

**ENGENHARIA DE SOFTWARE EXPERIMENTAL:
UMA INTRODUÇÃO**

KLEINNER SILVA FARIAS DE OLIVEIRA

São Leopoldo, 2016

SOBRE O AUTOR

Kleinner Silva Farias de Oliveira. Doutor em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Mestre em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul (PUC-RS). Bacharel em Ciência da Computação pela Universidade Federal de Alagoas (UFAL). Tecnólogo em Tecnologia da Informação pelo Instituto Federal de Alagoas (IFAL). Atua como professor assistente no Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA) e no curso de graduação em Análise e Desenvolvimento de Sistemas da Universidade do Vale do Rio dos Sinos (Unisinos) desde 2013. Tem experiência na área de Engenharia de Software, com ênfase nos seguintes temas: modelagem de software, desenvolvimento de software dirigido por modelos, engenharia de software experimental, desenvolvimento de software orientado a aspectos, engenharia reversa e arquitetura de software.

SUMÁRIO

1 INTRODUÇÃO	4
2 ESTRATÉGIAS EMPÍRICAS.....	7
2.1 Visão Geral das Estratégias Empíricas.....	7
2.2 Surveys	8
2.3 Estudo de Caso	11
2.4 Experimentos	13
2.5 Comparação dos Métodos Empíricos.....	15
3 REPLICAÇÃO E TRANSFERÊNCIA DE TECNOLOGIA	18
3.1 Replicação de Estudos Experimentais	18
3.2 Transferência de Tecnologia Baseada em Evidências.....	20
REFERÊNCIAS	25

1 INTRODUÇÃO

Este capítulo possui como objetivo apresentar uma visão geral sobre Engenharia de Software Experimental. Para isso, o propósito de uso de experimentação em Engenharia de Software é introduzido, os principais métodos empíricos são discutidos, bem como exemplos de estudos experimentais são citados para enriquecer a discussão.

Engenharia de Software pode ser definida como sendo a "aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software" (IEEE, 1990). Vários livros têm sido escritos ao longo dos anos para sintetizar as principais áreas e campos de atuação da disciplina de Engenharia de Software. Exemplos desses livros seriam os escritos por Sommerville (2007), Pfleeger e Atlee (PFLEEGER, 2009). A Engenharia de Software Experimental, por sua vez, tem como objetivo definir como métodos empíricos podem ser utilizados no contexto de Engenharia de Software, visando melhorar técnicas, ferramentas e processos utilizados.

A Engenharia de Software se caracteriza como uma área interdisciplinar, envolvendo desde questões técnicas (tais como o uso de bancos de dados, linguagens de programação e sistemas operacionais) até questões sociais e psicológicas (WOHLIN, 2012). Ao entender que trata-se de uma disciplina muito dependente de fatores humanos, então desenvolver software depende diretamente da criatividade e do desempenho dos envolvidos no processo de desenvolvimento. O uso de Experimentação em Engenharia de Software é, por sua vez, motivado, por exemplo, pela necessidade de justificar o uso de técnicas, ferramentas e processos utilizados com um embasamento técnico-científico.

Além disso, diante da necessidade de entregar resultados com qualidade, é necessário tornar a Engenharia de Software uma disciplina quanto mais científica possível. Isso significa em ter que embasar as decisões e escolhas—tomadas ao longo do processo de desenvolvimento—justificáveis através de critérios metodológicos mais rigorosos, ao invés de critérios e argumentos subjetivos. Se critérios técnicos-científicos não são utilizados, as decisões serão influenciadas pela experiência das pessoas, geralmente sendo aquelas consideradas como *experts* ou evangelistas. Sendo assim, as decisões terão um viés subjetivo, antes que técnico-científico. É comum também acontecer que os pontos de vistas dos *experts* sejam conflitantes, caracterizados pelas divergências no entendimento, por exemplo, do uso de um determinado framework Web.

Com o objetivo de tornar a Engenharia de Software mais científica, é crucial entender os métodos científicos disponíveis, os quais são o ferramental a ser utilizado para a realização de pesquisas científicas. Esse entendimento é fundamental para identificar as vantagens e as desvantagens, bem como as limitações e características de cada método (WOHLIN, 2012). Baseado no trabalho de Basili (BASILI, 1993), Glass em (GLASS, 1994) apresenta quatro tipos (ou categorias) de métodos de pesquisa que podem ser utilizados na área de Engenharia de Software (WOHLIN, 2012): (1) *científico*, o mundo é observado e um modelo é construído com base na observação, por exemplo, um modelo de simulação; (2) *engenharia*, as soluções atuais são estudadas e as mudanças são propostas, e então avaliadas; (3) *empírico*, um modelo é proposto e avaliado através de estudos empíricos, por exemplo, estudos de caso ou experimentos controlados; e (4) *analítico*, uma teoria formal é proposta e depois comparada com observações empíricas.

Complementar as essas categorias de métodos de pesquisa, existem dois tipos de paradigmas de estudos que têm abordagens diferentes para investigações empíricas (WOHLIN, 2012). O primeiro seria a *pesquisa exploratória*, a qual está preocupada com o estudo de objetos (por exemplo, técnicas de programação) em seu ambiente natural, deixando os resultados surgirem a partir de observações. Isto implica que é necessário um projeto experimental flexível (ANASTAS, 2000) para se adaptar às mudanças ocorridas no fenômeno observado. Ou seja, à medida que o fenômeno observado ocorre, o projeto experimental pode ser adaptado (WOHLIN, 2012). Pesquisa com projeto experimental flexível é também tratada como uma pesquisa qualitativa, pois é primeiramente direcionada por dados qualitativos. Muitas vezes tratada também como uma pesquisa indutiva, visto que tenta interpretar um fenômeno baseado em explicações que as pessoas apresentam. Além disso, pesquisa exploratória preocupa-se com a descoberta de causas observadas pelos participantes do estudo e pela compreensão da sua visão do problema investigado. O participante do estudo trata-se da pessoa que faz parte do estudo empírico com o objetivo de fazer parte da avaliação do objeto investigado (por exemplo, uma linguagem de modelagem) (WOHLIN, 2012).

O segundo paradigma seria a *pesquisa explanatória* (ou explicativa), a qual tem como objetivo principal avaliar relações entre variáveis com o objetivo de identificar relacionamento do tipo causa-efeito (WOHLIN, 2012). A pesquisa é usualmente conduzida através do projeto e execução de um experimento controlado. Este tipo de pesquisa se caracteriza por um projeto experimental fixo (ANASTAS, 2000). Isso significa que as variáveis (por exemplo, esforço de desenvolvimento) investigadas são predefinidas antes que

o estudo seja inicializado. Ou seja, desde o início do estudo o pesquisador dedica-se a investigar um conjunto de variáveis e, principalmente, as relações entre elas.

Em Farias (2012), os autores buscaram entender os efeitos de técnicas de modelagem de software (tais como, modelagem orientada a aspectos e orientada a objetos) em três variáveis, incluindo esforço para localizar inconsistências em modelos, grau de compreensibilidade e taxa de detecção de inconsistência. O uso da técnica de modelagem seria a variável independente, enquanto as outras três seriam as variáveis dependentes. Em um experimento desta natureza, os pesquisadores buscam investigar a relação e os efeitos que a variável independente tem sobre as variáveis dependentes (FARIAS, 2011; FARIAS, 2013). Em outro estudo (FARIAS, 2015), os autores investigaram o impacto do uso de técnicas de composição de modelos (a variável independente) no esforço para integrar modelos de projeto, o esforço para detectar inconsistências e o esforço para resolver as inconsistências (as três variáveis independentes).

