

A IMPORTÂNCIA DA EXPERIMENTAÇÃO EMPÍRICA NA PESQUISA EM CIÊNCIA DA COMPUTAÇÃO

THE IMPORTANCE OF EMPIRICAL EXPERIMENTATION IN RESEARCH ON COMPUTER SCIENCE

ANDERSON DA SILVA **MARCOLINO**¹, HELIO HENRIQUE LOPES COSTA **MONTE-ALTO**², LUCAS DE OLIVEIRA **TEIXEIRA**³, JEFERSON APARECIDO SILVA **BELGAMAZZI**⁴, JOSÉ CARLOS SANCHES AMENDOA **TISSEI**⁵, CLAUDIA HEIDEMANN DE **SANTANA**⁶

1. Doutorando em Ciências da Computação e Matemática Computacional do Instituto de Ciências Matemáticas e de Computação na Universidade Estadual de São Paulo; 2. Mestre em Ciência da Computação pela Universidade Estadual de Maringá, docente no curso de Ciência da Computação da Universidade Tecnológica Federal do Paraná; 3. Mestre em Ciência da Computação pela Universidade Estadual de Maringá, docente no curso de Ciências da Computação da Faculdade Ingá; 4. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 5. Aluno do curso de graduação em Ciências da Computação da Faculdade Ingá; 6. Mestre em Engenharia de Produção pela Universidade Estadual de Santa Catarina, docente no curso de Ciências da Computação da Faculdade Ingá.

Rua Lápiz Lazuli, 211, Jardim Santa Helena, Maringá, Paraná, Brasil. CEP: 87083-310. chsantana@gmail.com

Recebido em 05/10/2013. Aceito para publicação em 08/11/2013

RESUMO

A experimentação é um processo de validação de técnicas, abordagens, ferramentas e teses. Com isso, é importante em todas as áreas de conhecimento. Especificamente, na ciência da computação e na engenharia de software em que os estudos atuais quase sempre propõem abordagens, processos e produtos novos, o processo de experimentação deve ser parte fundamental do processo de criação de modelos. No entanto, nem sempre isso acontece na prática. Existem diversos motivos para isso, como: a intangibilidade do produto, muitas variáveis a ser controladas, custo elevado da experimentação, entre outros. Diante disso, o objetivo deste trabalho é apresentar os princípios benéficos provenientes da experimentação da ciência da computação. Além disso, mostrar quais as principais dificuldades e motivos da não realização de experimentos, bem como um conjunto de ferramentas especializadas de suporte ao processo de experimentação.

PALAVRAS-CHAVE: Engenharia de software, Engenharia de software experimental, Pesquisa em ciência da computação, Experimentação.

ABSTRACT

Experimentation is a process of validation of techniques, approaches, tools and theses. Thus, it is important in all areas of knowledge. Specifically in computer science and software engineering which the current studies almost always propose new approaches, processes and products, the process of experimentation must be a fundamental part of the model creation

process. However, this does not always happen in practice. There are many reasons for that, such as: the intangibility of the product, many variables to be controlled, high cost of experimentation, among others. Therefore, the aim of this paper is to present the main benefits from experimentation in computer science. Also, show the main difficulties and reasons for not carrying out experiments, as well as a set specialized tools for supporting the process of experimentation.

KEYWORDS: Software engineering, Experimental software engineering, Research on computer science, Experimentation.

1. INTRODUÇÃO

A ciência da computação, muitas vezes considerada uma ciência ainda imatura perante as mais tradicionais¹, possui fundamental importância tanto na vida acadêmica, quanto corporativa, incluindo também as demais áreas de tecnologia e afins. Os resultados obtidos na área trazem facilidades que vão de encontro com as tarefas do dia-a-dia e ainda, trazem lucros e aumento de produtividade.

