizados e produzidos durante o estudo [16].

Foram apresentadas algumas abordagens de processo para pesquisas de opinião cujo objetivo foi descrever as etapas fundamentais para aplicação de uma pesquisa de opinião e detalhar as atividades mais complexas envolvidas neste contexto. A fim de que o processo ofereça resultados válidos, ele deve ser apropriadamente organizado e acompanhado. Os processos são importantes à medida que podem ser utilizados como *checklists* e diretrizes do que se deseja fazer e como fazer.

3.6 Grau de relevância experimental

Survey é um dos estudos experimentais mais utilizados, pois possui grande abrangência de utilização, porém recebe pouca atenção da comunidade científica. Atualmente, poucas pesquisas possuem a documentação completa para que o estudo seja repetido, falta principalmente à definição de métodos de amostragem - deficiência grave em diversas áreas, especialmente em engenharia de software.

Geralmente é considerado um estudo fraco sob os aspectos de validade e confiabilidade, especialmente em engenharia de software [38], pois não permite controle sobre as variáveis, ou seja, é difícil identificar relações de causa e efeito tendo assim uma validade interna baixa, isso pode ser visto através da comparação feita na Tabela 2.4. Assegura-se melhor representatividade, permitindo a generalização.

3.7 Conclusão

Este capítulo apresentou os passos para a seleção do *survey* como estratégia empírica e algumas classificações e abordagens para o processo de preparação e execução de um *survey*, deixando evidente a necessidade de seguir os passos necessários para se obter um estudo experimental com qualidade. A fim de que o processo ofereça resultados válidos, ele deve ser propriamente organizado e acompanhado. Os processos são importantes à medida que podem ser utilizados como *checklists* e diretrizes do que se deseja fazer e como fazer.

Foram apresentados três processos que podem ser usados para a condução de *Survey*. O processo proposto por Wohlin [9] é um processo genérico, ou seja, ele pode ser usado tanto para um *Survey* quanto para um Experimento Controlado, por exemplo. Esse tipo de processo quando

seguido pode trazer alguns problemas por ser tão genérico e por isso deve ser usado como base para instâncias que deixe mais evidente passos específicos para cada estratégia empírica. Com isso Pflieger e Kitchenham [24] definiram um processo com atividades bem específicas para condução de *Surveys*, podendo assim ser obtido resultados mais significativos na aplicação dos *Surveys*. É importante destacar que Pflieger e Kitchenham focaram nas atividades mais relevantes para os *Surveys* ficando assim um processo específico.

Por último foi apresentado com detalhes o processo proposto por Mendonça [25], um trabalho mais recente que os outros processos apresentados, para este foi feita uma análise por Mendonça de processos já publicados na literatura para então ser feita uma análise de passos que podiam ser seguidos e que não tinham sido definidos em processos anteriores. É importante destacar que o processo proposto por Mendonça é uma revisão de passos já encontrados na literatura com mais alguns específicos para *Survey*.

Também foram apresentados os procedimentos de um planejamento de um *survey*, mostrando os fatores como: projeto experimental, taxas de respostas, seleção das questões e formato do questionário a fim de fundamentar a teoria para a execução de um *survey* no próximo capítulo.

Considerando a necessidade de reduzir as dificuldades e desafios associados à geração de evidências científicas na Engenharia de Software, mais especificamente para o Desenvolvimento de Software Orientado a Aspectos (DSOA), o próximo capítulo apresenta um estudo experimental (*survey*) cujo objetivo é uma avaliação empírica de uma lista de atributos que classificam as aplicações utilizadas em estudos empíricos sobre manutenibilidade de software orientado a aspectos [1].

Capítulo 4

Uma Avaliação Empírica do *Benchmarking Framework*

O intuito deste capítulo é apresentar o planejamento e avaliação do *survey* desenvolvido como parte deste trabalho de mestrado. Será apresentado o estudo experimental (*survey*) utilizado para avaliar a definição inicial dos atributos de classificação e cenários de mudança conforme a literatura [1], a partir da coleta de informações da realidade da experiência de especialistas da área de orientação a aspectos.

A Seção 4.1 descreve a definição do estudo experimental executado e a Seção 4.2 descreve o planejamento do estudo, mostrando como ele foi projetado. A Seção 4.3 discute as Ameaças à Validade, ou seja, quão válidos são os resultados e se estes podem ser generalizados. A Seção 4.4 fornece a Operação do Estudo Experimental, fazendo uma análise dos dados coletados. Por fim na Seção 4.5 é feita uma conclusão do que foi apresentado no capítulo. Os passos seguidos para descrever o estudo empírico (*Survey*) foram baseados no processo definido por Mendonça [25] conforme seção 3.5.3 com algumas adaptações necessárias para atender ao estudo feito.

4.1 Definição do Estudo Experimental

As seguintes seções descrevem o objetivo de estudo da pesquisa de opinião (*Survey*) executada.

4.1.1 Objetivo

A fim de capturar as informações do objetivo do estudo, foi utilizado o template da abordagem GQM Goal/Question/Metric definida por [34].

Analisar atributos de classificação e cenários de mudança de aplicações utilizadas em estudos empíricos sobre manutenibilidade de software Orientado a Aspectos

com o propósito de caracterizar uma avaliação dos atributos e cenários de mudança que devem apoiar pesquisadores na seleção de qual aplicação é mais adequada para realizar um experimento específico a partir de um conjunto de aplicativos e cenários disponíveis

com respeito a sua importância

do ponto de vista de especialistas do domínio de desenvolvimento de software orientado a aspectos

no contexto de definição de um modelo que possa auxiliar as decisões que precisam ser tomadas durante a seleção de aplicações na elaboração de estudos empíricos sobre manutenibilidade de software orientado a aspectos.

4.2 Planejamento

Esta seção descreve o planejamento de estudo mostrando como ele foi projetado. Isto permite que outros estudos sejam executados usando o mesmo plano o qual pode confirmar alguns dos nossos resultados e derivar novos que nós não pudemos analisar. Futuros estudos podem vir a trocar algumas variáveis a fim de medir seus impactos.

4.2.1 Seleção do contexto

O estudo será conduzido de forma *off-line*, ou seja, o questionário será entregue aos participantes e não será acompanhado. Cada especialista terá o seu tempo e ambiente para responder o questionário, colaborando com o estudo. Os participantes serão pesquisadores com conhecimento em estudos empíricos sobre manutenibilidade de software orientado a aspectos. O contexto possui caráter específico, uma vez que é realizada para uma área específica e endereça um problema real, que é a definição dos atributos necessários para apoiar a classificação de aplicações em estudos empíricos sob o contexto de manutenibilidade de software orientado a aspectos.

4.2.2 Seleção dos Participantes

Os indivíduos serão selecionados baseados em professores e pesquisadores (não estudantes) que fazem parte de comitês de programas de conferências relacionadas com DSOA. As conferências selecionadas foram:

- *International Conference on Aspect-Oriented Software Development* (AOSD) – anos: 2002, 2003, 2004, 2005, 2006, 2007, 2008 e 2009. Site: <http://www.aosd.net/>;
- *Latin American Workshop on Aspect-Oriented Software Development* (LA-WASP) – ano: 2009. Site: <http://www.cin.ufpe.br/~lawasp09/>;
- *International Conference on Object Oriented Programming, Systems, Languages and Applications* (OOPSLA) – anos: 2005, 2006, 2007, 2008 e 2009. Site: <http://splashcon.org/>;
- *European Conference on Object-Oriented Programming* (ECOOP) – anos: 2006, 2007, 2008 e 2009. Site: <http://www.ecoop.org/>;
- *International Conference on Software Engineering* (ICSE) – anos: 2006, 2007, 2008 e 2009). Site: <http://www.icse-conferences.org/>.

Para a população selecionada decidimos fazer um censo, onde foram observadas as respostas de todos os indivíduos da população, essa decisão foi devido a população ser pequena e onde o custo e o tempo do censo seria pouco maior que para uma amostra, outro motivo foi a representatividade dos resultados, pois através do censo teríamos uma melhor precisão.

A população selecionada foi comunicada previamente sobre a participação na pesquisa, onde enviaremos um email informando sua participação em uma pesquisa futura. Os objetivos desse aviso prévio são:

- Deixar os participantes atentos sobre sua importância na participação da pesquisa;
- Solicitar que os mesmos sugiram novos pesquisadores para participar do estudo;
- Dar a oportunidade aos participantes opinarem por não participarem do estudo.

Após a primeira seleção dos comitês de programas de conferências, foi obtida uma lista com 141 participantes, porém o estudo contou apenas com uma lista de 127 participantes que não solicitaram sua remoção do estudo após o primeiro contato.

4.2.3 Descrição da Instrumentação

Com o objetivo de avaliar uma lista de atributos e cenários de mudança que possam classificar as aplicações utilizadas em estudos empíricos sobre manutenção de software orientado a aspectos,

apoiando a seleção de qual aplicação é mais adequada para realizar um experimento específico, foi elaborado um questionário contendo a lista inicial de atributos (gerais e orientados a aspecto) e características de qualidade obtidas através de pesquisa na literatura [1]. O questionário foi respondido por especialistas no domínio, que avaliarão a importância dos atributos, além de sugerir novos atributos. A seleção da ferramenta (eSurveysPro¹) para auxiliar na distribuição do questionário via web, foi feita levando em consideração a qualidade dos serviços oferecidos pela ferramenta tanto para construção e disponibilização do questionário quanto no apoio para análise dos dados. Outras ferramentas (googleDocs², eSurveyCreator³) também foram analisadas porém não atenderam as necessidades.

4.2.4 Definição da Avaliação

A análise dos dados do estudo experimental depende da definição da avaliação, ou seja, para que se tenha uma análise com qualidade é necessário que antes se tenha uma definição da avaliação bem estruturada para ser aplicada. As opções para avaliação dos atributos e cenários de mudanças serão:

0 - Não

1 - Sim

Para saber se uma dada caracterização/classificação (Não/Sim) de um atributo de classificação e cenários de mudança [1] é confiável, é necessário ter estes atributos caracterizados ou classificados várias vezes, por exemplo, por vários especialistas.

Para descrevermos a intensidade da concordância entre os atributos definidos por Moura [1], entre dois ou mais especialistas, utilizamos a estatística Kappa [40] [41] que é baseada no número de respostas concordantes, ou seja, no número de casos cujo resultado é o mesmo entre os especialistas.

O Kappa é uma medida de concordância interobservador e mede o grau de concordância além do que seria esperado tão somente pelo acaso. Esta medida de concordância tem como valor máximo o 1, onde este valor 1 representa total concordância e os valores próximos e até abaixo de 0, indicam nenhuma concordância, ou a concordância foi exatamente a esperada pelo acaso. Um eventual valor de Kappa menor que zero, negativo, sugere que a concordância encontrada foi

¹ <http://www.esurveyspro.com/>

² <https://docs.google.com>

³ <http://www.esurveycreator.com/>

menor do que aquela esperada por acaso. Sugere, portanto, discordância, mas seu valor não tem interpretação como intensidade de discordância.

Tabela 4.1. Medidas para categorizar a concordância dos dados observados [43]

Values of Kappa	Interpretation
<0	No agreement
0 - 0.19	Poor agreement
0.20 - 0.39	Fair agreement
0.40 - 0.59	Moderate agreement
0.60 - 0.79	Substantial agreement
0.80 - 1.00	Almost perfect agreement

Essa avaliação de concordância através do Kappa é utilizada quando as escalas são categóricas e sempre quando estamos comparando dois ou mais avaliadores, neste estudo seriam os especialistas.

A estatística Kappa (k) pode ser calculada da seguinte forma:

1. Primeiro calculamos um índice que represente a percentagem de concordância esperada pelo acaso ($\text{Pr}(e)$);
2. Em segundo lugar, calculamos a concordância observada ($\text{Pr}(a)$);
3. Obtidos estes dois índices, a estatística k será calculada através da divisão da diferença entre a concordância observada ($\text{Pr}(a)$) e a concordância esperada pelo acaso ($\text{Pr}(e)$), pela diferença entre a concordância absoluta e a concordância esperada pelo acaso (a maior diferença possível entre concordância observada e esperada), conforme a Formula 4.1.

Fórmula 4.1:

$$\kappa = \frac{\text{Pr}(a) - \text{Pr}(e)}{1 - \text{Pr}(e)},$$

Os atributos e cenários de mudança sugeridos pelos especialistas serão analisados utilizando critérios qualitativos, e deverão ser avaliados em outro estudo, se necessário.

4.3 Ameaças a Validade

Esta seção discute como o instrumento da pesquisa será validado, ou seja, se o mesmo vai medir o que realmente se deseja medir na pesquisa. São cinco tipos de validade [38]. Validade de face

que é uma revisão superficial dos itens do instrumento por pessoas inexperientes. Validade de conteúdo é uma avaliação subjetiva de como o instrumento parece adequado a um grupo de pessoas com conhecimento no assunto. Essa validade geralmente envolve uma revisão sistemática do conteúdo do *survey* para assegurar que inclui tudo o que deve e se não falta nada. Esse grupo de pessoas inclui especialista, bem como membros da população-alvo. Validade de conclusão se relaciona com a habilidade de alcançar o objetivo correto sobre os dados coletados, utilizando testes estatísticos, e quão confiáveis são as medidas e esses dados. Validade de construção é responsável em assegurar que o tratamento reflete a causa e os resultados refletem o efeito. Validade externa define o quanto

4.3.1 Validade de Face

Foi apresentado o instrumento para pessoas que não tinham conhecimento no assunto avaliado pelo *survey*, com o intuito de revisar a estrutura e o design do questionário. Os participantes sugeriram algumas pequenas alterações com relação à forma de apresentação do questionário, visando facilitar o entendimento.

Também foram executados estudos pilotos, com o intuito de avaliar a compreensão das questões e se o grau de informações que o questionário disponibilizava estava adequado. Os participantes analisaram o questionário e sugeriram algumas pequenas alterações com relação à nomenclatura de expressões, visando facilitar o entendimento.

4.3.2 Validade de Conclusão

Foi utilizada a estatística para análise de concordância (Kappa) conforme 4.2.4. Através dessa validade foi possível analisar o grau de respostas concordantes entre os participantes mostrando assim a intensidade das respostas para os atributos gerais, orientado a aspectos e cenários de mudança. É importante ressaltar que não foi feita nenhuma análise com relação ao grau de significância e variância dos valores encontrados, sendo assim uma ameaça aos valores calculados.

4.3.3 Validade de Construção

Este estudo está caracterizado pelas conformidades dos atributos e cenários de mudança pertencentes à lista inicial e a lista de atributos e cenários de mudança reconhecidos como importantes pelos especialistas em estudos empíricos na avaliação e classificação de aplicações

sob o contexto de manutenibilidade de software orientado a aspectos. A lista inicial de atributos e cenários de mudança foram definidas baseando-se na literatura existente [1].