Estudos com projetos experimentais fixos são também tratados como pesquisas quantitativas, pois envolvem principalmente variáveis que podem ser quantificadas (WOHLIN, 2012). Estudos quantitativos são indicados para entender o efeito de uma variável sobre outra, como anteriormente mencionado nos trabalhos (FARIAS, 2012; FARIAS, 2015). Uma vantagem é que os dados quantitativos possibilitam comparações e análises usam métodos estatísticos.

Estudos com projeto experimental fixo (como o experimento controlado) são apropriados para testar os efeitos de um tratamento (por exemplo, uso de uma técnica de revisão de código), enquanto estudos com projeto experimental flexível são apropriados para entender porque os resultados de uma investigação quantitativa são como eles são. As duas abordagens devem ser consideradas como complementares em vez de concorrentes (WOHLIN, 2012). No próximo Capítulo, estratégias experimentais — tais como, *survey*, estudo de caso e experimento controlado — são apresentadas, bem como comparadas.

2 ESTRATÉGIAS EMPÍRICAS

Este capítulo possui como objetivo introduzir estratégias empíricas que podem ser utilizadas em Engenharia de Software Experimental. Para isso, uma visão geral das estratégias empíricas é apresentada, seguida pela descrição de cada uma das estratégias discutidas, incluindo *survey*, estudo de caso e experimento controlado. Após isso, uma comparação entre as estratégias experimentais é apresentada.

2.1 Visão Geral das Estratégias Empíricas

O tipo de estratégia (ou método) empírico a ser utilizado depende diretamente da finalidade da avaliação que se deseja fazer. Há três tipos de estratégias empíricas: *survey*, estudo de caso e experimento controlado, os quais são discutidos a seguir.

Survey pode ser definido como sendo um método para coletar informações para descrever, comparar ou explicar o conhecimento, atitudes e comportamentos de um determinado grupo de pessoas (WOHLIN, 2012; FINK, 2003). Um *survey* pode ser tipicamente uma investigação realizada em retrospectiva, quando, por exemplo, uma ferramenta ou técnica foi utilizada por um certo período de tempo (PFLEEGER, 1994). Sendo assim, busca-se coletar dados e informações sobre a experiência adquirida através do uso das mesmas. Neste tipo de estudo, entrevistas e questionários são as principais formas de coleta de dados — sendo eles qualitativos ou quantitativos. Para isso, uma amostra representativa de uma população (por exemplo, de desenvolvedores Java) é selecionada para ser estudada. Os resultados da pesquisa são então analisados para derivar conclusões descritivas e/ou explicativas (explicadas na Seção 2.2). Os resultados são então generalizados para a população da qual a amostra foi tomada.

Estudo de caso em Engenharia de Software trata-se de uma investigação empírica que se baseia em múltiplas fontes de evidência para investigar uma instância (ou um pequeno número de casos) de um fenômeno de Engenharia de Software dentro de seu contexto real, especialmente quando o limite entre fenômeno e contexto não pode ser claramente especificado (WOHLIN, 2012; RUNESON, 2012). O estudo de caso é normalmente destinado a rastrear uma variável específica ou estabelecer relações entre variáveis diferentes. Os dados coletados podem ser tratados e analisados usando métodos estatísticos. O nível de controle é menor em um estudo de caso do que em um experimento.

Um estudo de caso é tipicamente um estudo observacional, enquanto o experimento é um estudo controlado (ZELKOWITZ, 1998). Um estudo de caso pode, por exemplo, visar a construção de um modelo para prever o número de falhas nos testes de software (ANDERSSON, 2007). De acordo com Wohlin (2012), análise estatística multivariada é frequentemente aplicada neste tipo de estudos. Os métodos de análise incluem regressão linear e análise de componentes principais (MANLY, 1994). Estudo de caso tem sido explorado e investigado por vários pesquisadores, os quais disponibilizam uma série de trabalhos discutindo tópicos sobre estudo de caso (WOHLIN, 2012), tais como: Robson (2002), Stake (1995), e Yin (2009), e especificamente para engenharia de software por Pfleeger (1994), Kitchenham (1995), Verner (2009), e Runeson (2012).

Experimento controlado em Engenharia de Software (WOHLIN, 2012) trata-se de uma estudo empírico que controla variáveis em um determinado cenário estudado. Baseado em randomização, diferentes participantes executam tratamentos ou atividades inerentes ao estudo. Busca-se entender os efeitos de uma (ou mais) variável independente(s) sobre variáveis dependentes. Desse modo, tenta-se entender como a variável independente influencia os valores assumidos pelas variáveis dependentes. Em experimento com pessoas, os participantes usam diferentes técnicas, tecnologias ou processos para um determinado propósito. Em experimento com tecnologia, o pesquisador usa técnicas, ferramentas ou processos para um determinado propósito (WOHLIN, 2012).

2.2 Surveys

Surveys podem ser planejados e aplicados antes do uso de uma determinada técnica ou ferramenta, ou mesmo depois do uso da mesma (PFLEEGER, 1994). São utilizados para capturar o *status* atual sobre o entendimento de determinadas questões sobre uma técnica, processo, ou ferramenta, bem como podem ser usados em pesquisa de opinião. Uma fábrica de software poderia, por exemplo, planejar e executar um *survey* com o objetivo de entender como a adoção de um novo processo de desenvolvimento melhorou as ações e atitudes dos desenvolvedores em relação à garantia de qualidade, ou mesmo a priorização de atributos de qualidade (WOHLIN, 2012; KARLSTROM, 2002).

Para colocar essa pesquisa em prática, um questionário poderia ser elaborado, visando revelar quais práticas do processo estimularam a mudança de atitude dos desenvolvedores, levando, conseqüentemente, ao ganho de qualidade. Além disso, o survey

poderia ter também questões que permitissem identificar como os benefícios com a adoção do novo processo foram percebidos. Uma vez que o questionário tenha sido elaborado, uma amostra representativa dos desenvolvedores da fábrica de software precisaria ser selecionada para responder o questionário elaborado. Os dados coletados com o questionário serão, então, organizados e analisados quantitativamente e/ou qualitativamente (WOHLIN, 2012).

Surveys são raramente executados para criar um entendimento de uma amostra em particular; ao contrário, eles são executados com o propósito de criar um entendimento da população, de onde uma determinada amostra foi derivada. Essa é uma das características mais importantes dos *surveys* (WOHLIN, 2012). Suponha que uma empresa com 600 desenvolvedores, organizados em times globalmente distribuídos, buscase compreender os benefícios da adoção de um novo processo de desenvolvimento de software. Para isso, seria possível identificar uma amostra representativa desta população de 600 desenvolvedores ao selecionar, por exemplo, 50 destes desenvolvedores. Sendo assim, com a análise dos dados coletados com os questionários respondidos com os 50, espera-se revelar conclusões que sejam aplicáveis à população contendo os 600 desenvolvedores. Isto é, os *surveys* buscam produzir conclusões generalizáveis ao permitir derivar uma análise de uma amostra de 50 pessoas, porém gerando conclusões válidas para uma população de 600.