Quando traçamos um paralelo desta ciência com as demais, em termos acadêmicos, nos deparamos com problemas, principalmente relacionados aos estudos promovidos pelos cientistas e pesquisadores desta área: faltam resultados comprovadamente testados e experimentados. E assim, acumulasse estudos incompletos e muitas vezes duvidosos, devido aos critérios de avaliação utilizados.

As técnicas propostas pela ciência na área de expe-

rimentação não são práticas comuns, para a grande maioria dos estudiosos na área tecnológica¹, porém estas são fundamentais para comprovar as pesquisas realizadas, mesmo se tratando de uma área em que os experimentos e variáveis, em sua grande maioria, são intangíveis².

Existem ainda ferramentas de software que permitem reproduzir simulações com uma grande carga de iterações, com as propostas de pesquisa, gerando resultados confiáveis e que muitas vezes demandaria uma vasta equipe e demasiado tempo. Tais ferramentas, como os *benchmarks*, propiciam a comprovação de hipóteses e assim, permitem refutar ou aceitar estudos. No entanto, ao focarmos a área de engenharia de software, temos um fator que, muitas vezes não pode ser medido por *benchmarks*, o fator humano.

O objetivo deste trabalho é realizar uma revisão de literatura sobre a importância da experimentação empírica na ciência da computação. Além disso, discutir os possíveis motivos para a não realização de experimentos formais em computação, os principais benefícios e ferramentas de suporte ao processo de experimentação.

2. MATERIAL E MÉTODOS

O presente estudo foi realizado com base na consulta de bibliografia específica da área e da temática em questão. Foram consultados periódicos nacionais e internacionais disponíveis no portal EBSCO host da Faculdade Ingá. Foram selecionados 22 artigos entre os anos de 1964 e 2012, pela relevância de informações em face do objetivo do presente levantamento.

3. RESULTADOS E DISCUSSÃO

A experimentação é o centro de todo processo científico, ou seja, novos métodos, técnicas, linguagens e ferramentas não deveriam ser apenas sugeridos, publicados ou apresentados sem uma experimentação e validação minuciosa³. Porém, especificamente, no caso de engenharia de software, em que geralmente os pesquisadores buscam o desenvolvimento de novas técnicas, métodos ou ferramentas, a experimentação é pouco utilizada.

O experimento é uma etapa de pesquisa científica no qual o pesquisador manipula e controla uma ou mais variáveis independentes e observa a variação nas variáveis dependentes concomitantemente à manipulação das variáveis independentes⁴.

O objetivo de controlar e manipular as variáveis é capturar a relação entre as causas e seus respectivos efeitos, assim um experimento é um método que investiga relações causais entre as variáveis para comprovar ou refutar pressupostos teóricos (Figura 1).

Um pretexto bastante utilizado pelos pesquisadores para evitar a experimentação é também uma das principais diferenças entre a ciência da computação e as ciên-

cias tradicionais: o fato de que informação não é nem energia nem matéria. No entanto, isso não faz diferença na aplicabilidade do método científico tradicional. Para entender a natureza da informação, deve-se observar fenômenos, formular explicações e teorias, e testá-las.

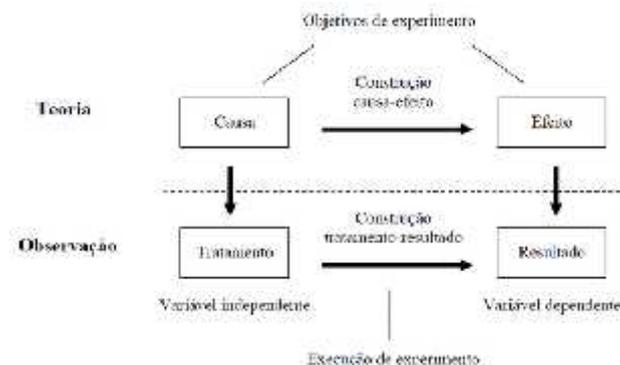


Figura 1. Conceitos de um Experimento⁵

Além disso, experimentação ajuda a observar novos fenômenos que podem ser usados para induzir novas teorias. Sintetizando os principais benefícios da experimentação, podemos citar⁶:

- A experimentação pode ajudar a construir uma base de conhecimento confiável e assim reduzir a incerteza sobre quais teorias, métodos e ferramentas são adequados;
- A observação e a experimentação pode conduzir a novos conhecimentos úteis e não esperados e, dessa forma, abrir novas áreas de investigação;
- A experimentação pode acelerar o processo de eliminação de abordagens infrutíferas, orientando a engenharia e a teoria às direções certas.

Quando focamos o processo de experimentação em engenharia de software, nos deparamos com diversas barreiras e algumas incógnitas geradas pela forma como esta é muitas vezes tratada, seja como engenharia ou como ciência. Desta forma, a experimentação para tais estudos requerem formalismos e técnicas das ciências humanas, quando nos referenciamos ao fator humano, diretamente presente e ativo em tais estudos.

Experimentação em engenharia de software

A engenharia de software, devido o caráter duplo do software, é muitas vezes considerada ciência e/ou engenharia. Este caráter duplo é identificado quando se considera, a princípio, o processo de criação do produto de software - sendo assim, apresenta características explícitas de produção ou engenharia. Quando são apresentados aspectos relacionados a *time-to-market* e competição, há a necessidade de melhoria contínua e sequencial da qualidade do processo e do produto - neste contexto, apresentasse características científicas⁷.

Deste modo, metodologias específicas são necessárias

para ajudar a estabelecer uma base de engenharia e de ciência para a engenharia de software. Existem quatro métodos relevantes para a condução de experimentos nesta área: científico, de engenharia, experimental e analítico⁸.

O método científico é um paradigma indutivo, que consiste na observação do mundo, sugestão de modelos ou teorias de comportamento, medições, análises e verificação de hipóteses do modelo ou da teoria. Este método pode ser utilizado quando é necessário entender o produto de software, o processo ou o ambiente. Assim, o objetivo é extrair do mundo algum modelo que possa explicar um fenômeno, e avaliar se o modelo é realmente representativo para tal fenômeno, que é foco da observação. Assim, se trata de uma abordagem para construção de modelos.

O método da engenharia mantém as observações nas soluções existentes, sugerindo as soluções mais adequadas, mede e analisa, e repete até que nenhuma melhoria adicional seja possível. Desta forma é uma abordagem orientada à melhoria evolutiva, assumindo sempre a existência de algum modelo do processo ou produto de software, modificando este com o propósito de melhorar os objetos de estudo.

O método analítico, ou também denominado matemático, sugere uma teoria formal, desenvolve a teoria, deriva os resultados e comparados tais resultados com as observações empíricas. Assim, é um método dedutivo que não necessita de um projeto experimental no sentido estatístico, entretanto, fornece uma base analítica para o desenvolvimento de modelos.

E por último, o método experimental desenvolve o método qualitativo e/ou quantitativo, aplica um experimento, mede e analisa, avalia o modelo e repete o processo. Com isso, pode ser caracterizado como uma abordagem orientada à melhoria evolucionária. O processo tem início com o levantamento de um modelo novo, sendo este não obrigatoriamente baseado no já existente, e tenta estudar o efeito do processo ou produto sugerido pelo novo modelo.

Estudos experimentais, atualmente, são considerados a melhor abordagem para o melhor entendimento e avaliação das tecnologias que estão sendo desenvolvidas⁹.

Mesmo tendenciados ao uso do método experimental, os demais podem ser necessários, como por exemplo: o método científico pode ser utilizado para compreender a maneira como o software está sendo construído por uma organização para a verificação de tal ferramenta, validando se esta pode ser utilizada para automatizar o processo; o método de engenharia ajuda a demonstrar que uma ferramenta possui um desempenho melhor que outra; e por último, o método analítico pode provar modelos matemáticos para conceitos, como o crescimento da confiabilidade, a complexidade do software, o projeto ou código propenso a erros, etc⁹.