4.3.4 Validade Externa

Os participantes do estudo em geral podem ser considerados representativos para a população de especialistas na pesquisa de opinião, já que foram coletados especialistas das diversas nacionalidades e com conhecimentos específicos para o trabalho em questão conforme a seção 4.2.2, podendo assim ser generalizado dentro do seu contexto. Os dados de caracterização dos indivíduos podem ser analisados a fim de avaliar o nível de experiência no desenvolvimento de estudos empíricos no contexto do DSOA. Os objetos utilizados no estudo podem ser considerados atuais e representativos para o problema sob análise, uma vez que foram definidos tendo como base a literatura atual [1] relacionada a atributos e cenários de mudança para apoiar a classificação de aplicações em estudos empíricos sob o contexto de manutenibilidade de software orientado a aspectos.

4.4 Operação do Estudo Experimental

Inicialmente foram executados dois estudos piloto, cada um com oito participantes, com o intuito de aumentar e avaliar a compreensão das questões, avaliarem provável taxa de respostas e a eficácia dos procedimentos de acompanhamento, avaliar a confiabilidade e validade do questionário e garantir que a análise dos dados será como planejada. Os participantes responderam o questionário e sugeriram algumas pequenas alterações com relação à nomenclatura de expressões, visando facilitar o entendimento. O questionário foi então alterado. Durante a atividade de operação do estudo foram coletados 91 questionários, mas por questões como falta de preenchimento de um dos itens ou não adequação a caracterização, foram contabilizados 72 respostas para os atributos gerais, 68 respostas para os atributos orientados a aspectos e 65 respostas para os cenários de mudança na análise dos dados. O questionário final utilizado como forma de instrumentação encontra-se <http://is.gd/5FkU9> e no Anexo 1.

4.4.1 Avaliação do Estudo Experimental

Os dados obtidos foram então organizados em planilhas disponíveis pela ferramenta, utilizadas para análise e interpretação.

Foram selecionados indivíduos (não estudantes) que fazem parte de comitês de programas de conferências relacionadas com DSOA, conforme descrito no planejamento do estudo na Seção 4.2.2.

Abaixo a análise dos dados resultante do estudo é apresentada em tabelas. Também disponibilizamos no link <http://to./429d> e no Anexo 2 uma versão resumida a qual foi gerada a partir da ferramenta que utilizamos para a pesquisa.

A Tabela 4.2 apresenta a caracterização dos participantes do estudo. A maioria dos participantes são professores e pesquisadores.

Tabela 4.2. Posição dos participantes

Position	Summary of responses	
	Number	%
Professor/Lecture	58	67.44
Researcher	24	27.91
Consultant	2	2.33
Entrepreneur	1	1.16
Other: Software Engineering Manager	1	1.16

Os tipos de atividades em estudos empíricos que os participantes estiveram envolvidos estão resumidos na Tabela 4.3.

Tabela 4.3. Tipos de atividades em estudos empíricos

Type of empirical study activity	Summary of responses	
	Number	%
Planning	24	14.63
Design	42	25.61
Implementation	47	28.66
Analysis	37	22.56
None	14	8.54

Todos os atributos de classificação e cenários de mudança foram avaliados e os resultados são apresentados nas Tabelas 4.4, 4.5 e 4.6. As tabelas apresentam o número total de participantes que responderam à pergunta (sim ou não) e também a porcentagem das respostas em relação ao total das respostas.

Tabela 4.4. Resultado da avaliação dos Atributos Gerais

General Attributes	Summary of responses			
	YES	%	NO	%
System name	25	34.72	47	65.28
System domain	57	79.17	15	20.83
Versions available	57	79.17	15	20.83
Source code availability	67	93.06	5	6.94
Used programming language	67	93.06	5	6.94
Operational system and minimum computational requirements to execute	34	47.22	38	52.78
Lifecycle documentation	54	75.00	18	25.00
Software development techniques and approaches used	63	87.50	9	12.50

A Tabela 4.4 apresenta os resultados dos atributos gerais. Considerando os 8 atributos gerais da Tabela 4.4 e a categorização das respostas como Sim ou Não dada por cada um dos 72 especialistas que responderam, obtemos um valor Kappa de 0,68, ou seja, tivemos uma concordância estatisticamente diferente de zero entre os especialistas. Considerando a Tabela 4.1 o valor Kappa foi *Substantial agreement*, ou seja, houve uma concordância considerável entre os especialistas. Em termos gerais os atributos gerais tiveram taxas de respostas positivas acima de 70%.

A Tabela 4.5 mostra os atributos orientados a aspectos. Considerando os 12 atributos orientados a aspectos da Tabela 4.5 e a categorização das respostas como Sim ou Não dada por cada um dos 68 especialistas que responderam, obtemos um valor Kappa de 0,69. Considerando a Tabela 4.1 o valor Kappa foi *Substantial agreement* semelhante aos atributos gerais. Em termos gerais apenas 1 atributo teve menos de 70% de avaliação positiva.

Da mesma forma, para os cenários de mudanças que foram 5 e 65 respostas obtemos um valor Kappa de 0,67, sendo *Substantial agreement* conforme a Tabela 4.6.

Outra contribuição da pesquisa foi sugestões de novos atributos. As tabelas 4.7, 4.8 e 4.9 mostram, respectivamente, os atributos gerais, orientados a aspectos e cenários de mudança. Esses resultados devem ser considerados com cuidado, uma vez que não foram avaliados como os atributos originais, mas são definidos como atributos candidatos pelos especialistas.

Tabela 4.5. Resultado da avaliação dos Atributos Orientado a Aspectos

Aspect-Oriented Attributes	Summary of responses			
	YES	%	NO	%
The list of functional crosscutting concerns implemented as aspects	65	95.59	3	4.41
The list of non-functional crosscutting concerns implemented as aspects	65	95.59	3	4.41
The list of homogeneous crosscutting concerns implemented as aspects	59	86.76	9	13.24
The list of heterogeneous crosscutting concerns implemented as aspects	59	86.76	9	13.24
The list of Intra-Component crosscutting concerns	47	69.12	21	30.88
The list of Inter-Component crosscutting concerns.	57	83.82	11	16.18
The list of concerns with Invocation-Based Composition	42	61.76	26	38.24
The list of concerns with Component-Level Interlacing	50	73.53	18	26.47
The list of concerns with Operation-Level Interlacing	49	72.06	19	27.94
The list of concerns with Overlapping	54	79.41	14	20.59
The Aspect Scope	56	82.35	12	17.65
AO Language Constructs	52	76.47	16	23.53

Tabela 4.6. Resultado da avaliação dos Cenários de Mudança

Maintenance Scenarios	Summary of responses			
	YES	%	NO	%
Scenario description	56	86.15	9	13.85
Each change imposed by the maintenance scenario should be classified as one amongst three types: corrective, adaptive, or perfective	47	72.31	18	27.69
Each change imposed by the maintenance scenario should be classified as behavior-modifying or behavior-preserving	45	69.23	20	30.77
Change level	55	84.62	10	15.38
The Nature of the Change	55	84.62	10	15.38

Tabela 4.7. Atributos Gerais sugeridos pelos participantes

Suggested General attributes
Represent real applications (industrial / non-industrial application)
Systems generally (not be one that the researchers themselves produced)
General stability metrics Standard cohesion/coupling metrics applied to base system before injecting with aspects
Maturity of developers
Motivation and drivers for change, e.g., business needs, change in target markets

Size by LOC, number of classes, packages, methods, inheritance structure
How usable is the version system; granularity of commits, design documentation - process, methods, notations, way-of-working of developers
Formal specification
Age (old vs. new software)
Access to original design team/coding team i.e. ability to question their decision-making e.g. when they decided to use AOP/when not for parts of application productivity measures from development e.g. time to complete parts of application
Current life cycle status of the application
Open source
Formal specification
Availability of (architecture) design documentation
Presence of both a pre-AOP and post-AOP version of the same system
Variability (as for a software product line)
Part of a SPL
How much effort (people/month) a change required
General acceptance and recognition in the research community
Availability source or byte code of the precise versions used in your study
Likelihood of changes of the selected elements that involve with a concern
The presence of test units

Tabela 4.8. Atributos Orientados a Aspectos sugeridos pelos participantes

Suggested Aspect-Oriented attributes
Issues regarding conflicts of aspects that emerge in a certain domain may be of interest
Symmetric or asymmetric AO
The AO framework used to implement the aspects
Split of dynamic / static crosscutting
Nature of the used pointcuts (enumeration, pattern-based)
% of crosscutting concerns of the total, heterogeneity of aspect languages used (e.g. aspectj, ceacerj, daop-adl)
Kind of concern listed
Aspects form hierarchies, aspect depend on some other aspect / how are they dealt with

Tabela 4.9. Cenários de mudança sugeridos pelos participantes

Suggested Maintenance Scenarios attributes
Preventive change
Whether the change was of a kind that was previously anticipated or unanticipated whether the change could be made in a relative localized way, or rather did it require extensive changes throughout a system
High-level operations that aggregate lower-level change operators
Quality measures from development e.g. test pass/fails
Variability-related attributes for inclusion/removal of features
The existence of real bugs, documented across several versions
Whether evolution transformations were automatic or manual
Anticipated or unanticipated changes. - Origin of the change: business requirement, change in the technological basis, change in the domain. - How often this kind of change happens: rarely, common, frequent.

Run-time evolution
The existence of defect records/statistics
High-level operations that aggregate lower-level change operators

Finalmente, a Tabela 4.10 apresenta os resultados que a maioria dos participantes (quase 90%) concordou que os atributos propostos ou a evolução deles após o *survey* podem guiar a seleção de aplicações candidatas a um estudo empírico de orientação a aspectos.

Tabela 4.10. Resultados da avaliação de um mecanismo que guie estudos empíricos OA

Do the attributes guide empirical studies?	Summary of responses	
	Number	%
YES	54	85.71
NO	9	14.29

Conforme a Seção 4.2.4, notamos que a concordância se manteve entre os especialistas para cada tipo de atributo e cenário de mudança proposto por Moura [1], concluindo assim que os atributos e cenários de mudança são considerados comuns para a maioria de especialista notáveis em DSOA selecionados e que os dados analisados dão suporte à tomada de decisão para considerar os atributos e cenários de mudança proposto por Moura [1].

4.5 Lições Aprendidas

Algumas lições aprendidas foram obtidas através da aplicação do estudo.

- É essencial observar algumas questões sobre como conduzir um *survey*, tendo algum apoio à gestão da pesquisa para controlar a distribuição dos formulários, é indicado o uso de uma ferramenta;
- Acompanhar os entrevistados para saber, por exemplo: quem respondeu incompleto, quem não respondeu e os que não desejam participar, com isso é possível manter um contato e um controle maior sobre os participantes podendo obter o máximo de respostas.
- Planejar cuidadosamente a sazonalidade que se destina aplicar o questionário. Algumas dificuldades foram encontradas devido ao período em que foi aplicado o questionário (feriados).

4.6 Conclusão

O estudo realizado avaliou uma lista de atributos definidos no trabalho de Moura [1] e analisou a maioria deles com especialistas através de um *survey*. Os resultados mostraram que houve uma concordância dos atributos e cenários de mudança entre os especialistas, e que apenas 2 dos 25 atributos inicialmente propostos [1] não tiveram uma taxa tão alta de respostas do tipo Sim, conforme a Tabela 4.4.

O uso de uma ferramenta no estudo ofereceu uma boa gestão, porém foi observada uma particularidade na ferramenta que foi utilizada, pois a mesma contabiliza as questões respondidas mesmo sem o participante chegar ao final do questionário e submeter à pesquisa, isso é um fator importante para a avaliação do estudo, pois pode não ser interessante essa forma em particular que a ferramenta utilizada para contabilizar as respostas, pois em alguns casos só será interessante, por exemplo, as respostas serem contabilizadas quando o questionário for respondido por completo.

Através do estudo para executar o *survey*, houve a necessidade de um estudo aprofundado nas análises estatísticas, onde foram obtidas informações de um estatístico da área para ajudar na decisão de como avaliar os dados obtidos.

Também é necessário avaliar no futuro os novos atributos sugeridos pelos participantes, a fim de fazer uma avaliação semelhante deles.

Capítulo 5

Replicação de uma Avaliação de Desempenho

O intuito deste capítulo é apresentar os conceitos principais de um estudo sobre Avaliação de Desempenho em Software feito há 10 anos. Utilizamos este estudo para mostrar os passos para relatar e executar uma replicação empírica e todos os detalhes envolvidos durante o processo de replicação. São apresentadas informações sobre a replicação, comparação dos resultados obtidos no estudo original com os dados da replicação, e por fim tirar as conclusões e lições aprendidas.

A Seção 5.1 mostra os conceitos principais do estudo original feito por Soares [4] dando uma visão sobre Análise de Desempenho em Software. Na Seção 5.2 é apresentada as informações sobre a replicação. A Seção 5.3 apresenta as comparações e análises dos dados obtidos no estudo original com os obtidos na replicação. Por fim na Seção 5.4 é feita uma conclusão do que foi apresentado no capítulo.

5.1 Informações sobre o Estudo Original

São apresentados os tópicos utilizados para relatar a replicação empírica, conforme a proposta de Carver [23] apresentado na seção 2.4.2.

Para qualquer sistema computacional, em nosso caso, leia-se software, o bom desempenho é um fator que deve ser buscado durante todo seu desenvolvimento. Em sistemas computacionais um desempenho ruim implica na necessidade de mais tempo para obter-se o resultado desejado ou de computadores mais potentes para a execução do sistema. De ambas as formas estaremos envolvidos com custos associados com a manutenção de pessoal e máquinas.

Como esses custos são na grande maioria das vezes proporcionais ao tempo necessário para executar a tarefa, fica claro que um sistema com baixo desempenho significa alto custo. Sendo assim, a busca pelo desempenho se torna importante para sistemas de todo tipo, principalmente os que sejam muito utilizados ou que tenham carga computacional elevada [44].

Porém, para termos condições de dizer se um determinado software possui tal fator de qualidade (desempenho) precisamos de algumas coisas. A primeira é ter uma definição clara do que significa a palavra desempenho empregada ao contexto de engenharia de software. A segunda é termos conhecimento das ferramentas de avaliação de desempenho disponíveis no mercado e suas principais características.

Desempenho

O que é desempenho? Segundo o dicionário *Webster's: The manner in which a mechanism performs*. Segundo o dicionário Aurélio: Mil. Conjunto de características ou de possibilidades de atuação de uma aeronave, tais como velocidade de cruzeiro, velocidade de pouso, capacidade de carga, autonomia de vôo, etc.

Podemos citar como exemplo desta visão de desempenho, as medidas de desempenho de um automóvel, que seriam: velocidade máxima, aceleração (tempo para ir de 0 a 100 km/h), espaço de frenagem a uma dada velocidade e etc. Porém, em sistemas computacionais temos que ser um pouco mais específicos [44].