É importante também destacar que *surveys* têm a capacidade de dar suporte a um grande número de variáveis em estudos experimentais reais. Exemplos de variáveis seriam a percepção das vantagens e desvantagens do uso de um determinado processo, a produtividade de desenvolvedores, a precisão da estimativa de esforço de desenvolvimento de software. Para isso, várias questões precisam ser elaboradas para que os participantes possam opinar sobre cada questão formulada. Porém, não recomenda-se que um *survey* tenha um número elevado de questões, visto que isso pode torná-lo tedioso para responder, o que pode levar a um impacto negativo na qualidade dos dados coletados. Sendo assim, deve-se buscar um entendimento maior e mais detalhado sobre um número menor de variáveis e, conseqüentemente, do número de questões, o que favorecerá a qualidade da coleta e análise dos dados (WOHLIN, 2012).

Um *survey* pode ser realizado com um dos seguintes propósitos (BABBIE, 1990): descritivo, explanatório, exploratório. Um *survey* com o propósito descritivo visa coletar dados para permitir a elaboração de afirmações sobre uma população. Essas afirmações são elaboradas a partir da determinação de distribuições dos dados coletados de certas características ou variáveis observáveis da população. Esse tipo de *survey* não busca explicar

as distribuições, ou mesmo justificar as causas que levaram àquelas distribuições; ao contrário, o propósito é, de fato, gerar uma distribuição (WOHLIN, 2012).

Um *survey* explanatório, por sua vez, busca coletar e analisar os dados com o objetivo de gerar explicações sobre a população. Por exemplo, uma empresa — estudando como seus desenvolvedores usam uma determinada técnica de revisão de código — poderia tentar explicar porque a maior parte dos desenvolvedores preferem uma técnica, enquanto a menor parte prefere uma outra técnica. Explicações sobre os dados coletados podem ser produzidas ao examinar os relacionamentos entre as técnicas analisadas e variáveis que foram quantificadas (ou mesmo avaliada qualitativamente) através do questionário.

Por fim, o *survey* exploratório pode ser utilizado como um estudo inicial, sendo um primeiro passo em direção a investigações mais aprofundadas. Ele busca garantir que a coleta de dados de questões importantes (porém não previstas) possa ser realizada. Esse tipo de *survey* pode ser elaborado através de um questionário, o qual não necessariamente precisa ter uma estrutura rigorosa. Os dados gerados com as respostas de uma determinada população serão analisados e ajudarão no projeto de futuras investigações mais detalhadas.

De acordo com (WOHLIN, 2012), as duas formas mais frequentes de coletar dados é através de questionários e entrevistas. Os *questionários* podem ser elaborados em papel ou eletronicamente. Atualmente, há várias aplicações que dão suporte à elaboração dos questionários. Exemplos dessas aplicações seriam o Google Doc¹, Survey Monkey², Qualtrics³, entre outras aplicações. Tipicamente, os questionários são enviados via e-mail ou disponibilizados em uma página web, juntamente com as instruções para respondê-lo. O questionário eletrônico pode facilitar a distribuição e coleta dos dados, porém pode correr o risco de uma baixa taxa de adesão. Por outro lado, as entrevistas guiadas por um questionário oferecem algumas vantagens, tais como uma taxa de resposta maior, as respostas tendem a ser mais completas e o entrevistador pode direcionar as questões de acordo com o andamento das respostas do entrevistado. As desvantagens são o custo e o tempo para aplicar as entrevistas, pois, dependendo do tamanho da amostra, a execução das entrevistas em um curto espaço de tempo pode ser inviabilizada. Estudos mais aprofundados sobre *surveys* podem ser encontrados em Fink (2003), Robson (2002) e Wohlin (2012).

¹ Google Docs: <https://docs.google.com>

² Survey Monkey: <https://www.surveymonkey.com/>

³ Qualtrics: <https://www.qualtrics.com/>

2.3 Estudo de Caso

De acordo com Wohlin (2002), um estudo de caso é conduzido para investigar uma entidade, variável ou fenômeno em seu contexto real dentro de um espaço de tempo específico. Usualmente, o fenômeno observado encontra-se muito integrado ao seu ambiente, o que pode ser difícil de distinguir claramente do ambiente. O pesquisador coleta informações detalhadas sobre, por exemplo, um único projeto durante um período de tempo adequado. Durante a realização de um estudo de caso, diferentes procedimentos de coleta de dados devem ser aplicados (RUNESON, 2012).

Um estudo pode ser definido como sendo um estudo de caso ou um experimento, dependendo da escala de avaliação, do nível de controle e isolamento das variáveis envolvidas na pesquisa, bem como da viabilidade da execução de randomização. Um estudo de caso pode ser planejado e executado para avaliar, por exemplo, a produtividade de desenvolvedores de software após adoção de um novo processo de desenvolvimento. Essa análise pode ser feita comparando os dados gerados com dados históricos, isto é, uma linha base (do inglês, *baseline*) (KITCHENHAM, 1995).

Em Wohlin (2012), os autores destacam que estudos de caso são adequados para estudos empíricos na indústria para avaliar métodos, ferramentas e processos de desenvolvimento de software, visto que eles podem evitar problemas de escalonamento. A principal diferença entre estudos de caso e experimentos controlados é que, enquanto no experimento as variáveis investigadas são selecionadas e controladas sistematicamente, no estudo de caso as variáveis investigadas são aquelas que representam uma situação típica do contexto onde se executa o estudo (WOHLIN, 2012).

Uma vantagem dos estudos de caso é que eles são mais fáceis de planejar e são mais realistas. Por outro lado, as desvantagens são que os resultados coletados e analisados são difíceis de serem generalizados e interpretados. A principal razão dessa dificuldade é pelo fato das variáveis investigadas serem analisadas em uma situação típica do contexto no qual ela é analisada. Consequentemente, a especificidade do contexto de onde os dados foram coletados exigirá uma análise mais refinada para viabilizar a generalização da situação investigada com outras situações (YIN, 2009).

Várias áreas da ciência como a sociologia, a medicina e a psicologia (WOHLIN, 2012) têm utilizado estudo de caso como um método padrão para realizar estudo empíricos. Na

Engenharia de Software, os estudos de caso não devem ser usados apenas para avaliar como ou por que certos fenômenos ocorrem, mas também para avaliar as diferenças entre, por exemplo, dois processos de desenvolvimento, técnicas de refatoração de código, linguagens de programação, ambientes de desenvolvimento, entre outros exemplos. Isto significa, em outras palavras, avaliar quais dos dois métodos são mais apropriados para uma determinada situação (YIN, 2009; WOHLIN, 2012).

Estudo de caso pode ser utilizado como uma estratégia empírica para realizar comparações, permitindo comparar os resultados do uso de um método, técnica, ferramenta ou processo em relação a outro investigado. Porém, para que essa comparação seja, de fato, válida, é necessário remover qualquer viés, objetivando que a análise comparativa não favoreça qualquer uma das abordagens analisadas. Caso algum viés exista, a validade interna do estudo será comprometida. Um estudo de caso pode ser aplicado como uma estratégia de pesquisa comparativa, ao permitir comparar os resultados do uso de um método em relação ao uso de outro método (WOHLIN, 2012). Para evitar o viés e para garantir a validade interna, é necessário criar uma base sólida para avaliar os resultados do estudo de caso.