Ainda em relação à pesquisa experimental, existem

vários métodos, podendo ser classificados em¹⁰:

- Pré-experimentais: São estudos exploratórios e não se tem quase nenhum controle sobre as variáveis, a seleção das unidades não é aleatória, portanto são experimentos simples e pouco confiáveis;

- Experimentais verdadeiros: Possuem grupos de controle além dos grupos experimentais e a seleção de unidades é totalmente aleatória;

- Quase experimentais: Estudos que realizam a medição ao longo de um determinado período de tempo, porém a seleção de unidades não é aleatória;

- Estatísticos: Utilizam controles estatísticos e análise de variáveis externas, são mais complexos, porém apresentam mais informações e permitem a avaliação de mais de uma variável ao mesmo tempo.

A escolha de qual método de pesquisa utilizar depende de vários fatores, como: os requisitos da investigação, disponibilidade das unidades e recursos, propósito da pesquisa e da análise dos dados obtidos⁸.

Entretanto, mesmo tendo conhecimento do processo de experimentação em si e até mesmo nos demais métodos, ainda há uma grande lacuna a ser preenchida quando observamos os trabalhos na área de ciência da computação¹¹. Existem evidências de que mais de 30% de artigos publicados em diversos periódicos relacionados à software não possuem qualquer validação experimental e apenas 10% dos artigos demonstram algum tipo de abordagem formal de experimentação¹¹.

Quando focamos em engenharia de software, uma pesquisa comandada por Tichy *et al.* (1995)¹² apresentou 400 artigos desta área, dos quais 40% não apresentaram nenhum tipo de experimentação, sendo um número alto, visto que em outras disciplinas o percentual é mantido em 15%.

Tais estudos nos levam à conclusão de que a comunidade de engenharia de software deve melhorar seus trabalhos, reportando resultados avaliados corretamente. Com isso, facilitando a adoção de tais resultados na indústria e em outros ramos². Deste modo, passamos a analisar as várias desculpas dos cientistas da computação e pesquisadores da área para a não realização de experimentos.

No estudo desenvolvido por Juristo & Moreno (2010)² são apresentados algumas dificuldades para o uso da experimentação na engenharia de software:

- Desenvolvedores de software não estão a par da importância e do significado do método científico, que atuam validando teorias e convertendo-as em fatos;

- Desenvolvedores de software não são capazes de entender facilmente como analisar os dados de um experimento ou como eles serão analisados por outros, devido à falta de treinamento;

- A ausência de livros sobre projeto e análise experimental na área de engenharia de software dificulta o processo de aprendizagem e utilização desta em estudos;

- Estudos empíricos, conduzidos para verificar propostas já existentes, apresentadas por outros pesquisadores, não são muito publicados;

- A grande quantidade de variáveis que influenciam no desenvolvimento de software;

- Dificuldades em obter resultados globais, pois é difícil determinar sobre quais circunstâncias uma técnica deve ser selecionada. Em outras palavras, é difícil provar que uma determinada alternativa é sempre a melhor alternativa;

- Outra restrição importante sobre a execução de experimentos em engenharia de software é o efeito do fator humano, pois esta é uma disciplina cujo resultado é dependente dos praticantes;

- Por último, um fator já apresentado anteriormente: o fato da grande quantidade de dinheiro circulando no mercado de software hoje, o que implica em uma gama de novas abordagens sendo injetadas no mercado sem uma real análise dos benefícios de seu uso.

Estas e outras dificuldades poderiam ser combatidas, se os clientes - usuários finais de tais objetos de estudo - solicitassem resultados que fossem validados experimentalmente.