Medidas de desempenho em sistemas computacionais

Para medição de desempenho em sistemas computacionais, temos medidas específicas, como relacionadas abaixo:

- Vazão/Taxa (*Throughput*), refere-se a taxa na qual as solicitações são atendidas pelo sistema.
- Tempo de resposta, refere-se ao tempo decorrido desde que a solicitação é recebida até que uma resposta à solicitação seja emitida. Este aspecto de desempenho é geralmente o mais observado em um software.

Quando testamos o desempenho de uma determinada peça de software, temos que levar em conta alguns aspectos do teste que se está aplicando e do ambiente de testes em si. Um dos fatores que mais se tornam óbvios é a diferença de equipamento. A diferença de equipamento onde o mesmo software está sendo executado pode render muitas diferenças de uma performance na mudança de uma plataforma para outra. Por exemplo, isto pode ocorrer por diferença entre

processadores não só pelo fato de um processador ser mais rápido do que outro em Clock (MHz), mas também pelo fato de um dos processadores ser projetado para tratar de modo diferenciado determinados tipos de instruções [44].

Por estes e outros motivos é importante, para que os resultados tenham credibilidade suficiente, que o ambiente de testes seja mantido sob controle, inclusive evitando-se a interferência de outros aplicativos nas métricas aplicadas.

Testes de desempenho em sistemas paralelos e ou distribuídos

Como esta é hoje a área mais complexa no assunto de medições de desempenho trataremos do assunto de maneira um pouco mais aprofundada, visualizando quais métodos existe hoje para melhoria de software nesta área.

A escolha de uma técnica de avaliação de desempenho neste caso torna-se muito mais importante tendo-se em vista que os fatores envolvidos nas leituras de resultados são incrivelmente maiores comparados com os fatores envolvidos em medições de sistemas sem o uso de paralelismo.

A área de análise de desempenho de sistemas paralelos e distribuídos apresenta uma grande variedade de ferramentas e técnicas para análise. Tais ferramentas e técnicas podem ser agrupadas segundo diferentes estratégias de classificação. Isto acaba por dificultar o processo de escolha da ferramenta mais adequada para uma dada aplicação. Este processo torna-se mais complexo quando se busca ferramentas ou técnicas para análise de sistemas paralelos, quando o desempenho passa a depender também de um bom sincronismo entre os vários processos. Dentre as diversas formas para classificação de ferramentas de análise de desempenho, algumas podem ser consideradas como mais abrangentes e outras como de grande uso [44].

Os grupos de classificação destas ferramentas podem ser resumidas em:

- Métodos analíticos: são baseados em modelos matemáticos, como Redes de Petri e Cadeias de Markov. Consistem no principal meio de predição do desempenho de sistemas que ainda não estão fisicamente disponíveis. A criação de um modelo analítico preciso para determinado sistema é muito difícil, devido a complexidade do seu processo de desenvolvimento do modelo, especialmente quando se trata de sistemas paralelos.
- Os métodos baseados em simulação: se assemelham muito com os métodos analíticos (grande parte desses métodos é baseada na simulação de eventos em grafos dirigidos e em Redes de Petri). A diferença entre métodos analíticos e de simulação está na maneira como os resultados são obtidos. Nos métodos baseados em simulação, no lugar das

equações matemáticas, existem regras de comportamento que definem o comportamento dos eventos e dos estados do sistema. O grande problema no uso de simulação reside na obtenção de um modelo que seja fiel ao problema real.

5.1.1 Questionamentos

- Os questionamentos abaixo são base para o estudo original [4]: Qual impacto que sistemas baseados na Internet podem ter com relação à concorrência?
- Quais métodos de desenvolvimento em Java existem para tratar a concorrência em ambientes paralelo-distribuídos orientado a objeto?
- Os tratamentos de concorrência utilizados em sistemas orientados a objetos são feitos com base em quais evidências?
- Qual o *overhead* obtido quando utilizado os métodos de controle de concorrência existentes?

5.1.2 Participantes

O estudo original realizado por Soares não teve participação de outras pessoas fora o próprio autor, pois a análise de desempenho executada dependia apenas da execução de testes (simulações) nas máquinas utilizadas. Soares utilizou da sua experiência e orientação no seu trabalho de mestrado para a realização de testes com o intuito de medir o *overhead* em sistemas concorrentes e comparar os resultados obtidos no tratamento sugerido por ele.

5.1.3 Projeto

O estudo original realizados por Soares não teve um projeto experimental elaborado, apenas descrições do trabalho realizado sem uma padronização do que é orientado na literatura para descrever um experimento. Abaixo um resumo da descrição do estudo:

Um Método para Controle de Concorrência

O advento de sistemas de informação baseados na Internet e na World Wide Web (WWW) levou a um aumento no impacto de programas concorrentes na sociedade. Tal aumento torna necessário a definição de diretrizes que guiem o desenvolvimento seguro e eficiente de programas concorrentes, uma vez que a complexidade de desenvolvimento e testes de sistemas em ambientes concorrentes é superior a encontrada em ambientes sequenciais.

O trabalho de Soares define um método de tratamento de concorrência em sistemas orientados a objetos que dá suporte no desenvolvimento destes sistemas, evitando que sejam realizados tratamentos baseados na intuição. Além disso, a definição de um método documenta, e portanto padroniza, o tratamento a ser aplicado, favorecendo a manutenibilidade do sistema.

A definição do método de tratamento de concorrência proposto por Soares se preocupa em garantir reuso e extensibilidade, legibilidade, modularidade e por consequência manutenibilidade e produtividade (fatores de qualidade de software), através do uso de técnicas de orientação a objetos e de padrões de projeto.

O objetivo do método é permitir que classes de um sistema orientado a objetos possam ser executadas com segurança em um ambiente concorrente.

5.1.4 Artefatos

O trabalho de Soares definiu um método de tratamento de concorrência utilizando as diretrizes de controle de concorrência por ele detalhado. O método visa definir de que maneira as diretrizes apresentadas no seu trabalho devem ser utilizadas, de modo a tratar situações de concorrência indevidas, garantindo a execução correta e segura de um sistema concorrente desenvolvido segundo uma arquitetura por ele adotado. O Sistema Teste desenvolvido por Soares para simular situações de concorrências e coletar informações para suas comparações e conclusões podem ser obtidos através do link: <http://www.cin.ufpe.br/~scbs/syscoop/>.

5.1.5 Variáveis de Contexto

As variáveis de contexto que afetaram a interpretação dos resultados foram os tratamentos utilizados para cada simulação.

5.1.6 Resumo dos Resultados

É de supra importância a definição e padronização de regras que dêem suporte ao desenvolvimento de sistemas concorrentes.

Certamente, a contribuição maior do trabalho de Soares foi a definição e avaliação de regras para o tratamento de concorrência em sistemas implementados em Java com base na arquitetura e nos padrões de projetos sugeridos. Foi observado quão complexo é a análise de sistemas concorrentes devido ao elevado grau de não determinismo normalmente apresentados pelos mesmos. Sem dúvida, qualquer tratamento de concorrência que não utilize um padrão documentado é passível de erros, isto sem falar dos problemas de manutenibilidade e reuso que

podem ser ocasionados pela falta de um padrão. Os experimentos realizados no estudo original permitiram validar as diretrizes e a utilidade do método, mostrando ganhos em importantes aspectos de qualidade de software como reuso, manutenibilidade, corretude, produtividade e eficiência. Os experimentos também mostraram que o método definido por Soares pode ser utilizado para analisar sistemas já implementados, validando o tratamento de concorrência feito.

5.2 Informações sobre a Replicação

Esta seção será utilizada para explicar a replicação do experimento realizado no trabalho por Soares [4], feito há 10 anos, mostrando o impacto causado na performance de um sistema por alguns tipos de tratamento de concorrência.

5.2.1 Motivação para a Realização da Replicação

As motivações desta replicação são:

- Com novas especificações de softwares, como SGBD e com as novas especificações de hardwares, ou seja, com as máquinas atuais que temos hoje, será que o impacto de *overhead* do método proposto por Soares continua com as mesmas conclusões que ele obteve há 10 anos? Através deste questionamento a replicação têm o intuito de ampliar os resultados, sendo assim uma replicação teórica pois é destinada a produzir resultados contrastantes por razões previsíveis.
- A falta de padronização e as dificuldades encontradas para se fazer e relatar replicações de estudos empíricos na Engenharia de Software.

5.2.2 Nível de Interação com o Pesquisador do Estudo Original

A replicação contou com a participação do pesquisador do estudo original, sendo assim uma replicação interna.

5.2.3 Mudanças no Estudo Original

Para realizar os testes, foi implementado um pequeno sistema de cadastro de clientes (Sistema Teste), que permite cadastrar, procurar e atualizar objetos no estudo original [4]. Os mesmos testes sem nenhuma alteração foram utilizados na replicação do estudo.

É importante salientar que o objetivo do estudo original e da replicação é perceber qual o impacto em desempenho que os tratamentos conhecidos de controle de concorrência e o método

proposto por Soares têm. Os testes devem realizar operações de inclusão, consulta e alteração feitas em um sistema com os seguintes tratamentos de concorrência [4]:

- Nenhum tratamento: Métodos sem nenhum *synchronized* e sem implementar transações;
- Fachada sincronizada: Todos os métodos da classe fachada sincronizados;
- Fachada com transações: Todos os métodos da classe fachada implementando transações;
- Sincronização nas coleções de negócio: Método cadastrar da coleção de negócio sincronizado;
- Gerenciador de concorrência nas coleções de negócio: Método cadastrar da coleção de negócio utilizando a classe GerenciadorConcorrenca [42];
- *Timestamp*: Mecanismo de *timestamp* implementado para a classe básica Pessoa.
- Tratamento sugerido: Aplicando o método definido [4], com gerenciador de concorrência nas coleções de negócio e *timestamp* para a classe Pessoa.

Como no estudo original, a abordagem que não realiza nenhum tratamento é utilizada como referência na comparação com alguns métodos, mostrando o impacto de cada tratamento no desempenho da aplicação, pois a abordagem sem tratamento seria a ideal em termos de *overhead*. A atualização dos softwares e hardwares podem determinar os resultados obtidos em experimentos que utilizam a performance de sistemas concorrentes como o foco principal. De acordo com o trabalho de Soares, essa seção vai analisar as diferenças encontradas na replicação do estudo executado.

Na Tabela 5.1 são mostradas as especificações do hardware e do software utilizados para realização dos testes há 10 anos com as especificações do hardware e do software utilizado nesse trabalho pra replicação do estudo.

Tabela 5.1. Comparação das Especificações Gerais dos Testes

<i>Especificações gerais dos testes</i>		
	<i>Experimento [4]</i>	<i>Replicação do experimento</i>
SGBD relacional	Oracle 8.04	Oracle 10gR2
Número inicial de registro na tabela	700 registros	700 registros
Número máximo de conexões com o SGBD	12 conexões	12 conexões
Versão do JDK	JDK-1.2-V	JDK-1.6.0_10
Número de <i>threads</i> simultâneos	6, 60 e 600 <i>threads</i>	6, 60 e 600 <i>threads</i>
Número de experimentos	10 experimentos	10 experimentos

<i>Especificação do hardware de execução dos testes</i>		
Modelo	AMD K6-2 500 MHz	Intel Core 2 Duo 2,00 GHz
Memória RAM	128 MB	4,00 GB
Sistema Operacional	Windows NT	Windows Vista
<i>Especificação do hardware de execução do SGBD</i>		
Modelo	Power RS6000 39H (IBM)	Itautec Pentium III 1GHz
Memória RAM	512 MB	1 GB
Sistema Operacional	Unix AIX 4.2	Linux Fedora 4

5.3 Comparação dos Resultados

As tabelas a seguir resumem os vários testes de eficiência realizados com as diferentes variações como carga do sistema e “peso” de execução dos métodos [4]. A sigla NSD significa que a comparação não é significativamente diferente segundo a análise estatística utilizada por Soares, ou seja, estatisticamente não há impacto no desempenho do referido tratamento de concorrência. Para maiores detalhes sobre a estatística utilizada pode ser consultado o trabalho original de Soares.

- **Abordagens**

A Tabela 5.2 resume a comparação das diferentes abordagens para tratamento de concorrência em relação ao sistema sem nenhum tratamento de concorrência. Os números da tabela são percentuais, indicando quão mais lenta (overhead) é cada abordagem para o tratamento de concorrência quando comparada ao sistema sem nenhum tratamento. A tabela mostra os valores do estudo original e da replicação.

Podemos notar que no estudo de Soares as variações de carga no sistema influenciaram bem mais nos tempos de execução de métodos leves do que em métodos pesados, enquanto que na replicação do estudo notamos que essa variação de carga teve uma influência por igual. Por exemplo, se observarmos o sistema com a fachada totalmente sincronizada no experimento executado por Soares podemos notar que executando métodos leves há uma grande variação nos tempos de execução, enquanto que no estudo replicado essa variação ficou alta tanto pra os métodos leves quanto para os pesados. Também podemos notar que a variação de valores com relação ao número de *threads* foi mais alta na replicação do que no estudo de Soares, isso se deve a atualização do hardware, como visto na Tabela 5.1.

Na mesma tabela podemos ainda perceber que o mesmo *overhead* significativo que Soares obteve no seu estudo com relação ao conceito de transações em todos os métodos da fachada, na replicação também foi obtido, porém com valores menores na maioria dos casos, motivando assim o uso do método definido por Soares. As linhas que mostram o impacto causado pela aplicação do método definido por Soares, continuaram a mostrar, na replicação do estudo, que a perda é bem menor comparada as outras abordagens. Também notamos a existência de percentuais negativos durante a replicação do estudo, quando aplicado os métodos pesados, esses valores são contra intuitivo já que esperamos que qualquer tratamento aplicado possa ter um overhead quando comparado a um sistema sem nenhum tratamento, também para alguns valores que mostraram uma grande diferença entre o estudo original e a replicação (47,24 e 117,17, 41,54 e 210,08), como no tratamento da fachada sincronizada com métodos leves e 6 threads, e com métodos pesados e 60 threads respectivamente. Para esses casos são necessários estudos posteriores que possam investigar o porquê desses acontecimentos, podendo até analisar o código fonte de cada teste, já que para essa replicação não tivemos esse intuito.