De acordo com (KITCHENHAM, 1995), há várias maneiras de organizar o estudo para viabilizar a comparação, visando minimizar possíveis vieses. A primeira seria comparando os resultados obtidos em relação a uma linha base previamente registrada. Por exemplo, constantemente as empresas estão ajustando seus processos para se adequarem às novas demandas de projetos, os quais possuem um número cada vez mais elevado de requisitos complexos. Além disso, tais requisitos devem ser implementados em um menor espaço de tempo, ao mesmo tempo que a qualidade da implementação não seja comprometida. Sendo assim, uma empresa pode coletar dados e resultados produzidos ao adotar um processo de desenvolvimento. Posteriormente, esse processo pode ser adaptado, produzindo novos dados e resultados. Dessa forma, é possível avaliar a adaptação do processo ao comparar os resultados obtidos antes e depois da adaptação (WOHLIN, 2012).

A segunda maneira seria usando projetos semelhantes para propósito de comparação. É possível desenvolver projetos similares para viabilizar a comparação dos dados obtidos. A similaridade é fundamental para tornar os resultados obtidos comparáveis. Sendo assim, ao realizar um estudo de caso seguindo essa estratégia, deve-se prestar atenção nas características dos projetos, as quais devem ser semelhantes; quanto mais similares as características forem, melhor será a comparação. Por exemplo, ao comparar os benefícios obtidos com duas linguagens de programação em projetos distintos (porém

similares), deve-se garantir que os projetos sejam comparáveis, em termos de número de desenvolvedores, complexidade dos requisitos, experiência dos desenvolvedores, processo utilizado, tempo do projeto, entre outras características.

2.4 Experimentos

Experimentos devem ser utilizados quando deseja-se controlar a situação, o contexto e as variáveis de forma direta, precisa e sistemática. Além disso, geralmente experimentos exigem que dois ou mais tratamentos sejam executados para permitir a comparação dos resultados gerados (WOHLIN, 20012). Se for possível controlar quem usará uma técnica e quem usará outra técnica, como tais técnicas serão manipuladas, e onde as técnicas serão utilizadas, então há condições da realização de um experimento.

O controle em um experimento pode ser feito em uma situação *off-line*, por exemplo, em um laboratório sob condições controladas, onde os eventos são organizados para simular situações reais, ou *on-line*, neste caso, a avaliação é feita usando configurações reais (BABBE, 1990), por exemplo, as encontradas em uma fábrica de software. A abordagem *on-line* tem algumas vantagens, incluindo o uso de configurações reais das variáveis controladas, a utilização de código e modelos com tamanhos e complexidades reais, entre outras. Porém, o nível de controle é mais difícil em uma situação *on-line*, pois alguns fatores são controláveis enquanto outros não.

Por exemplo, seria indicado executar um experimento para entender se o uso da UML (*Unified Modeling Language*) (OMG, 2015), oferece ganhos significativos na compreensibilidade de componentes arquiteturais, em comparação à linguagem de modelagem orientada a aspectos (GUIMARÃES, 2010; FARIAS, 2010). Em Farias (2012), os autores executaram um experimento controlado para comparar os benefícios do uso de modelagem orientada a aspectos, em comparação ao uso da UML, na detecção de inconsistências e no esforço de detecção de tais inconsistências. O experimento contou com dois tratamentos. No primeiro tratamento, os participantes executaram cinco atividades. Cada atividade consistia em indicar uma possível implementação para um diagrama de classes e um diagrama de sequência, com notações representando conceitos de orientação a aspectos. No segundo tratamento, os participantes executaram também cinco atividades. Cada atividade também exigia que os participantes fornecessem uma possível implementação para um diagrama de classes e um diagrama de sequência da UML, porém,

neste caso, os diagramas tinham notações de conceitos de programação orientada a aspectos.

Os experimentos podem ser de dois tipos: *orientados a humanos* ou *a tecnologias*. Em experimentos orientados a humanos, os participantes aplicam diferentes tratamentos aos objetos que estão sendo investigados, por exemplo, duas técnicas de revisão de código. Por outro lado, em experimentos orientados a tecnologia, usualmente diferentes técnicas são aplicadas a diferentes objetos, por exemplo, duas ferramentas de geração de casos de teste são aplicadas a um mesmo código de uma aplicação (WOHLIN, 20012). Destaca-se que o experimento orientado a humanos tende a ter um menor nível de controle, comparado ao orientado a tecnologia, pois os participantes se comportam de maneira diferente em ocasiões diferentes, enquanto as técnicas e ferramentas utilizadas no experimento orientada tecnologias são determinísticas. Além disso, devido aos efeitos de aprendizagem, os participantes de um experimento não podem aplicar duas técnicas de inspeção de código ao mesmo código, o que duas técnicas ou ferramentas podem fazer sem viés (WOHLIN, 2012).

Considerando a diferença entre estudo de caso e experimento, ela se torna ainda mais clara ao considerar a definição de contexto. Exemplos de contexto diferentes podem ser a área de aplicação e o tipo de sistema (PETERSEN, 2009). De acordo com Wohlin (2012), em um experimento, os pesquisadores podem selecionar os contextos de interesse, suas variáveis e as amostras que serão utilizadas. Isto significa dizer que algumas características típicas de um ambiente real de uma organização são selecionadas para definir o contexto no qual uma determinada técnica, método ou ferramenta será avaliada. Exemplos de contextos seriam os tipos de sistemas utilizados, as linguagens de programação utilizadas nos sistemas, entre outros.

Por exemplo, seria possível investigar o impacto de um método de inspeção na identificação de falhas em dois sistemas diferentes, os quais foram implementados em duas linguagens de programação diferentes. Desse modo, os diferentes sistemas são o contexto de avaliação do método de inspeção. Portanto, objetos semelhantes são necessários na experiência; caso contrário, a comparação poderia ter algum viés indesejado. Um projeto experimental deve ser conduzido de tal forma que os objetos investigados (por exemplo, duas técnicas de modelagem de software) representem o que se deseja estudar.

De acordo com Wohlin (2012), os experimentos são apropriados para diferentes finalidades (SJØBERG, 2008), incluindo: (1) *confirmar teorias*, testando teorias existentes; (2) *confirmar a conhecimento convencional*, testando o conhecimento amplamente difundido

entre as pessoas; (3) *explorar relacionamentos*, testando se uma determinada relação acontece ou não; (4) *avaliar a precisão dos modelos*, explorando se a precisão de certos modelos é, de fato, a esperada; (5) *validar métricas*, avaliando se uma determinada técnica realmente mensura aquilo a que ela foi destinada. A força de um experimento é que ele pode investigar em que situações as hipóteses formuladas podem ser validadas. Além disso, o uso de experimento pode fornecer um contexto no qual certos padrões, métodos e ferramentas são recomendados para uso. O planejamento e a execução de um experimento exigem que várias etapas e passos sejam executados, incluindo a definição de escopo, planejamento, operação, análise e interpretação dos dados, e apresentação dos resultados (WOHLIN, 2012).

2.5 Comparação dos Métodos Empíricos

Os pré-requisitos de um estudo experimental influenciam o tipo de método de pesquisa a ser utilizado. Exemplos de quatro pré-requisitos (ou fatores) seriam, por exemplo, o nível de controle de execução, o nível de controle da medição das variáveis, o custo da investigação, bem como o grau de dificuldade ou facilidade de replicação do estudo. A Tabela 1 apresenta uma comparação dos métodos empíricos, considerando esses quatro fatores (PFLEEGER, 1994).

Tabela 1 – Comparação das estratégias empíricas.

Fatores	<i>Survey</i>	Estudo de caso	Experimento
Controle de execução	Não	Não	Sim
Controle de medição	Não	Sim	Sim
Custo da investigação	Baixo	Médio	Alto
Facilidade de replicação	Alta	Baixa	Alta

Fonte: (Wohlin, 2012)

Os quatro fatores são discutidos a seguir:

- *Controle de execução*: este fator se refere ao grau de controle que o pesquisador tem sobre o estudo. Considerando o estudo de caso, por exemplo, os dados são coletados durante a execução de um projeto. Se os

gestores de uma empresa decidem interromper o projeto estudado, devido, por exemplo, a razões econômicas, o estudo de caso em execução terá que ser finalizado. Diante do conhecimento que as empresas de desenvolvimento de software têm uma alta rotatividade das equipes, ao executar um estudo de caso, os membros de um time de desenvolvimento podem ser alterados, sem qualquer controle por parte dos investigadores. Essa alteração na equipe pode ter motivações diversas, variando desde uma simples realocação de desenvolvedores em novas equipes a demissões. Situação inversa é encontrada ao executar um experimento controlado, no qual os investigadores possuem o completo controle da execução do estudo, minimizando qualquer ameaça em relação ao completa execução do estudo.

- *Controle da medição*: o controle de medição é o grau em que o pesquisador pode decidir quais variáveis serão controladas e como elas serão coletadas. O pesquisador poderá incluir ou excluir determinadas variáveis durante a execução do estudo. Um exemplo seria a coleta de dados sobre a variável que quantifica o grau de volatilidade dos requisitos de uma aplicação em constante evolução ou manutenção. Durante a execução de um *survey*, não é possível incluir este tipo de quantificação da volatilidade, porém em um estudo de caso ou em um experimento é possível incluí-la. Isso é porque no *survey* os dados coletados são relacionados com a opinião das pessoas sobre a volatilidade dos requisitos.
- *Custo da investigação*: dependendo da estratégia escolhida, o custo para a execução da investigação é diferente. Em parte, esse custo está diretamente relacionado com o tamanho da investigação e a necessidade de recursos. Observando a Tabela 1, observa-se que o *survey* é o método com menor custo, uma vez que não requer uma grande quantidade de recursos. Em segundo lugar vem o estudo de caso, e o experimento como sendo método com o maior custo para execução. Pode-se dizer, desta forma, que quanto maior o controle de execução, maior será o custo da investigação.
- *Facilidade de replicação*: a replicação do estudo realizado é um fator fundamental, visando a generalização dos resultados. Em outras palavras, o objetivo da replicação é verificar se os resultados obtidos em uma investigação também são encontrados em um outro estudo realizado com outra amostra ou contexto. Dessa forma, busca-se investigar, por exemplo, se

um estudo inicial conseguiu produzir resultados que também são válidos para uma população maior. Uma replicação atingirá o seu objetivo se tanto o projeto experimental como os resultados forem replicados. Por exemplo, um pesquisador poderia executar um experimento controlado com os estudos de uma universidade A e, posteriormente, executar o mesmo experimento com outros estudantes de uma universidade B. Neste caso, se os resultados obtidos forem iguais, tem-se uma replicação (ou similaridade) de projeto e resultado do experimento.

Outro aspecto relacionado à replicação, no sentido de que se trata de estudos ao longo do tempo, são estudos longitudinais (RAINER, 2011). A principal diferença entre um estudo longitudinal e uma replicação é que um estudo longitudinal é conduzido principalmente com os mesmos participantes e uma replicação, por sua vez, é essencialmente um estudo realizado com novos participantes. Em outras palavras, a replicação significa executar vários estudos, enquanto um estudo longitudinal trata-se de um único estudo.

O estudo longitudinal é conduzido durante um certo período de tempo. Por exemplo, um *survey*, como um estudo longitudinal, pode ser executado várias vezes, bem como o estudo de caso pode ser executado ao longo de um período de tempo. Por outro lado, o experimento é replicado, não sendo executado ao longo do tempo. Um estudo longitudinal é normalmente conduzido para compreender, descrever ou avaliar algo que muda ao longo do tempo (ROBSON, 2002).

Diante do exposto, a escolha da estratégia empírica depende dos pré-requisitos para a investigação, da sua finalidade e dos recursos disponíveis. Além disso, depende também da maneira como se deseja analisar os dados recolhidos (WOHLIN, 2012). Em (EASTERBROOK, 2008), Easterbrook fornece mais orientações sobre quais métodos de pesquisa utilizar. O leitor deve estar ciente de que o limite entre os diferentes tipos de métodos de investigação nem sempre é claro ou evidente. Por exemplo, um estudo de caso comparativo também pode ser tratado como um quase-experimento em um contexto considerando configurações reais. Um estudo sobre os benefícios de técnicas de modelagem de software com alunos, pode também ser considerado como um experimento (WOHLIN, 2012).

3 REPLICAÇÃO E TRANSFERÊNCIA DE TECNOLOGIA

Este capítulo possui como objetivo descrever o papel que a replicação de estudos experimentais desempenha para aumentar a confiança na generalização dos resultados obtidos. Além disso, um modelo de transferência de tecnologia é apresentado, o qual pode ser visto como um guia para o desenvolvimento e transferência de tecnologia baseada em evidência.

3.1 Replicação de Estudos Experimentais

O objetivo central na replicação de estudos consiste na produção de novos resultados a partir da repetição da execução de um projeto experimental em condições semelhantes. A produção de novos resultados é importante para aumentar a confiança da análise dos resultados produzidos até então. Wohlin (2012) reforça que produzir novos dados ajuda a descobrir o grau de confiança que se deve depositar em relação aos resultados do experimento. A confiança nos dados gerados depende de quão bem feita foi a randomização e a seleção dos participantes. Se a randomização e a seleção estiverem sido feitas corretamente, então os participantes foram representativos de uma população maior. Sendo assim, os dados coletados no experimento, bem como nas suas replicações, terão uma chance muito alta de serem semelhantes. Se resultados semelhantes não foram produzidos, então não foi possível capturar todos os aspectos, características do contexto e variáveis que afetaram os resultados (WOHLIN, 2012).

A replicação, em termos gerais, se dará através da execução de um projeto experimental, previamente já validado e executado, em um novo contexto. Sendo assim, para que replicações produzam resultados comparáveis, é necessário ter um controle rigoroso no planejamento experimental, a fim de que todas as variáveis (por exemplo, tipo de linguagem de programação utilizada, tamanho dos sistemas usados, nível de experiência dos participantes) possam ser identificadas e controladas. De acordo com Juristo (2011) e Shull (2008), as replicações podem ser de tipos diferentes, incluindo as replicações exatas e as diferenciadas.

As *replicações exatas* seguem rigorosamente os procedimentos do experimento definidos previamente. Essa modalidade de replicação busca sempre manter os procedimentos definidos, com o objetivo de reduzir a influência que alterações nos procedimentos experimentais podem causar nos resultados (WOHLIN, 2012). Sendo assim,

entende-se que os resultados obtidos, a partir de replicações que seguiram os mesmos protocolo, terão maiores chances de serem comparáveis. Uma vez que os resultados são comparáveis, serão identificadas oportunidades de generalização dos resultados em relação às populações que os participantes representam. A segunda modalidade de replicação seria a *diferenciada*. Ao contrário das replicações exatas, as quais buscam manter as questões de pesquisa e características do contexto, a *replicação diferenciada* mantém apenas as questões de pesquisa e adota outros procedimentos experimentais (WOHLIN, 2012). Essa variação nos procedimentos adotados previamente pode ser motivada por vários fatores.