Tabela 5.2. Comparação entre diferentes tratamentos de concorrência

Experimento [4]						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Fachada sincronizada	47,24	113,99	58,20	35,99	41,54	30,84
Fachada com transações	31,35	61,37	57,55	26,48	17,87	28,83
Aplicando o método	NSD	4,23	NSD	NSD	0,79	1,19
Replicação do experimento						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Fachada sincronizada	117,17	51,75	33,29	29,60	210,08	113,90
Fachada com transações	141,51	27,37	22,33	-28,51	64,61	17,69
Aplicando o método	NSD	NSD	NSD	-42,41	14,41	NSD

As Figuras 5.1 e 5.3 apresentam gráficos de barra que mostram o impacto destes tratamentos em relação ao sistema sem nenhum tratamento. Para gerar estes gráficos fizemos uma pequena alteração nos testes realizados, onde coletamos informações em mais pontos (número de threads simultâneo). O mesmo foi feito para a replicação do estudo nas Figuras 5.2 e 5.4 mostrando respectivamente que houve um overhead menor

(tanto nos métodos normais quanto nos métodos pesados) em todas as abordagens e em especial no método definido por Soares em comparação com as tabelas 5.1 e 5.3.

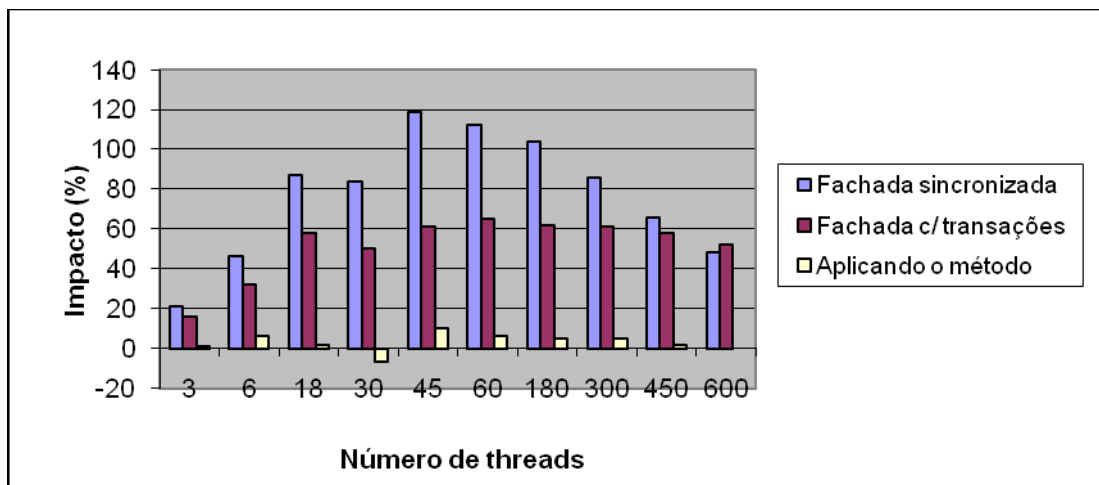


Figura 5.1 Impacto de diferentes tratamentos de concorrência [4]

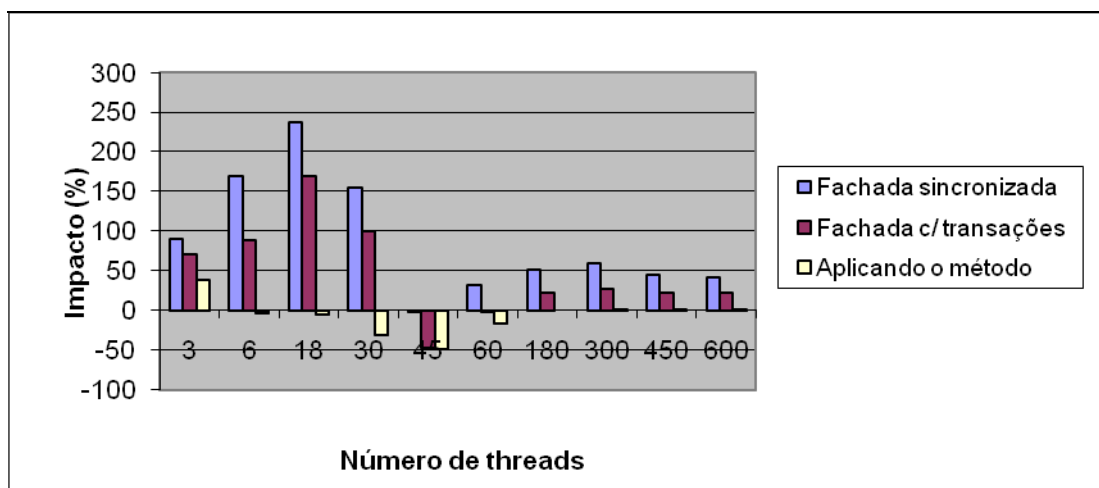


Figura 5.2 Impacto de diferentes tratamentos de concorrência na replicação

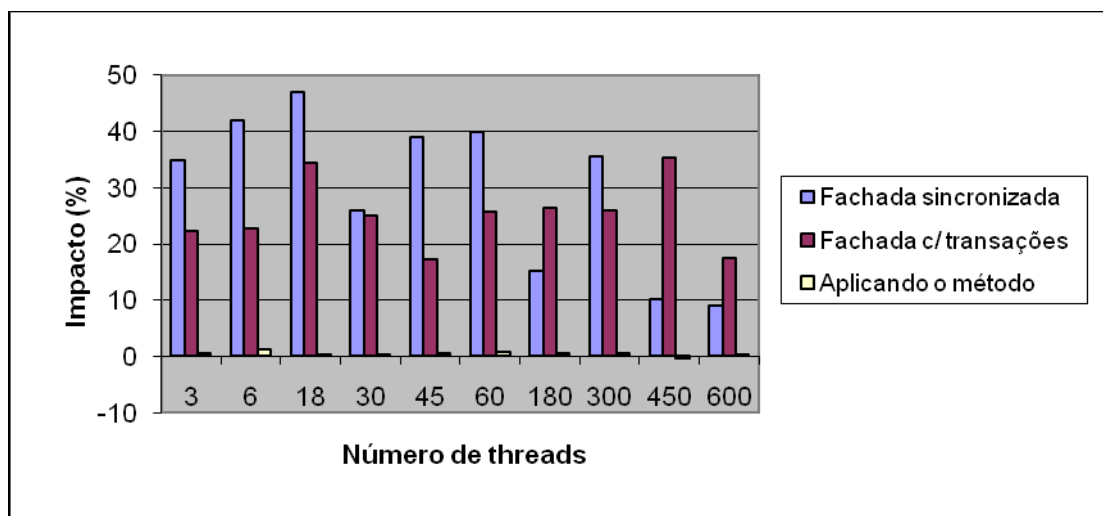


Figura 5.3 Impacto de diferentes tratamentos de concorrência com métodos pesados [4]

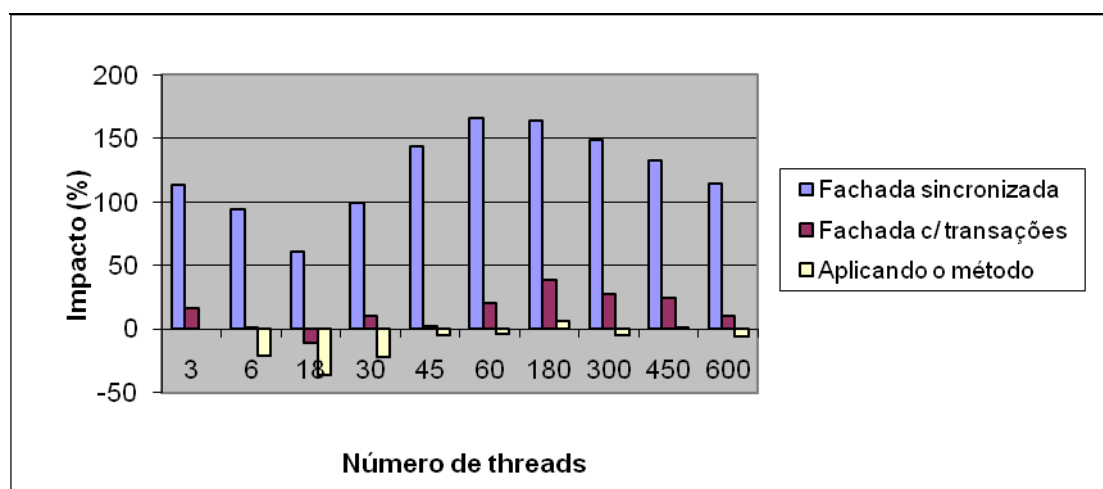


Figura 5.4 Impacto de diferentes tratamentos de concorrência com métodos pesados na replicação

- **Timestamp**

A Tabela 5.3 mostra o impacto que a implementação do mecanismo de *timestamp* ocasiona no sistema, comparando o estudo de Soares com a replicação do mesmo. Esta solução implementada isoladamente, necessariamente não garante que o sistema possa ser executado concorrentemente com segurança [4]. Podemos observar que a variação para esse mecanismo foi significativo comparando os valores obtidos por Soares [4] com os valores obtidos na replicação do estudo. Também observamos que devido à mudança de ambiente, conforme a Tabela 5.1, a replicação do estudo com o mecanismo de *timestamp*

nos métodos leves tornou-se mais rápido com 600 usuários acessando o sistema que o sistema sem nenhum tratamento. Também observamos a presença de percentuais negativos para os métodos pesados com 6 e 600 usuários acessando o sistema concorrentemente, tendo a mesma conclusão que são valores contra intuitivos e necessitam de novas avaliações.

Tabela 5.3. Comparação do impacto do mecanismo de *timestamp*

<i>Experimento Original</i>						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Implementando <i>timestamp</i>	NSD	NSD	4,03	NSD	11,37	10,15
<i>Replicação do experimento</i>						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Implementando <i>timestamp</i>	NSD	NSD	NSD	-33,84	12,30	-8,58

Similar ao que foi feito na comparação entre fachada sincronizada, fachada com transações e aplicando o método, podemos observar nas Figuras 5.5 e 5.7 o impacto de aplicar o mecanismo de *timestamp*. Podemos notar também nesse caso que o overhead diminuiu na replicação do estudo, comparando as Figuras 5.5 com 5.6 e 5.7 com 5.8. Podemos ver que o impacto do mecanismo em métodos mais pesados é maior. Isso ocorre devido ao número de canais de comunicação que é um fator limitante, neste caso. Nestes testes o número de canais de comunicação disponíveis no mecanismo de persistência é 12, conforme mostrado na Tabela 5.1.

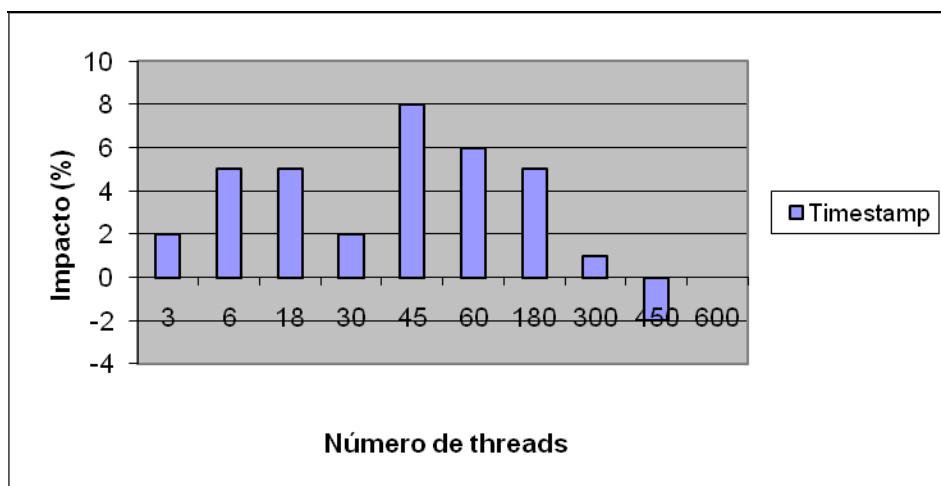


Figura 5.5 Impacto do *timestamp* [4]

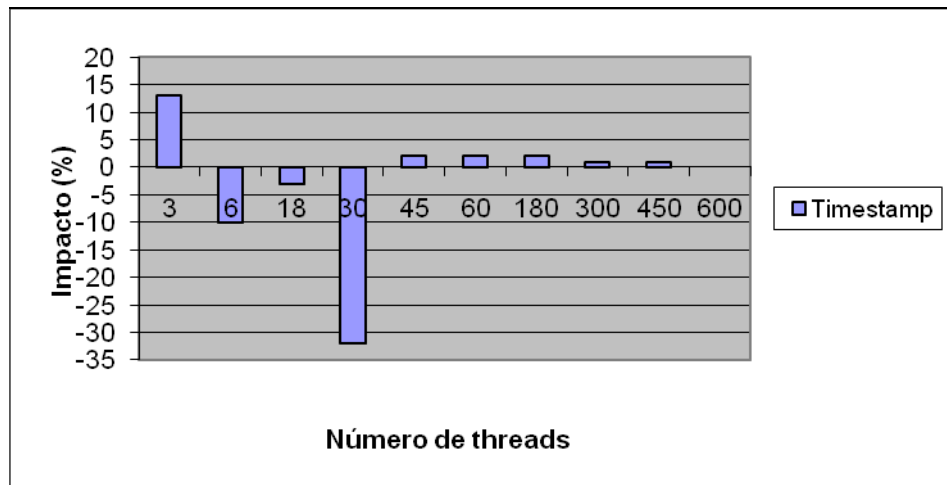


Figura 5.6 Impacto do *timestamp* na replicação

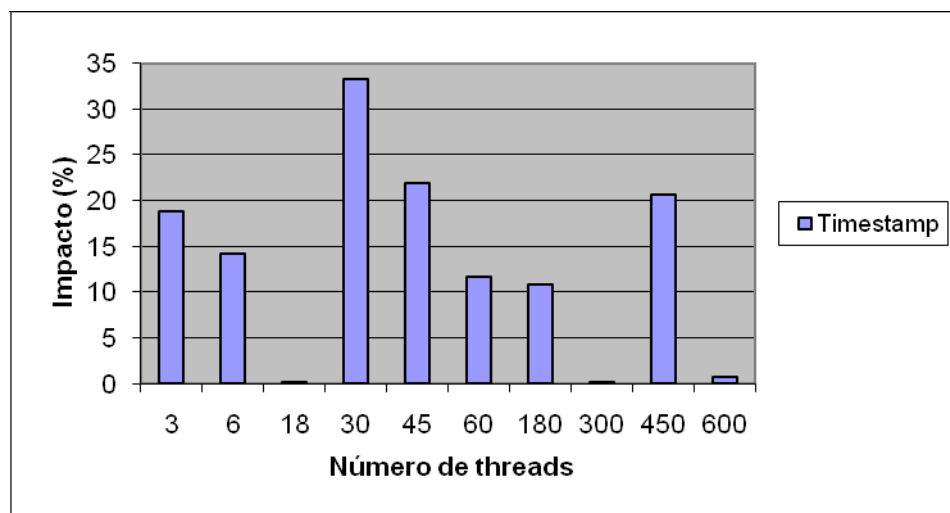


Figura 5.7 Impacto do *timestamp* com métodos pesados [4]