Um primeiro fator de variação seria a localização dos participantes. Os procedimentos de um experimento pode exigir que todos os participantes recebam um treinamento presencial, visando garantir que todos os participantes tenham um conhecimento comum e nivelado dos temas abordados no estudo. Porém, por razões da localização dos participantes, o treinamento não será fornecido para todos os participantes de forma presencial, ou mesmo alguns dos participantes não receberá o treinamento previamente projetado (WOHLIN, 2012). A principal razão da não adoção dos procedimentos para todos os participantes é que frequentemente os participantes recrutados estão em diferentes universidades, ambientes de trabalho, ou mesmo diferentes cidades e países.

Um segundo fator seria o uso dos recursos adotados para a execução do experimento. Um estudo pode investigar o grau de compreensibilidade de uma linguagem de programação (por exemplo, Java) em uma IDE (*Integrated Development Environment*) (por exemplo, Eclipse) que exige uma licença especial. Dependendo do local da execução do experimento, por exemplo, em uma empresa ou universidade que não possua uma licença para a IDE previamente utilizada. Embora o foco da investigação seja entender o impacto de uma linguagem de programação, os recursos visuais oferecidos pelas IDE podem influenciar na compreensibilidade.

Uma vez que IDE diferentes podem fazer uso de diferentes recursos visuais (por exemplo, destaque da sintaxe (do inglês, *syntax highlighting*), a compreensibilidade pode ser influenciada, visto que algumas dimensões cognitivas podem influenciadas pela forma que as notações da linguagem de programação são exibidas para os participantes. Em (GREEN, 1996), os autores destacam algumas dessas dimensões cognitivas, tais como gradiente de abstração.

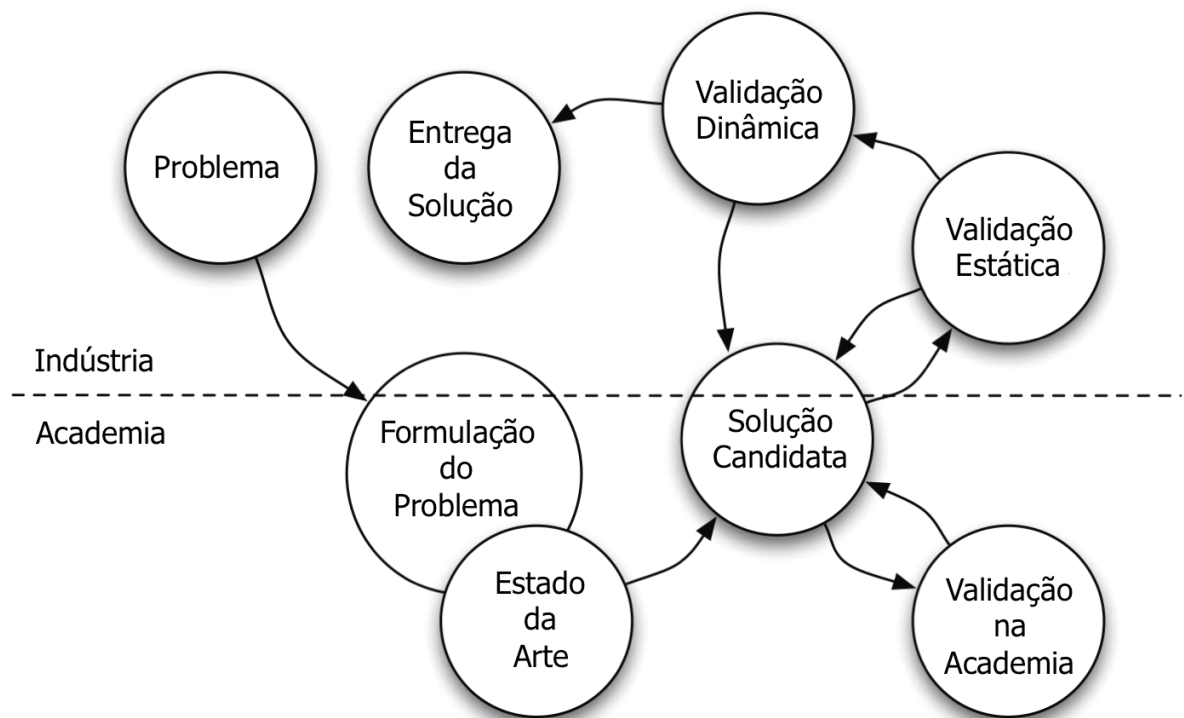
Ambas as replicações possuem vantagens. Um das vantagens das replicações exatas é que as variáveis conhecidas são mantidas sob controle, potencializando a confiança no resultado. Sendo assim, as replicações exatas, às vezes, exigem que os mesmos pesquisadores conduzam o estudo, pois eles têm conhecimento tácito sobre os procedimentos experimentais que dificilmente podem ser documentados (SHULL, 2004). No entanto, existe um risco substancial considerando o viés do experimentador em estudos de replicação exata (KITCHENHAM, 2008). Além disso, há questionamentos se replicações na área de Engenharia de Software podem ser classificadas como exatas, pois muitas variáveis podem variar na configuração complexa de um experimento controlado em Engenharia de Software (JURISTO, 2011).

Por outro lado, as repetições diferenciadas podem ser utilizadas para estudos mais exploratórios. Se as diferenças de fatores e configurações são bem documentadas e analisadas, mais conhecimento pode ser obtido a partir de estudos replicados. De acordo com (JURISTO, 2011), algumas das variáveis a considerar e relatar em estudos usando replicações diferenciadas são (WOHLIN, 2012): local onde a experiência é conduzida, experimentadores responsáveis por conduzir o experimento, o projeto experimental escolhido para o experimento, as variáveis medidas, os voluntários que participaram do experimento, a instrumentação utilizada, incluindo os formulários, as ferramentas usadas, entre outras variáveis.

3.2 Transferência de Tecnologia Baseada em Evidências

Estudos realizados conjuntamente entre a indústria e a academia representam uma excelente oportunidade para melhorar a forma como software é desenvolvido. Essas melhorias podem ser produzidas através da transferência de conhecimento da academia para a indústria (e vice-versa) (KITCHENHAM, 2004) através da realização de conhecimento empíricos (WOHLIN, 2012). O conhecimento prático e as evidências empíricas produzidas potencializam a elaboração de novos métodos, tecnologias e ferramentas, as quais estão em sintonia com as necessidades práticas da indústria, bem como respeitam critérios metodológicos rigorosos. A Figura 1 mostra um modelo para transferência de tecnologia proposto por Gorschek em (GORSCHKE, 2006).

Figura 1 – Modelo de transferência de tecnologia.



Fonte: adaptada pelo autor de Wohlin (2012).

O modelo é composto por sete etapas, as quais mostram como diferentes métodos empíricos podem ser utilizados conjuntamente na geração de conhecimento e transferência de tecnologias. As sete etapas são descritas a seguir (WOHLIN, 2012).

Identificação do problema. A primeira etapa consiste na identificação de problemas reais enfrentados na indústria. Para isso, recomenda-se que pesquisadores estejam em contato direto com os desafios enfrentados no contexto onde eles acontecem. Isso implica que pesquisadores precisam estar e participar da realidade da indústria, o que pode ser viabilizado através de projetos de pesquisa e colaboração. A identificação dos desafios pode ser feita usando questionários, entrevistando os profissionais da indústria, observando as atividades realizadas diariamente, entre outros métodos (WOHLIN, 2012). O objetivo é capturar os desafios e, em particular, as questões que são adequadas para a realização de uma pesquisa. O pesquisador deve focar na identificação de problemas reais, ao mesmo tempo que garanta que eles sejam mapeados para problemas de pesquisa, os quais serão explorados em profundidade. Deve-se evitar, desse modo, trabalhar com

desafios que não representam desafios de pesquisa, os quais podem ser resolvidos com uso de técnicas existentes em um curto prazo de tempo. Além disso, essa etapa busca também estabelecer um vínculo de confiança entre os profissionais da indústria e os pesquisadores. Isso é possível ao evidenciar possíveis benefícios para ambas as partes com a realização da pesquisa (WOHLIN, 2012).

Formulação do problema. Com base nos desafios identificados, o próximo passo é definir um problema de pesquisa e as questões de pesquisa. Se forem identificados vários desafios diferentes, há a necessidade de priorizar quais abordar primeiro. Além disso, recomenda-se definir um canal de comunicação entre os pesquisadores e os profissionais da indústria ao definir um contato principal para o desafio escolhido. Esse contato desempenha um papel fundamental ao permitir que os pesquisadores tenham acesso aos sistemas, documentação e dados quando necessário. Como parte natural da formulação do problema de pesquisa, os pesquisadores realizam uma pesquisa bibliográfica. Isso pode ser feito como uma revisão sistemática da literatura, *survey*, ou mapeamento sistemático da literatura (GONÇALES, 2015). Por exemplo, em Gonçalves (2015), os autores realizam um mapeamento sistemático da literatura sobre comparação de modelos de software. É necessária uma pesquisa bibliográfica para conhecer as abordagens existentes para o desafio industrial identificado. Ele fornece uma base para a compreensão da relação entre as abordagens disponíveis e as reais necessidades industriais.

Solução candidata. Com base nas abordagens disponíveis e nas necessidades reais, uma solução candidata é desenvolvida, a qual pode incluir adaptação aos processos, métodos, tecnologias e ferramentas correntes utilizados na empresa. De preferência, a solução deve ser desenvolvida de forma colaborativa com os integrantes da empresa parceira, a fim de que a aplicabilidade do que se encontra em desenvolvimento seja assegurada (WOHLIN, 2012). Embora uma solução específica para uma empresa possa ser desenvolvida, a intenção do pesquisador é frequentemente desenvolver uma solução genérica, a qual poderá atender às necessidades de várias empresas através da instanciação da solução para contextos específicos.

Validação na academia. Um vez que uma solução candidata tenha sido desenvolvida, o próximo passo é validá-la. Para isso, recomenda-se que a primeira validação da solução proposta seja preferencialmente conduzida em um ambiente acadêmico para minimizar possíveis riscos. Dado que a solução proposta possa ter ajustes a serem feitos, os quais só serão identificados após a primeira validação, o preço do uso de uma solução com imperfeições pode ser alto em um ambiente real. Uma consequência direta da adoção de

uma técnica sem ter passado por uma validação rigorosa seria a elevação do risco de insucesso de projetos (WOHLIN, 2012). Através de um estudo acadêmico será possível identificar pontos de melhorias com baixo custo. Esta avaliação pode ser feita por meio de um estudo de caso, ou mesmo um experimento controlado com estudantes e os profissionais da indústria em um ambiente controlado (WOHLIN, 2012).

Validação estática. Na validação estática, os representantes da indústria avaliam a solução candidata *off-line*. Isto é, sem utilizá-la em projetos reais. Essa validação pode ser feita através de uma apresentação da solução candidata seguida por entrevistas com diferentes representantes da empresa, preferencialmente tendo pessoas com diferentes papéis que utilizarão a solução. Além disso, recomenda-se também fazer uma apresentação geral à organização para torná-los cientes da solução proposta nesta fase inicial. Isso pode ser feito, por exemplo, através de um workshop (WOHLIN, 2012). Essa primeira apresentação dará oportunidade a todos de se manifestarem em relação ao que está sendo proposto como solução. Destaca-se também que a validação estática visa permitir reduzir possíveis resistências futuras na adoção da solução proposta. Com base na validação estática, dois possíveis desdobramentos podem acontecer. O primeiro seria que a solução candidata foi validada, sendo, portanto, encaminhada para a próxima etapa, na qual buscará avaliar a solução através de outras perspectivas, fazendo uso de critérios mais objetivos. O segundo desdobramento seria a identificação de pontos de melhorias na solução candidata que precisará ser ajustada. Após a implementação das melhorias, uma nova validação estática será executada (WOHLIN, 2012).

Validação dinâmica. Uma vez que a solução candidata tenha passado pela validação estática, a próxima etapa é a validação dinâmica. Nesta etapa, busca-se avaliar a solução candidata através de um estudo piloto, o qual permitirá verificar a aderência da solução às necessidades reais. A forma como a avaliação será executada dependerá do tipo de solução proposta. Se pontos de melhorias forem identificados, a solução candidata passará por melhorias a fim de que uma nova solução candidata possa ser gerada (WOHLIN, 2012).

Entrega da solução. Após a solução candidata passar pela avaliação dinâmica, o próximo passo é entregar a solução gerada. Para isso, a solução gerada será implantada na empresa, sendo necessário o acompanhamento dos pesquisadores a fim de que a instanciação da solução possa ser feita de forma correta (WOHLIN, 2012).

O modelo de transferência mostrado na Figura 1 ajuda também a entender como diferentes estratégias empíricas podem ser aplicadas para viabilizar transferência de resultados de pesquisa, a partir da identificação de necessidades reais da indústria. É importante destacar que as empresas buscam parcerias de pesquisa com a academia diante da necessidade de soluções adaptadas as suas necessidades. Por sua vez, os pesquisadores trabalham com o interesse de prover tal solução, porém buscam também produzir uma solução genérica que atenda às necessidades de um público mais amplo. Diante deste enfoque na produção de algo específico para um contexto, ao mesmo tempo de algo genérico por partes dos pesquisadores, é fundamento ter um bom canal de comunicação e alinhamento das expectativas criadas por ambas as partes (academia e indústria) (WOHLIN, 2012).