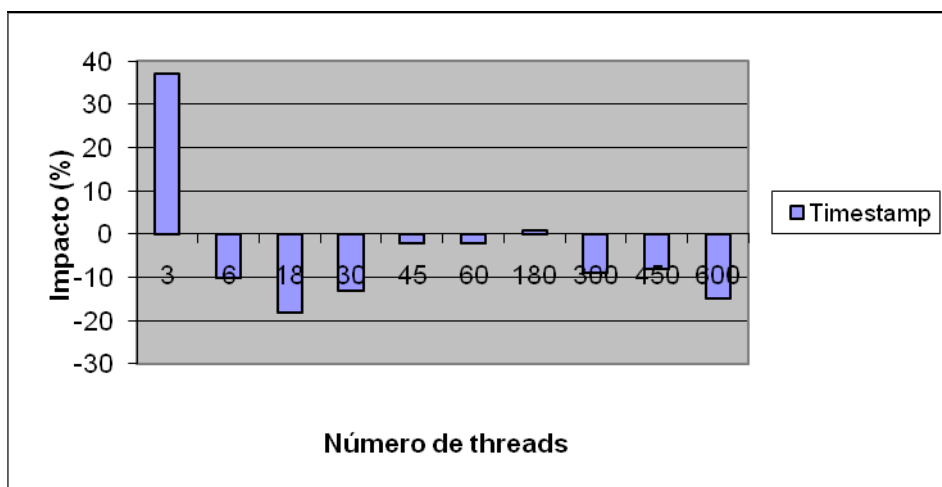


Figura 5.8 Impacto do *timestamp* com métodos pesados na replicação

- **Sincronização da Coleção de Negócio**

A última comparação feita foi entre a sincronização de métodos da coleção de negócio e a utilização do gerenciador de concorrência [4], o mesmo para a replicação do estudo. Podemos observar na Tabela 5.4 que a sincronização dos métodos na coleção de negócio é igual (não significativamente diferente) ou pior que a utilização do gerenciador de concorrência para o estudo de Soares, na a replicação do estudo o overhead aumenta ainda mais na maioria dos casos, principalmente naqueles que a situação de corrida acontece. Estudos posteriores podem analisar os códigos dos testes para verificar o porquê desse aumento de overhead, visto que para essa replicação só foi alterado os hardwares e softwares conforme a Tabela 5.1.

Tabela 5.4. Comparação de *synchronized* versus GerenciadorConcorrencia

<i>Experimento Original</i>						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Sem concorrência de objetos	4,55	63,53	16,08	7,35	15,74	5,95
Com concorrência de objetos	NSD	23,13	10,08	NSD	12,25	3,42
<i>Replicação do experimento</i>						
	<i>Métodos leves</i>			<i>Métodos pesados</i>		
<i>Número de threads</i>	6	60	600	6	60	600
Sem concorrência de objetos	34,51	19,78	13,88	74,95	68,40	44,11
Com concorrência de objetos	NSD	23,64	29,43	32,65	105,60	108,51

As Figuras 5.9 e 5.11 mostram o impacto de utilizar o qualificador de métodos *synchronized* em detrimento do gerenciador de concorrência. Também podemos ver a comparação em relação à replicação do estudo nas Figuras 5.9 com 5.10 e 5.11 com a 5.12., notando pouco ganho.

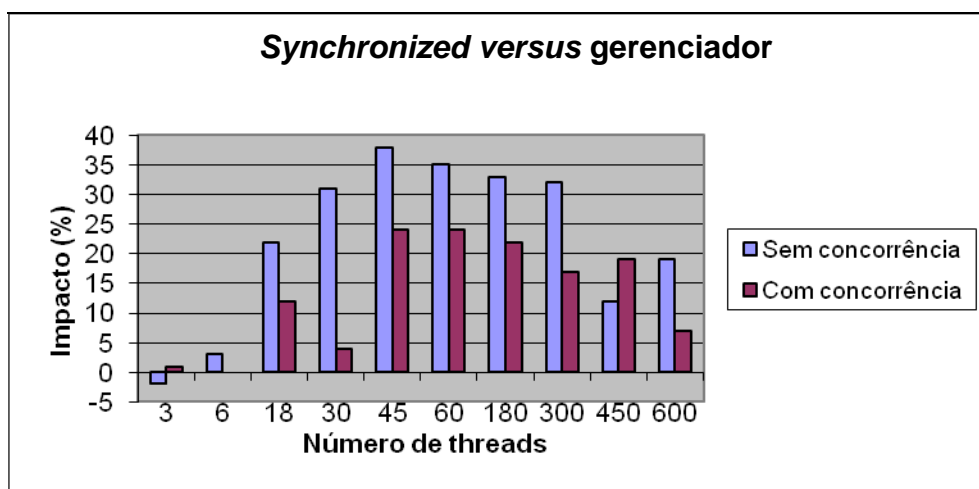


Figura 5.9 Impacto do *synchronized* em relação ao gerenciador de concorrência [4]

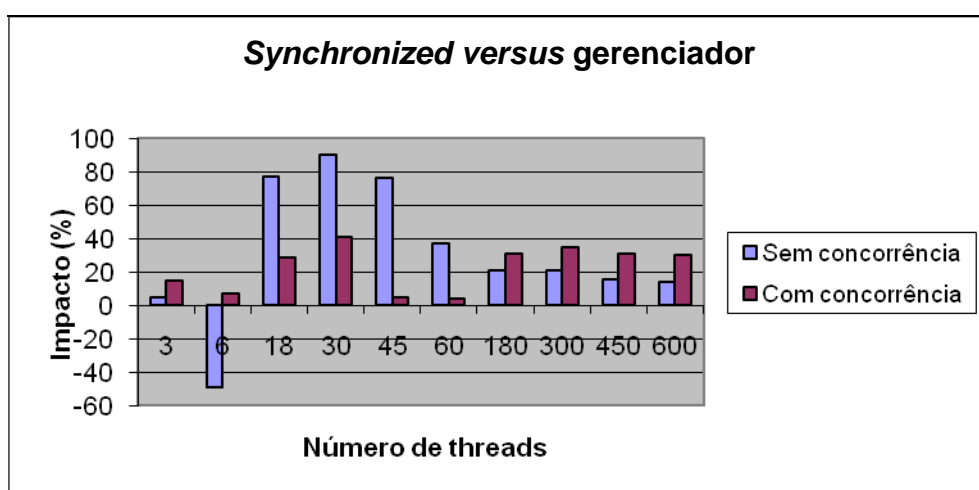


Figura 5.10 Impacto do *synchronized* em relação ao gerenciador de concorrência na replicação

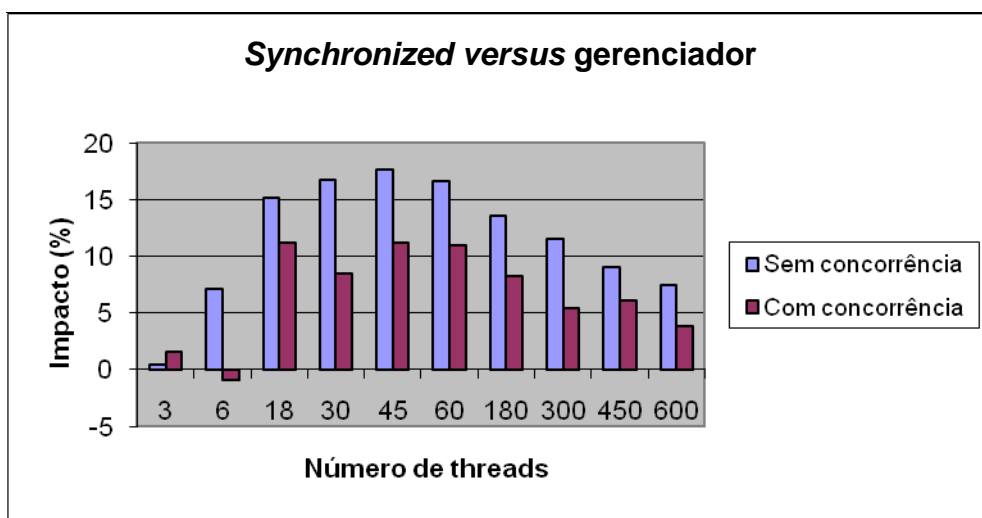


Figura 5.11 Impacto do *synchronized* em relação ao gerenciador de concorrência com métodos pesados [4]

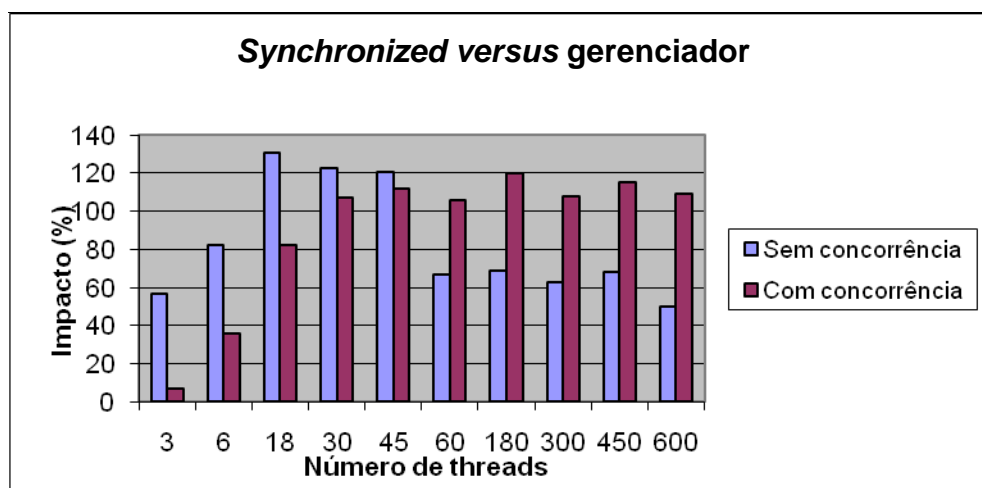


Figura 5.12 Impacto do *synchronized* em relação ao gerenciador de concorrência com métodos pesados na replicação

5.4 Lições Aprendidas

Uma das contribuições principais da replicação feita nas seções anteriores do experimento feito por Soares [4] foi analisar os resultados obtidos alterando a variável ambiente, ou seja, como

novas especificações de hardware e softwares iriam reagir a performance de sistemas concorrentes (avaliação de desempenho) de acordo com o método definido.

A facilidade encontrada na replicação foi:

- Atuação de forma efetiva do pesquisador que executou o experimento pela primeira vez, com isso foi possível analisar, adaptar, organizar de forma mais rápido o código e configurações necessárias para a replicação.

As dificuldades durante a execução da replicação foram:

- O tempo que passou desde primeira execução (10 anos);
- Configuração e adaptação do código para funcionar com o novo SGBD;
- Falta de documentação do planejamento do experimento;
- Dificuldade no ambiente físico para execução da replicação, visto que foi feito na universidade e o acesso era restrito.
- Falta de análise com mais detalhes do hardware de execução do SGBD, pois só depois que foi executada a replicação percebemos que a configuração do hardware não estava tão atualizada como era esperado, pois usamos a infra-estrutura de rede do CIn-UFPE

5.5 Conclusão

Através da replicação do estudo executado foi feita uma análise dos resultados obtidos, mostrando as diferenças dos valores do experimento feito há 10 anos com os resultados de hoje.

Considerando que os números das tabelas/gráficos para a primeira abordagem (Fachada sincronizada, Fachada com transações e Aplicando o método) são percentuais, ou seja, indica quão mais lenta é cada abordagem para o tratamento de concorrência quando comparada ao sistema sem nenhum tratamento, concluímos que nos métodos leves houve uma grande variação nos valores na replicação do estudo. Também notamos que o overhead foi menor, ou seja, mais eficiência, com uma quantidade maior de threads comparado com o estudo original de Soares. Já a aplicação do método teve uma maior eficiência na replicação do estudo, notando assim que a atualização dos hardware/software influenciou no tempo comparado com o estudo original [4]. Com os métodos pesados notamos que na replicação do estudo, o uso da "Fachada sincronizada" teve uma diminuição considerável na eficiência comprado com estudo original [4], enquanto que "Aplicando o método" teve um aumento na eficiência na replicação do estudo.

Na abordagem de *timestamps* notamos uma maior eficiência na replicação do estudo. Podemos ver que o impacto do mecanismo em métodos normais e pesados foi grande com relação ao estudo original [4].

Na comparação feita entre a sincronização de métodos da coleção de negócio e a utilização do gerenciador de concorrência, consideramos que os números das tabelas/gráficos indicam quão mais lenta é a utilização do *synchronized* quando comparada ao sistema com um gerenciador de concorrência. Notamos que tanto a utilização de corrida quanto a não utilização de corrida diminuiu bastante eficiência para os métodos pesados comparado ao estudo original.

Essas diferenças encontradas na replicação do estudo motivam um estudo mais aprofundado para que sejam detalhadas as causas com relação à atualização do ambiente comparado a implementação dos códigos testes.

Concluimos que estes experimentos permitiram validar as diretrizes e a utilidade do método [4], mostrando ganhos em importantes aspectos de qualidade na performance de sistemas concorrentes. Os experimentos e suas replicações mostraram que o método definido por Soares [4] pode ser utilizado para analisar sistemas já implementados, validando o tratamento de concorrência feito.

Capítulo 6

Conclusão e Trabalhos Futuros

Neste capítulo é apresentada a conclusão desta dissertação juntamente com os trabalhos relacionados e trabalhos futuros que podem dar continuidade a este trabalho. Na Seção 6.1 são apresentadas as conclusões, enquanto a Seção 6.2 mostra os trabalhos relacionados e na Seção 6.3 os trabalhos futuros que podem ser feitos dando continuidade a este, aprimorando e estendendo os resultados sobre estudos empíricos na Engenharia de Software.

6.1 Conclusões

Este trabalho propôs um estudo em estudos empíricos, apresentando a complexidade envolvida em tais estudos. Esta proposta visa facilitar os estudos executados, a fim de oferecer aos pesquisadores informações mais detalhadas sobre a execução de experimentos, estudos de caso e em especial *survey*. Para atingir este objetivo, primeiramente foi apresentado com detalhes todo o processo para aplicação de um *survey*, deixando evidente a necessidade de seguir os passos necessários para se obter um estudo experimental com qualidade.