REFERÊNCIAS

- ANASTAS, J. W.; MACDONALD, M. L. Research Design for the Social Work and the Human Services, 2nd Ed. *Columbia University Press*, New York, 2000.
- ANDERSSON, C., RUNESON, P. A Spiral Process Model for Case Studies on Software Quality Monitoring – Method and Metrics, *Software Process: Improvement and Practice*, Vol. 12, Issue 2, pages 125–140, 2007.
- BABBIE, E. R. Survey Research Methods, *Wadsworth*, Belmont, 1990.
- BASILI, V. R. The Experimental Paradigm in Software Engineering, In: Experimental Software Engineering Issues: Critical Assessment and Future Directives, *Lecture Notes in Computer Science*, Vol. 706, Springer, Berlin Heidelberg, 1993.
- EASTERBROOK, S.; SINGER, J.; STOREY, M.; DAMIAN, D. Selecting Empirical Methods for Software Engineering Research. In: *Guide to Advanced Empirical Software Engineering*, Springer, London, 2008.
- FARIAS, K.; GARCIA, A.; WHITTLE, J. Assessing the Impact of Aspects on Model Composition Effort, In: 9th *International Conference on Aspect-Oriented Software Development* (AOSD'10), pages 73-84, Rennes and Saint-Malo, France, 2010.
- FARIAS, K. Analyzing the Effort on Composing Design Models in Industrial Case Studies, In: 10th *International Conference on Aspect-Oriented Software Development Companion* (AOSD'11), pages 79-80, Porto de Galinhas, Brazil, 2011.
- FARIAS, K.; GARCIA, A.; WHITTLE, J. Evaluating the Impact of Aspects on Inconsistency Detection Effort: A Controlled Experiment, In: 15th *International Conference on Model-Driven Engineering Languages and Systems* (MODELS'12), Vol. 7590, pages 219-234, Innsbruck, Austria, 2012.
- FARIAS, K.; GARCIA, A.; WHITTLE, J.; LUCENA, C. P. Analyzing the Effort of Composing Design Models of Large-Scale Software in Industrial Case Studies, 16th *International Conference Model-Driven Engineering Languages and Systems*, pages 639-655, Miami, FL, USA, 2013.
- FARIAS, K.; GARCIA, A.; WHITTLE, J.; CHAVEZ, C.; LUCENA, C. Evaluating the Effort of Composing Design Models: A Controlled Experiment, *Journal on Software and Systems Modeling*, pages 1-17, 2015.
- FINK, A. The Survey Handbook, 2nd Edn., *SAGE*, Thousand Oaks/London, 2003.
- GLASS, R. L. The Software Research Crisis, *IEEE Software*, Vol. 11, pages 42–47, 1994.
- GORSCHEK, T.; GARRE, P.; LARSSON, S.; WOHLIN, C. A Model for Technology Transfer in Practice, *IEEE Software*, Vol. 23, No. 6, pages 88–95, 2006.
- GUIMARÃES, E.; GARCIA, A.; FARIAS, K. Analyzing the Effects of Aspect Properties on Model Composition Effort: A Replicated Study, In: 6th *Workshop on Aspect-Oriented Modeling* at MODELS'10, Oslo, Norway, 2010.

GONÇALES, L.; FARIAS, K.; SCHOLL, M.; VERONEZ, M.; OLIVEIRA, T. C.; Comparison of Design Models: A Systematic Mapping Study, *International Journal of Software Engineering and Knowledge Engineering*, Vol. 25, 9-10, pages 1765-1770, 2015.

GREEN, T. R. G.; PETRE, M. Usability Analysis of Visual Programming Environments: a 'Cognitive Dimensions' Framework, *Journal of Visual Languages and Computing*, vol. 7, pages 131–174, 1996.

IEEE, IEEE Standard Glossary of Software Engineering Terminology, *Technical Report*, IEEE Std 610.12-1990, 1990.

JURISTO, N.; VEGAS, S. The Role of Non-exact Replications in Software Engineering Experiments, *Empirical Software Engineering*, Vol. 16, pages 295–324, 2011.

KARLSTRÖM, D.; RUNESON, P.; WOHLIN, C. Aggregating Viewpoints for Strategic Software Process Improvement, *IEE Process Software*, Vol. 149, No. 5, pages 143–152, 2002.

KITCHENHAM, B. A. The Role of Replications in Empirical Software Engineering – A Word of Warning, *Empirical Software Engineering*, Vol. 13, pages 219–221, 2008.

KITCHENHAM, B. A.; PICKARD, L. M.; PFLEEGER, S. L. Case Studies for Method and Tool Evaluation, *IEEE Software*, 12(4): 52–62, 1995.

KITCHENHAM, B. A.; DYBA, T.; JØRGENSEN, M. Evidence-Based Software Engineering. In: *26th International Conference on Software Engineering*, Edinburgh, pages 273–281, 2004.

MANLY, B. Multivariate Statistical Methods: A Primer, 2nd Ed., *Chapman and Hall*, London, 1994.

PETERSEN, K.; WOHLIN, C. Context in Industrial Software Engineering Research, In: *3rd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Lake Buena Vista, pages 401–404, 2009.

PFLEEGER, S.L.: Experimental Design and Analysis in Software Engineering part 1–5, *ACM Sigsoft Software Engineering Notes*, Vol. 19, No. 4, pages 16–20, 1994.

PFLEEGER, S. L.; ATLEE, J.M.: Software Engineering: Theory and Practice, 4th Ed., *Pearson Prentice-Hall*, Upper Saddle River, 2009.

RAINER, A. W. The Longitudinal, Chronological Case Study Research Strategy: a Definition, and an Example From IBM Hursley Park, *Information Software Technology*, Vol. 53, No. 7, pages 730–746, 2011.

ROBSON, C. Real World Research: A Resource for Social Scientists and Practitioners-Researchers, 2nd Edn., *Blackwell*, Oxford/Madden, 2002.

RUNESON, P.; HÖST, M. Guidelines for Conducting and Reporting Case Study Research in Software Engineering, *Empirical Software Engineering*, Vol. 14, No. 2, pages 131–164, 2009.

RUNESON, P.; HOST, M.; RAINER, A. W.; REGNELL, B. Case Study Research in Software Engineering, *Guidelines and Examples*, Wiley, Hoboken, 2012.

SHULL, F.; MENDONCA, M. G.; BASILI, V.; CARVER, J.; MALDONADO, J.; FABBRI, S.; TRAVASSOS, G.; FERREIRA, M. C. Knowledge-Sharing Issues in Experimental Software Engineering, *Empirical Software Engineering*, Vol. 9, pages 111–137, 2004.

SHULL, F.; CARVER, J.; VEGAS, S.; JURISTO, N. The Role of Replications in Empirical Software Engineering, *Empirical Software Engineering*, Vol. 13, pages 211–218, 2008.

SJØBERG, D. I. K.; DYBA, T.; ANDA, B.; HANNAY, J. E. Building Theories in Software Engineering. In: *Guide to Advanced Empirical Software Engineering*, Springer, London, 2008.

SOMMERVILLE, I. *Engenharia de Software*. 8 ed. São Paulo: Pearson Addison-Wesley, 2007.

STAKE, R. E. The Art of Case Study Research, *SAGE Publications*, Thousand Oaks, 1995.

VERNER, J. M.; SAMPSON, J.; TOSIC, V.; ABU BAKAR, N. A.; KITCHENHAM, B. A. Guidelines for Industrially-Based Multiple Case Studies in Software Engineering, In: *3rd International Conference on Research Challenges in Information Science*, pages 313–324, 2009.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M.; REGNELL, B.; WESSLÉN, A. Experimentation in Software Engineering, *Springer Science & Business Media*, 2012.

YIN, R. K. Case Study Research Design and Methods, 4th Edn, *Sage Publications*, Beverly Hills, 2009.

ZELKOWITZ, M. V.; WALLACE, D. R. Experimental Models for Validating Technology, *IEEE Computing*, Vol. 31, No. 5, pages 23–31, 1998.