Através da fundamentação teórica sobre os métodos de investigação experimental juntamente com os passos necessários para selecionar o método de investigação mais adequado para o problema em questão foi elaborado um *survey* a fim de avaliar a proposta de Moura [1], o *Benchmarking Framework* (BF), onde o mesmo propôs um conjunto de atributos e cenários de mudança [1] para classificar aplicações e serem utilizados por estudos empíricos sobre manutenibilidade de software orientado a aspectos (OA), apoiando a seleção de qual aplicação e cenários são mais adequados para um experimento específico, a partir de um conjunto de aplicativos e cenários disponíveis. Além disso, este conjunto de atributos de classificação [1]

executa outro papel de auxiliar pesquisadores a considerar outras questões em seus estudos empíricos, por expor diversas características, das aplicações e dos cenários de mudança. Através da utilização do BF também se pretende acelerar a transferência e disseminação das tecnologias utilizadas.

A proposta de Moura [1] teve avaliações simples necessitando assim de novas avaliações, pois faltam comprovações e evidências científicas suficientes que atestem sobre as vantagens e desvantagens associadas à completude e ao uso do *Benchmarking Framework*.

Através do *survey* foi possível coletar informações da realidade da experiência de especialistas da área de orientação a aspectos, buscando evidências empíricas que pudessem ajudar a avaliar o BF, ou seja, mostrar que os atributos do BF podem corresponder aos atributos abordados por vários especialistas notáveis na área e assim concluir que o BF pode ajudar como um guia na classificação de aplicações.

A definição do estudo experimental (*survey*) mostra o quanto é importante descrever o objetivo do estudo, enquanto que o planejamento mostra como o estudo foi projetado, permitindo que outros estudos sejam executados usando o mesmo plano o qual pode confirmar ou refutar alguns dos nossos resultados e derivar novos que nós não pudemos analisar. Futuros estudos podem vir a trocar algumas variáveis a fim de medir seus impactos. Também é importante analisar as ameaças à validade verificando como os resultados serão validados e se estes podem ser generalizados. Os resultados mostraram a concordância existente entre os especialistas para cada grupo de atributos: atributos gerais, atributos orientados a aspectos e cenários de mudança através a análise estatística Kappa (k), mostrando uma concordância substancial para todos os grupos de atributos, além de novos atributos serem sugeridos pelos participantes. É importante ressaltar que as análises dos dados dão suporte à decisão dos pesquisadores para considerarem os atributos e cenários de mudança do BF [1] visto que tivemos uma alta concordância entre os especialista e que apenas 2 (atributos gerais) dos 25 atributos e cenários de mudança inicialmente propostos por Moura [1] tiveram taxas abaixo de 50% como respostas positivas. Lições aprendidas foram obtidas através do estudo, mostrando a importância de seguir todo o processo definido para a elaboração de um *survey*.

Outra o objetivo deste trabalho foi realizar a replicação de um estudo feito há quase 10 anos por Soares [4]. O intuito desta parte do trabalho foi apresentar os conceitos principais de Avaliação de Desempenho, como fundamentação teórica, e por fim dá uma ênfase na replicação do experimento [4], mostrando a análise e comparação dos resultados coletados e também

algumas lições aprendidas em relação à dificuldade que os pesquisadores têm quando se trata de uma replicação.

O trabalho de Soares define um método de tratamento de concorrência em sistemas orientados a objetos que dá suporte no desenvolvimento destes sistemas, evitando que sejam realizados tratamentos baseados na intuição. O objetivo do método de controle de concorrência é permitir que classes de um sistema orientado a objetos possam ser executadas com segurança em um ambiente concorrente. Através da replicação do estudo comparamos o impacto dos resultados obtidos há 10 anos, com os resultados de hoje, onde tivemos a alteração da variável ambiente, ou seja, novas especificações do SGBD, novas versões dos softwares e novas especificações dos hardwares. Também avaliamos as facilidades e dificuldades encontradas quando é feito uma replicação de um estudo empírico.

Concluimos que estes experimentos (original e a replicação) permitiram avaliar o impacto no desempenho de sistemas concorrentes, mostrando ganhos em importantes aspectos de qualidade na performance. Os experimentos e suas replicações mostraram que o método definido por Soares pode ser utilizado para tratar sistemas já implementados, validando o tratamento de concorrência feito.

6.2 Trabalhos Relacionados

Nesta seção apresentamos os trabalhos que deram suporte aos estudos empíricos na Engenharia de Software de maneira semelhante ao que usamos.

O trabalho de Easterbrook [12] apresenta o suporte que é necessário para elaborar estudos empíricos, mostrando passos que dão suporte à decisão do pesquisador. Através do estudo de Easterbrook tivemos uma visão para a seleção da estratégia empírica que usamos no estudo do Capítulo 4. Com uso do trabalho de Moura [1], onde foi definido o *Benchmarking Framework*, fomos capazes de elaborar um novo estudo empírico com o intuito de obter mais evidências sobre a importância dos atributos do BF. É importante ressaltar que estudos de casos já haviam sido realizados com o intuito de mostrar a importância do BF. Através do trabalho de Mendonça [25] foi obtido conhecimentos sobre a execução de um *survey* e assim ver na prática o desenvolver de uma pesquisa *survey*. O trabalho de Carver [23] deu suporte e padronizou a documentação necessária para elaboração da replicação empírica, podendo, através do estudo de Soares [4], fazer uma replicação bem relatada da avaliação de desempenho por ele executada.

6.3 Trabalhos Futuros

Certamente ainda há muito que se fazer para gerar resultados no âmbito de estudos empíricos, considerando que é uma área ampla e que carece de evidência experimental na Engenharia de Software. Sendo assim, dando continuidade ao trabalho aqui exposto, novos estudos poderão surgir com a extensão do *Benchmarking Framework* [1] para outro domínio que não seja orientação a aspectos, ou mesmo para avaliar os novos atributos sugeridos pelos participantes do *survey* executado neste trabalho, pois antes de qualquer adoção é extremamente necessário que novas invenções e sugestões sejam rigorosamente avaliadas e comparadas com outras já existentes, para que os riscos associados sejam bem conhecidos.

A replicação do *survey* elaborado neste trabalho será de fácil obtenção uma vez que foi elaborado um planejamento que permite que outros estudos sejam executados usando o mesmo plano o qual pode confirmar ou refutar alguns dos nossos resultados e derivar novos que nós não pudemos analisar. Futuros estudos podem vir a trocar algumas variáveis a fim de medir seus impactos. Também ficou evidente que dos métodos de investigação experimentais apresentados o *survey* é o mais fácil para repetição e tem um custo baixo, apesar de não ter controle na execução e nem controle na medição, comparado com experimentos e estudo de casos.

Outro estudo poderá dar continuidade ao experimento replicado na avaliação de desempenho, como por exemplo, a construção de um plano experimental deixando documentado todo o processo do experimento ou ainda alterar outras variáveis a fim de obter novos resultados.

Bibliografia

- [1] Moura, M., Um Benchmarking Framework para Avaliação da Manutenibilidade de Software Orientado a Aspectos. 2008. 95 f. Dissertação (Mestrado em Engenharia da Computação) – Departamento de Sistemas e Computação, Universidade de Pernambuco, Recife, 2008.
- [2] Tichy, W. F (1998) “Should Computer Scientist Experiment More?”, IEEE Computer, 31 (5), pp. 32-39.
- [3] Gunther, H., 1999, — "Como elaborar um questionário", In: Instrumentos psicológicos: manual prático de elaboração. 1 ed. Brasília : LabPAM/IBAPP, p. 231-258.
- [4] Sérgio Soares. Desenvolvimento Progressivo de Programas Concorrentes Orientados a Objetos. Tese de Mestrado, Centro de Informática, Universidade Federal de Pernambuco, Fevereiro 2001.
- [5] Pfleeger, S. Software Engineering: Theory and Practice. Prentice Hall. 2001.
- [6] Pressman, R. S. Software Engineering: a Practitioner's Approach. McGraw-Hill Science/Engineering/Math. 2004.
- [7] Sommerville, I. Software Engineering. 7th Edition, Addison-Wesley, 2006.
- [8] IEEE, “Standard Glossary of Software Engineering Terminology”, IEEE Std 610.12-1990, 1990.
- [9] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. 2000 Experimentation in Software Engineering: an Introduction. Kluwer Academic Publishers.
- [10] Robson, C. Real World Research: A Resource for Social Scientists and Practitioners-Researchers, Blackwell, 1993.
- [11] Basili V., "The Role of Experimentation in Software Engineering: Past, Present, Future", Proceedings of the 18th International Conference on Software Engineering, 1(2), pp. 133-164, 442-449, 1996.
- [12] Easterbrook, S., Singer, J., Storey, M. A., Damian, D. Selecting Empirical Methods for Software Engineering Research. In: Guide to Advanced Empirical Software Engineering.

Springer, 2008.

- [13] Meltzoff, J. (1998) *Critical Thinking About Research: Psychology and Related Fields*. American Psychological Association, Washington DC.
- [14] Chalmers, A. (1999) *What Is This Thing Called Science?* 3rd Edition, Hackett Publishing Co, Indianapolis.
- [15] Creswell, J.W. (2002) *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*. 2nd Edition, Sage Publications, Thousand Oaks, CA.
- [16] Travassos, G., Gurov, D., Amaral, E. (2002) “Introdução à Engenharia de Software Experimental”. In: Relatório Técnico ES-590/02-Abril, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ.
- [17] Cook, T., Campbell, D. (1979) “Quase-Experimentation – Design and Analysis Issues for Field Settings”, Houghton Mifflin Company, 1979.
- [18] Kitchenham, B. (2004b) “Procedures for Performing Systematic Reviews”, Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia.
- [19] V. R. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Trans. Softw. Eng.*, pages 456{473, 1999.
- [20] B. A. Kitchenham, T. Dyba, and M. J. Jørgensen. Evidence-based software engineering. In *ICSE '04: Proc. 26th Int. Conf. on Softw. Eng.*, pages 273{281, Washington, DC, USA, 2004. IEEE Computer Society.
- [21] D. I. K. Sjoberg, J. E. Hannay, O. Hansen, V. B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A. C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Trans. Softw. Eng.*, 31(9):733{753, 2005.
- [22] F. Shull, V. Basili, J. Carver, J. C. Maldonado, G. H. Travassos, M. Mendonça, and S. Fabbri. Replicating software engineering experiments: Addressing the tacit knowledge problem. In *ISESE '02: Proc. Int. Symp. on Empirical Softw. Eng.*, Washington, DC, USA, 2002. IEEE Computer Society.
- [23] Carver, J. C. Towards Reporting Guidelines for Experimental Replications: A Proposal. In *ICSE '10: International Conference on Software Engineering, 2010 – 1st International Workshop on Replication in Empirical Software Engineering Research*, Cape Town, South Africa.
- [24] Pfleeger, S. L., Kitchenham, B. A., 2001, — "Principles of survey research: part 1: turning lemons into lemonade", *ACM SIGSOFT Software Engineering Notes*, v. 26, n.6, pp.16-18.

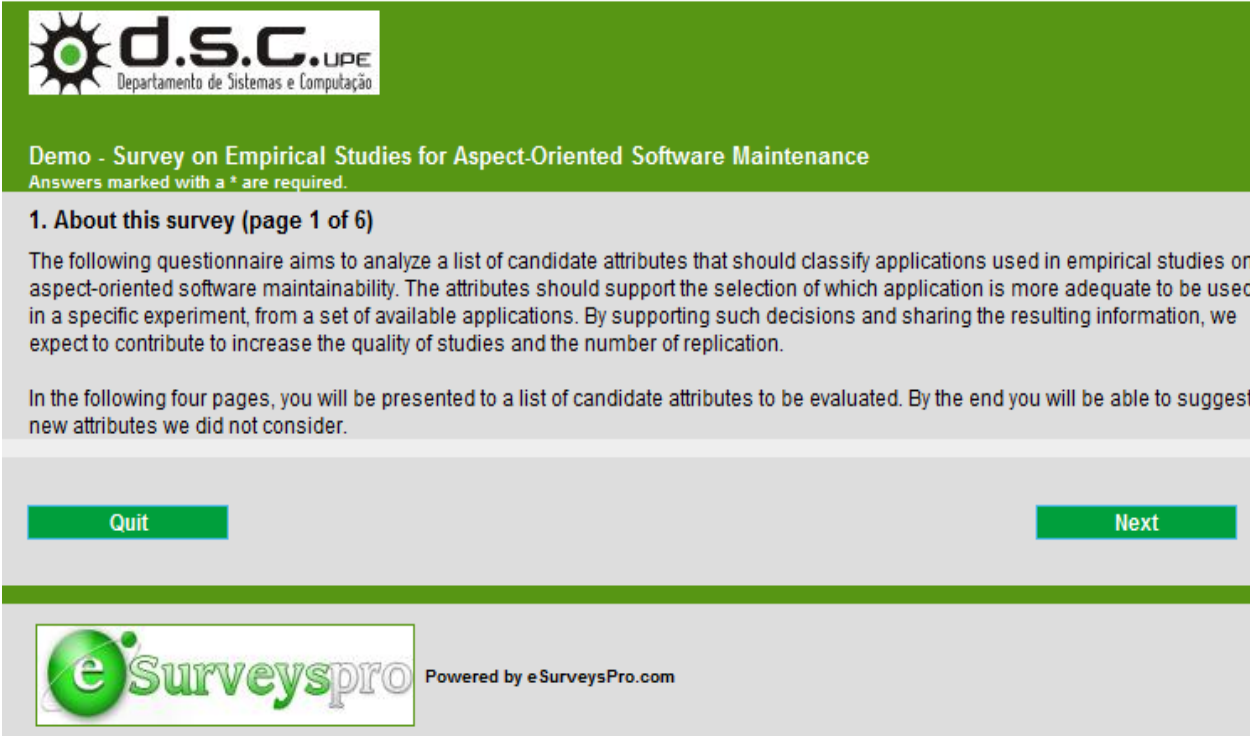
- [25] Mendonça, C. C, Uma Infra-Estrutura para Apoio ao Planejamento e Execução de Pesquisas de Opinião na WEB. 2005. 117 f. Dissertação (Mestrado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, 2005.
- [26] Gamma E., Helm R., Johnson R., Vlissides J., Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison Wesley, 1995.
- [27] Apel, S., Kastner, C., and Trujillo, S. 2007. On the Necessity of Empirical Studies in the Assessment of Modularization Mechanisms for Crosscutting Concerns. In Proceedings of the International Conference on Software Engineering Workshops (ICSEW), page 161. IEEE Computer Society Press, May 2007.
- [28] Burkey, J., Kuechler, W. L., 2003, — "Web-Based Surveys for Corporate Information Gathering: A Bias-Reducing Design Framework", IEEE Transactions On Professional Communication, v. 46, n. 2, pp.81-93.
- [29] Pinsonneault, A., Kraemer, K. L., 1993, — "Survey research methodology in management information systems: an assessment", Journal of Management Information System, v. 10, n. 2, p. 75-105.
- [30] Solomon, D. J., 2001, "Conducting web-based surveys", Practical Assessment, Research & Evaluation, 7(19).
- [31] Cook, C., Heath, F., Thompson, R., 2000, "A meta-analysis of response rates in web or internet based surveys", Educational and Psychological Measurement, 60, pp. 821-836.
- [32] Kittleson, M., 1997, "Determining effective follow-up of e-mail surveys", American Journal of Health Behavior. 21, 193-196.
- [33] Pfleeger, S. L., Kitchenham, B. A., 2002a , "Principles of survey research part 2: designing a survey", ACM SIGSOFT Software Engineering Notes, v.27, n.1, pp 18-20.
- [34] Basili, V. R., 1992, Software Modeling and Measurement: The Goal Question Metric Paradigm, Computer Science Technical Report Series, CS-TR-2956 (UMIACS-TR-92-96), University of Maryland, College Park, Md., September.
- [35] Pfleeger, S. L., Kitchenham, B. A., 2002b, "Principles of survey research part 3: constructing a survey instrument", ACM SIGSOFT Software Engineering Notes, v. 27, n.2, pp.20-24.
- [36] Pressman, R. S., 2001, Software Engineering: A Practitioner`s Approach, 5 ed., McGraw-Hill.
- [37] Pfleeger, S. L., Kitchenham, B. A., 2002c, "Principles of survey research part 4: questionnaire evaluation", ACM SIGSOFT Software Engineering Notes, v.27, n.3, pp.20-23.
- [38] Kitchenham, B. A., Pfleeger S. L. Personal Opinion Surveys. In: Guide to Advanced

Empirical Software Engineering. Springer, 2008.


- [39] Pfleeger, S. L., Kitchenham, B. A., 2002e, - "Principles of survey research part 6: data analysis", ACM SIGSOFT Software Engineering Notes, v. 28, n.2, pp.24-27.
- [40] Siegel S, Castellan N. Nonparametric Statistics for the Behavioral Sciences. 2.ed. New York: McGraw-Hill, 1988. p 284-285.
- [41] Fleiss Joseph L. Statistical methods for rates and proportions. New York: John Wiley, 1981. p 212-236.
- [42] Sérgio Soares and Paulo Borba. *Concurrency Manager*. In: First Latin American Conference on Pattern Languages Programming - SugarLoafPLoP 2001. October 3-5, 2001. Rio de Janeiro, Brazil. Published in UERJ Magazine: Special Issue on Software Patterns, June 2002, pages 221-231.
- [43] Landis JR, Koch GG. The measurement of observer agreement for categorical data. *Biometrics* 1977; 33: 159-174.
- [44] Rosa, A., Avaliação de Desempenho de Software. 2005. Monografia (Graduação em Sistema de Informação) – Universidade Federal de Santa Catarina, Florianópolis.
- [45] Basili, V. R., Selby, R. W., Hutchens, D. H. (1986) "Experimentation in Software Engineering", *IEEE Transactions on Software Engineering*, 12 (7), pp. 733-743.
- [46] Potts, C. (1993) "Software Engineering Research Revisited", *IEEE Software*, pp. 19-28, September 1993.
- [47] Fenton, N., Pfleeger, S., Glass, R (1994) "Science and Substance: A Challenge to Software Engineers", *IEEE Software*, pp 86-95, July, 1994.
- [48] Glass, R. (1994) "The Software Research Crisis", *IEEE Software*, pp. 42-47, November, 1994.
- [49] Kitchenham, B., Pickard, L., Pfleeger, S. (1995) "Case Studies for Method and Tool Evaluation", *IEEE Software*, pp. 52-62, July, 1995.
- [50] Juristo, N., Moreno, A. (2002) "Reliable Knowledge for Software Development", *IEEE Software*, pp. 98-99, sep-oct, 2002.
- [51] Ticky, W., Lukowicz, P., Prechelt, L., Heinz, E. (1995) "Experimental Evaluation in Computer Science: A Quantitative Study", *Journal of System and Software*, 28(1), pp. 9-18.
- [52] Zelkowitz, M., Wallace, D. (1998) "Experimental Models for Validating technology", *IEEE Computer*, 31(5), pp. 23-31.
- [53] Pfleeger, S. (1999) "Albert Einstein and Empirical Software Engineering", *IEEE Computer* 32(10): 32-37.

Anexo 1 – Questionário Utilizado no *Survey*

Este anexo apresenta o questionário utilizado no estudo empírico descrito no Capítulo 4 desta dissertação.



The screenshot shows a survey interface with a green header and footer. The header contains the logo for 'd.s.c. UPE Departamento de Sistemas e Computação'. The main content area is white with a green border. It displays the survey title 'Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance' and a note that 'Answers marked with a * are required.'. Below this is a section titled '1. About this survey (page 1 of 6)' which explains the purpose of the questionnaire: to analyze candidate attributes for software maintainability. At the bottom of the main content area, there are two green buttons labeled 'Quit' and 'Next'. The footer contains the 'eSurveyspro' logo and the text 'Powered by eSurveysPro.com'.

 **d.s.c. UPE**
Departamento de Sistemas e Computação


Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance
Answers marked with a * are required.

1. About this survey (page 1 of 6)

The following questionnaire aims to analyze a list of candidate attributes that should classify applications used in empirical studies on aspect-oriented software maintainability. The attributes should support the selection of which application is more adequate to be used in a specific experiment, from a set of available applications. By supporting such decisions and sharing the resulting information, we expect to contribute to increase the quality of studies and the number of replication.

In the following four pages, you will be presented to a list of candidate attributes to be evaluated. By the end you will be able to suggest new attributes we did not consider.

[Quit](#) [Next](#)

 **eSurveyspro** Powered by eSurveysPro.com



Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance

Answers marked with a * are required.

2. Participant data (page 2 of 6)

Please provide the following personal data.

All the private data will be kept confidential, and will be presented only in a form that no correspondence could be made between a participant and his/hers answers.

1. First name: *

2. Last name: *

3. Affiliation: *

4. E-mail: *

5. Position: *

 Professor/Lecture Researcher Consultant Entrepreneur Other (Please Specify)

6. What type of empirical study activity you were involved?

Select all that apply. *

 Planning Design Implementation Analysis None

Powered by eSurveysPro.com



Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance

Answers marked with a * are required.

3. General Attributes (page 3 of 6)

The following candidate attributes are non aspect-oriented attributes that classify an application. You should evaluate if the information they provide can support selecting an application for an empirical study in the context of aspect-oriented software maintainability.

7. System name *

- Yes
 No

8. System domain *

- Yes
 No

9. Versions available *

- Yes
 No

10. Source code availability *

- Yes
 No

11. Used programming language *

- Yes
 No

12. Operational system and minimum computational requirements to execute *

- Yes
 No

13. Lifecycle documentation *

- Yes
 No

14. Software development techniques and approaches used *

- Yes
 No

Quit

Back

Next



Powered by eSurveysPro.com



Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance

Answers marked with a * are required.

4. Aspect-Oriented Attributes (page 4 of 6)

The following candidate attributes are aspect-oriented specific attributes that classify an aspect-oriented application. You should evaluate if the information they provide supports selecting a representative application for an empirical study in the context of aspect-oriented software maintainability.

15. The list of functional crosscutting concerns implemented as aspects *

- Yes
 No

16. The list of non-functional crosscutting concerns implemented as aspects *

- Yes
 No

17. The list of homogeneous crosscutting concerns implemented as aspects *

- Yes
 No

18. The list of heterogeneous crosscutting concerns implemented as aspects *

- Yes
 No

19. The list of Intra-Component crosscutting concerns.

An Intra-Component crosscutting concern affects one or more points in the system in a single component (class or aspect) *

- Yes
 No

20. The list of Inter-Component crosscutting concerns.

An Inter-Component crosscutting concern affects one or more points in the system in more than one component (class or aspect) *

- Yes
 No

21. The list of concerns with Invocation-Based Composition.

Invocation-Based Composition arises if two concerns, C1 and C2, have no classes or aspects in common, i.e., they only communicate via method calls *

- Yes
 No

22. The list of concerns with Component-Level Interlacing.

Component-Level Interlacing occurs if multiple concerns have one or more components (classes or aspects) but no operation (i.e. method, advice, etc.) in common, i.e., they are tangled at the component level only. *

- Yes
 No

23. The list of concerns with Operation-Level Interlacing.

Operation-Level Interlacing occurs if multiple concerns have one or more operations in common, i.e., they are interlaced at the operation level. *

- Yes
 No

24. The list of concerns with Overlapping.

Overlapping occurs where multiple concerns share one or more statements, operations or components (aspects or classes). *

- Yes
 No

25. The Aspect Scope

The Aspect Scope attribute identifies the stage (e.g. requirements, architecture, detailed design and implementation) at which a particular concern emerges in an application. *

- Yes
 No

26. AO Language Constructs

The list of language elements used to implement a crosscutting concern *

- Yes
 No

[Quit](#)[Back](#)[Next](#)



Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance

Answers marked with a * are required.

5. Maintenance Scenarios (page 5 of 6)

The following questions are associated with candidate attributes that classifies the maintenance scenarios available for an aspect-oriented application. You should evaluate if they can support selecting an application for an empirical study in the context of aspect-oriented software maintainability.

In our context, the candidate applications will have several versions, a base version and one version for each maintenance scenario. Such scenarios are crucial to a software maintainability empirical study.

27. Scenario description

The name and description of the scenario *

- Yes
 No

28. Each change imposed by the maintenance scenario should be classified as one amongst three types: corrective, adaptive, or perfective *

- Yes
 No

29. Each change imposed by the maintenance scenario should be classified as behavior-modifying or behavior-preserving. *

- Yes
 No

30. Change level

This involves specifying the Changes at the Requirements Level, Changes at the Analysis and Design Level, and Changes to the Implementation attributes. *

- Yes
 No

31. The Nature of the Change

The change should be classified as an addition (e.g. introduction of a new method), subtraction (deletion of an attribute), or alteration (to modify an existing element) of functionality. *

- Yes
 No

Quit

Back

Next



Powered by eSurveysPro.com



Demo - Survey on Empirical Studies for Aspect-Oriented Software Maintenance

Answers marked with a * are required.

6. Suggesting new attributes (page 6 of 6)

Now that you had evaluated all the candidate attributes, you will have a chance to propose any additional attributes that you consider important for a empirical study on aspect-oriented software maintenance.

32. What new General attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

33. What new Aspect-Oriented attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

34. What new Maintenance Scenarios attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

35. According to your experience, the attributes we propose, or an evolution of them (after the survey), could guide selecting targets applications to an aspect-oriented empirical study? *

Yes

No

36. Additional comments (questions, problems or suggestions):

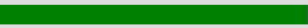




Quit


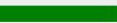
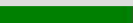


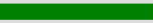
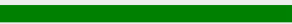
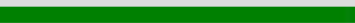



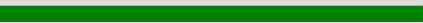





Back

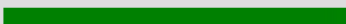











Finished


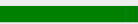


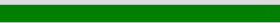

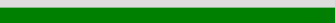



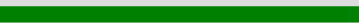



Anexo 2 – Resumo da Análise dos Dados do *Survey*

Este anexo apresenta um resumo dos dados resultantes do estudo empírico descrito no Capítulo 4 desta dissertação.

5. Position:	% of Respondents	Number of Respondents
Professor/Lecture 	67.44%	58
Researcher 	27.91%	24
Consultant 	2.33%	2
Entrepreneur 	1.16%	1
Software Engineering Manager 	1.16%	1
	<i>Number of respondents</i>	<i>76</i>
	<i>Number of respondents who skipped this question</i>	<i>15</i>

6. What type of empirical study activity you were involved? Select all that apply.	% of Respondents	Number of Respondents
Planning 	14.63%	24
Design 	25.61%	42
Implementation 	28.66%	47
Analysis 	22.56%	37
None 	8.54%	14
<i>Number of respondents</i>		76
<i>Number of respondents who skipped this question</i>		15
Page 3. General Attributes (page 3 of 6)		
7. System name	% of Respondents	Number of Respondents
Yes 	34.72%	25
No 	65.28%	47
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
8. System domain	% of Respondents	Number of Respondents
Yes 	79.17%	57
No 	20.83%	15
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
9. Versions available	% of Respondents	Number of Respondents
Yes 	79.17%	57
No 	20.83%	15
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
10. Source code availability	% of Respondents	Number of Respondents
Yes 	93.06%	67
No 	6.94%	5
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
11. Used programming language	% of Respondents	Number of Respondents
Yes 	93.06%	67
No 	6.94%	5
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
12. Operational system and minimum computational requirements to execute	% of Respondents	Number of Respondents
Yes 	47.22%	34
No 	52.78%	38
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19

13. Lifecycle documentation	% of Respondents	Number of Respondents
Yes 	75.00%	54
No 	25.00%	18
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
14. Software development techniques and approaches used	% of Respondents	Number of Respondents
Yes 	87.50%	63
No 	12.50%	9
<i>Number of respondents</i>		72
<i>Number of respondents who skipped this question</i>		19
Page 4. Aspect-Oriented Attributes (page 4 of 6)		
15. The list of functional crosscutting concerns implemented as aspects	% of Respondents	Number of Respondents
Yes 	95.59%	65
No 	4.41%	3
<i>Number of respondents</i>		68
<i>Number of respondents who skipped this question</i>		23
16. The list of non-functional crosscutting concerns implemented as aspects	% of Respondents	Number of Respondents
Yes 	95.59%	65
No 	4.41%	3
<i>Number of respondents</i>		68
<i>Number of respondents who skipped this question</i>		23
17. The list of homogeneous crosscutting concerns implemented as aspects	% of Respondents	Number of Respondents
Yes 	86.76%	59
No 	13.24%	9
<i>Number of respondents</i>		68
<i>Number of respondents who skipped this question</i>		23
18. The list of heterogeneous crosscutting concerns implemented as aspects	% of Respondents	Number of Respondents
Yes 	86.76%	59
No 	13.24%	9
<i>Number of respondents</i>		68
<i>Number of respondents who skipped this question</i>		23

19. The list of Intra-Component crosscutting concerns.	% of Respondents	Number of Respondents
An Intra-Component crosscutting concern affects one or more points in the system in a single component (class or aspect)		
Yes 	69.12%	47
No 	30.88%	21
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
20. The list of Inter-Component crosscutting concerns.	% of Respondents	Number of Respondents
An Inter-Component crosscutting concern affects one or more points in the system in more than one component (class or aspect)		
Yes 	83.82%	57
No 	16.18%	11
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
21. The list of concerns with Invocation-Based Composition.	% of Respondents	Number of Respondents
Invocation-Based Composition arises if two concerns, C1 and C2, have no classes or aspects in common, i.e., they only communicate via method calls		
Yes 	61.76%	42
No 	38.24%	26
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
22. The list of concerns with Component-Level Interlacing.	% of Respondents	Number of Respondents
Component-Level Interlacing occurs if multiple concerns have one or more components (classes or aspects) but no operation (i.e. method, advice, etc.) in common, i.e., they are tangled at the component level only.		
Yes 	73.53%	50
No 	26.47%	18
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
23. The list of concerns with Operation-Level Interlacing.	% of Respondents	Number of Respondents
Operation-Level Interlacing occurs if multiple concerns have one or more operations in common, i.e., they are interlaced at the operation level.		
Yes 	72.06%	49
No 	27.94%	19
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
24. The list of concerns with Overlapping.	% of Respondents	Number of Respondents
Overlapping occurs where multiple concerns share one or more statements, operations or components (aspects or classes).		
Yes 	79.41%	54
No 	20.59%	14
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23
25. The Aspect Scope	% of Respondents	Number of Respondents
The Aspect Scope attribute identifies the stage (e.g. requirements, architecture, detailed design and implementation) at which a particular concern emerges in an application.		
Yes 	82.35%	56
No 	17.65%	12
	<i>Number of respondents</i>	68
	<i>Number of respondents who skipped this question</i>	23

26. AO Language Constructs		% of Respondents	Number of Respondents
The list of language elements used to implement a crosscutting concern			
	Yes	76.47%	52
	No	23.53%	16
		<i>Number of respondents</i>	68
		<i>Number of respondents who skipped this question</i>	23

Page 5. Maintenance Scenarios (page 5 of 6)

27. Scenario description		% of Respondents	Number of Respondents
The name and description of the scenario			
	Yes	86.15%	56
	No	13.85%	9
		<i>Number of respondents</i>	65
		<i>Number of respondents who skipped this question</i>	26

28. Each change imposed by the maintenance scenario should be classified as one amongst three types: corrective, adaptive, or perfective		% of Respondents	Number of Respondents
	Yes	72.31%	47
	No	27.69%	18
		<i>Number of respondents</i>	65
		<i>Number of respondents who skipped this question</i>	26

29. Each change imposed by the maintenance scenario should be classified as behavior-modifying or behavior-preserving.		% of Respondents	Number of Respondents
	Yes	69.23%	45
	No	30.77%	20
		<i>Number of respondents</i>	65
		<i>Number of respondents who skipped this question</i>	26

30. Change level		% of Respondents	Number of Respondents
This involves specifying the Changes at the Requirements Level, Changes at the Analysis and Design Level, and Changes to the Implementation attributes.			
	Yes	84.62%	55
	No	15.38%	10
		<i>Number of respondents</i>	65
		<i>Number of respondents who skipped this question</i>	26

31. The Nature of the Change		% of Respondents	Number of Respondents
The change should be classified as an addition (e.g. introduction of a new method), subtraction (deletion of an attribute), or alteration (to modify an existing element) of functionality.			
	Yes	84.62%	55
	No	15.38%	10
		<i>Number of respondents</i>	65
		<i>Number of respondents who skipped this question</i>	26

Page 6. Suggesting new attributes (page 6 of 6)

32. What new General attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

1. The "size" of the application, whither measured as number of components, classes, or whatever. The smaller the application, the easier maintainability becomes.
2. Size and complexity. Size is easier to characterize, but the effort to characterize complexity is worthwhile imho.
The most important criterion in my view is that the application(s) selected for study be meaningful. By meaningful, I mean that they should represent real applications developed by a group (could be industrial or research) for a serious purpose, and that the system should not be trivial. The systems should generally not be one that the researchers themselves produced, because then questions arise as to credibility of the evaluation. Determining how well aspect-oriented constructs work depends on testing claims against the needs of "real" or at least "representative" systems. This is not to say that demonstrations of feasibility or utility in small-scale experiments is not important or meaningful: it certainly can be, but to be really convincing, most people want to see evidence of utility in the context of realistic systems.
3. General stability metrics Standard cohesion/coupling metrics applied to base system before injecting with aspects
4. The competency of the developers who developed the versions of the application
5. size
6. support for requirements change (a few more could be thought if we think in terms of reverse engineering, like legacy code, design and documentation, for example)
7. Motivation and drivers for change, e.g., business needs, change in target markets, etc.
8. size by LOC, number of classes, packages, methods. inheritance structure.
9. - how usable is the version system; granularity of commits; can changes (really) be traced to change requests?? - availability of (architecture) design documentation - process, methods, notations, way-of-working of developers -> see 36
10. I think it really depends on the study that you are developing. Most of the attributes that you suggest seems to be relevant, but they are always dependent from the study perspective that you are developing.
11. Formal specification
12. Age (old vs new software)
13. Access to original design team/coding team i.e. ability to question their decision-making e.g. when they decided to use AOP/when not etc for parts of application Productivity measures from development e.g. time to complete parts of application
Current life cycle status of the application. It would be useful, for example, for the application to be currently undergoing maintenance --- not being initially developed, and not past the end of its life. Retrospective studies can be useful, but it would often be more interesting to choose an application that is of current interest! Willingness of the maintenance team to participate in the study. It would obviously be good for the persons conducting the study to be able to interact with the actual maintainers. This may only be a willingness to respond to questions, or it might be more, such as a willingness to adopt AO techniques on the mainline or on some branch of the application development.
14. I am not sure I understand the question!
15. It would be interesting if we could have access to real applications. I believe that building "toy programs" to meet a set of requirements may create a serious bias. Moreover, the developers expertise who build the application should also be taken into account.
I think it is would be also interesting to distinguish between maintenance of end-user software and maintenance of reusable software such as libraries.
16. For evaluation of maintainability it is also important to identify the dependencies on third-party software. Porting to new versions of used libraries and platforms is one of the major sources of maintenance activities. Maybe it is already covered by the category "Domain", but it is important to see what technical domains are covered by the software (databases, user interface, distribution, and similar)
17. - industrial / non-industrial application - age of the application
18. open source or not
19. I believe the previously discussed attributes are enough.
20. Presence of both a pre-AOP and post-AOP version of the same system.
21. Variability (as for a software product line), recurrency of crosscutting concerns in several case studies (i.e. chose many case studies, all of them including the same concerns, and see if the way of composing these concerns is similar in all case studies).
22. -whether it is part of a SPL or not
23. - Empirical data about how much effort (people/month) a change required.
24. General acceptance and recognition in the research community. archived availability of source or byte code of the precise versions used in your study
25. Likelihood of changes of the selected elements that involve with a concern.
26. The extent to which the system is used in practice The size of the system (# components, LOC) The "fame" of the system (using well-known systems is usually an advantage)
27. the presence of test units

Number of Respondents 29

Number of respondents who skipped this question 62

33. What new Aspect-Oriented attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

1. Seems ok.
2. "Serious application not of the researchers' own design." "Reasonably substantial (not toy) systems"
3. Issues regarding conflicts of aspects that emerge in a certain domain may be of interest.
4. Do aspects form hierarchies, i.e. can one aspect depend on some other aspect? Are there any conflicts among aspects in the application? If yes, how are they dealt with?
5. I have no idea. This question seems very general.
6. evolution (also, I have not seen much before on modularization and encapsulation (or breaking of it))
7. symmetric or asymmetric AO.
8. see 36.
9. Maybe something related to the interaction between different aspects.
10. -
11. None.
12. I am not sure I understand the question!
13. I think the list of criteria is already quite detailed. I would maybe additionally propose to look whether there are reusable aspects. Presence of some form of dynamic aspects may also be an interesting criterion.
14. The AO framework used to implement the aspects
15. - granularity of advised code - split of dynamic / static crosscutting
16. I believe the previously discussed attributes are enough.
17. Nature of the used pointcuts (enumeration, pattern-based, ...)
18. % of crosscutting concerns of the total, heterogeneity of aspect languages used (e.g. aspectj, ceacerj, daop-adl, ...)
19. -maturity of developers
20. ?
21. To what extent an aspect-oriented approach was followed, whatever the language support (developers can come close if they think in an AO way even without an AO language). One way this could be done is to know, for each kind of concern listed, does the system actually use some AO technique to separate that kind of concern. Some studies will want systems that aggressively use AO, others might want systems that do not, to assess the potential or the implications of not having done so.

Number of Respondents 21

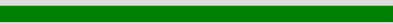

Number of respondents who skipped this question 70

34. What new Maintenance Scenarios attributes you would add to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability?

1. In question 28, "preventive change" is missing.
2. complexity of change size of change required understanding of the system
3. Whether the change was of a kind that was previously anticipated or unanticipated Whether the change could be made in a relatively localized way, or rather did it require extensive changes throughout a system
4. Scenarios where a small change in one location has a large rippling effect across other structures (e.e., where some coupling causes a challenge in maintenance due to dependencies that can be challenging in the presence of a single small change)
5. Exception handling could be represented as aspects in some scenarios.
6. High-level operations that aggregate lower-level change operators.
7. see 36
8. -
9. Quality measures from development e.g. test pass/fails etc
10. None.
11. I am not sure I understand the question!
12. An important kind of maintenance scenarios that I missed, is the maintenance resulting from evolution of used third-party software (platform and libraries).
13. - variability-related attributes for inclusion/removal of features
14. I believe the previously discussed attributes are enough.
15. - the existence of real bugs, documented across several versions
16. -Whether evolution transformations were automatic or manual.
17. - Anticipated or unanticipated changes. - Origin of the change: business requirement, change in the technological basis, change in the domain, etc. - How often this kind of change happens: rarely, common, frequent.
18. ?
19. The existence of defect records/statistics, to allow, for example, assessment of defects attributable to areas of code. An indication of whether the evolution involved adoption of AO, especially refactoring to AO. Such systems would allow comparison of before-AO and with-AO for the same system, as well as study of the adoption process.
20. run-time evolution

Number of Respondents 20

Number of respondents who skipped this question 71

35. According to your experience, the attributes we propose, or an evolution of them (after the survey), could guide selecting targets applications to an aspect-oriented empirical study?		% of Respondents	Number of Respondents
Yes		85.71%	54
No		14.29%	9
<i>Number of respondents</i>			63
<i>Number of respondents who skipped this question</i>			28
36. Additional comments (questions, problems or suggestions):			
1.	It is not clear if you are going to analyze only products, but if not, I've missed people characterization.		
2.	I went through the questions to see the entire survey, but must admit to not understanding the majority of the questions at all. I believe that you should not consider my answers in your results, as they don't represent my actual opinion on the issues.		
3.yes, they give guidance. but if the question would have been "would I only use the attributes" for selecting a target application, the answer would have been no.		
4.	I fear that the appropriate applications depend on what exactly you are trying to study, and that the generic properties you describe here would be of only limited use. Also, do you expect that both an aspect-oriented and a non-aspect version are available? This would be very rare...so the application as selected very probably would have to be expanded or alternative versions made before or as part of the empirical study... I do not mean to endorse a list of criteria for selecting candidates for case studies. The reason is that such a list could either exclude candidates that might otherwise make excellent case studies, or alternatively such a list could appear to legitimize the selection of candidates that, for reasons not on the list, are not actually very good candidates. I think that what's really most important is that researchers argue convincingly on a case by case basis that they've picked systems to study that can be expected to produce meaningful results -- and that it should be explained why.		
5.	I am not sure if I replied as you were expecting. All of the attributes that were suggested seem to have some relevance so I answered yes to most questions.		
6.	just a general comment: I am not sure if my answers were/are useful for you. My apologies about it. I tried to be realistic in my answers.		
7.	I believe that almost all the answers to the survey questions are "yes" since the most complete the system is the best. A big problem that was not mentioned on the survey was the fact that although we need complete applications to perform case studies there is a limited set of available ones.		
8.	I still couldn't use the "back" button to go back to remember what you have proposed, so I decided to close the questionnaire here. sorry.		
9.	I found the intra-component and inter-component crosscutting unclear. Hence marked my answers as no.		
10.	"Each change imposed by the maintenance scenario should be classified as behavior-modifying or behavior-preserving." This does not make sense to me... What would be the point of a change that preserves the actual behavior of the application? Perhaps you are thinking about a scenario in which the *intended* behavior is preserved, but the actual behavior is modified to match the intended behavior?		
11.	the question you ask is: "You should evaluate if they can support selecting an application for an empirical study in the context of aspect-oriented software maintainability. " now almost any information can be useful for that purpose.. it really depends on what you are trying to investigate... I think that it is "generally" not possible to specify *general* attributes to support the selection of an adequate application to be used in an empirical study on aspect-oriented software maintainability. This is because such attributes strongly depend on the particular (kind of) modification which is supposed to be investigated in such studies. In other words, I don't believe that a general statement about "maintainability" can be achieved. It is only possible to investigate the performance of aspect-orientation in respect to a particular modification (e.g. "approach/technique A is better than approach/technique B in performing this or that particular modification in this or that particular situation/setting"). And then, I would suggest to keep		
12.	all attributes fixed - except the one attribute which is to be investigated (e.g. "approach/technique A is better than approach/technique B in performing modification on heterogenous concerns" or "approach/technique A is better than approach/technique B in modifying applications in the domain Y"). Of course, then it would be nice to have a pool of applications available which involve "heterogenous concerns" or stem from "domain Y", for example, and which could be used in the study... ..and so, with that in mind, I answered this survey such that I clicked "yes" for each attribute which I image would be worth investigating "if approach/technique A is better than approach/technique B in performing modifications with respect to that attribute" - in a study on its own...		
13.	From the point of view of the study, I think you should also include the option "depends / other answer " on your study.		
14.	-		
15.	Surely the point here is that in a well formed empirical study you select independently of the attributes otherwise you bias the findings.		
16.	Sorry for my late response. I was supposed to do this before.		
17.	- I would suggest some attributes to evaluate effectiveness of XPIs		
18.	I felt the survey should have a range of values instead of yes/no for each attribute. Pretty much all the questions were a "yes" for me. But some factors are more instructive than others (so I entered a couple of "no"s for that reason).		
19.	I had trouble with understanding motivation of some questions. More detailed explanation could improve the survey. Perhaps using some examples. Good luck.		
20.	The main problem is to find existing AO applications with a significative user base... If you start with too many constraints, you will end up with a list of zero applications...		
21.	I fail to understand why you are undertaking this survey. The questions are both very abstract - giving few concrete details and no examples - and very closed, many offering only "yes/no" answers, or at least being very easy to answer "yes" to everything! It is hard to see what validity you could claim from such a survey		
22.	I said yes to most attributes, because they can provide some useful information for selecting systems for some studies. For any particular study, with a particular focus, obviously only a subset might be of value.		
23.	Yes these can be used to select applications but really the question is for a particular study what are you trying to show? It really isn't possible ot say no to any or many attributes you asked about. More information about a system is always better!		
24.	As I already mentioned in an email message to Sergio, I'm quite unsure about how to answer many of these questions. I tried to answer honestly and carefully, but I probably couldn't answer exactly the same if I did it again, simply because I don't have any hard facts or well-established principles to apply. I hope this expression of vague feelings about your questions will be helpful anyway.		
<i>Number of Respondents</i>			25
<i>Number of respondents who skipped this question</i>			66

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)