No estudo apresentado por Feitelson (2006)¹³, este conclui, após apresentar diversos fatores, que as falhas no processo de realização de experimentação em ciência da computação é na verdade um problema cultural e não técnico. Ainda há muitas barreiras a serem derrubadas, para isso, basta analisarmos as muitas argumentações infundadas e utilizadas como desculpas para não avaliar as pesquisas.

Deste modo, é necessário explicitar o quão fundamental são os resultados validados com o método científico e quais os reais benefícios para a comunidade da ciência da computação, e principalmente, para a engenharia de software, objetivando assim atingir adeptos à prática da experimentação.

Argumentos utilizados para não realizar experimentação

É uma tendência na área de ciência da computação, principalmente por pesquisadores de engenharia de software, ignorar ou dar pouca importância à validação das ideias por meio de experimentação. Apesar de ser uma tendência geral, questiona-se o fato dos pesquisadores brasileiros não realizarem experimentos de software pela forma como o assunto foi abordado em sala de aula durante sua vida escolar. É também famosa a cultura do “*jeitinho brasileiro*”, em que geralmente prefere-se executar ao planejar, optando pelo caminho mais fácil.

É notável que em outros países exista mais foco na experimentação como parte fundamental do ensino. É comum em filmes estrangeiros, que apresentam o contexto colegial, professores e alunos realizarem experi-

mentos. Essa cultura está presente desde os níveis mais básicos da educação, tornando a experimentação presente na maioria dos trabalhos científicos.

Tichy (1998) [6] defende que o principal assunto de pesquisa em ciência da computação não é meramente o computador, mas sim estruturas e processos de informação. Desta forma, deve-se pensar sobre computadores e programas como modelos. De fato, modelagem é uma das principais tradições utilizadas na ciência, uma vez que ajuda a estudar fenômenos mais detalhadamente.

Muitos pesquisadores alegam que o nível de experimentação nos trabalhos em geral é bom o suficiente. Tal argumento mostra-se falho quando se observa a quantidade de trabalhos que requerem suporte empírico, mas não o fazem. Cerca de 40 a 50% dos artigos publicados na área de computação carecem desse tipo de validação [6].

Outro argumento usado pelos pesquisadores da área para eliminar a necessidade de experimentação é o custo adicional acarretado pela realização dos experimentos, uma vez que requerem muitos recursos. No entanto, ao compara com outras áreas, é possível perceber que tal custo é aceitável, pelo fato de que muitas vezes, os benefícios compensam o custo.

Alguns pesquisadores também alegam que demonstrações são suficientes, de modo que substituem a necessidade de experimentos. De fato, demonstrações podem prover provas de conceito ou incentivos para estudos mais aprofundados, de modo que apenas ilustram o potencial da abordagem proposta. Assim, demonstrações não produzem evidências sólidas, uma vez que dependem da capacidade do observador de vislumbrar potenciais. É necessário que haja uma análise cuidadosa envolvendo experimentos para se obter evidências sólidas.

Outra linha de defesa contra a experimentação é a alegação de que há muito ruído na coleta dos dados. Muitas vezes, por haver muitas variáveis para controlar, os pesquisadores imaginam que resultados não terão significância por causa do ruído. No entanto, tal problema pode ser minimizado pelo uso de *benchmarks*. Um *benchmark* é uma amostra de um conjunto de trabalho que deve ser executado pelo computador, com ou sem acompanhamento humano. Tais programas devem ser projetados de forma que se possa coletar os dados desejados com o mínimo de ruído e ainda facilitar a repetição de experimentos.

A construção de *benchmarks* costuma ser custosa e trabalhosa. No entanto, é possível que vários laboratórios cooperem entre si para sua elaboração, uma vez que determinados tipos de experimento podem ser de interesse de vários grupos de pesquisa. Além disso, uma vez definido, um *benchmark* pode ser executado repetidamente a um custo moderado. *Benchmarks* vêm sendo utilizados com sucesso em diversas áreas, incluindo recuperação de informações, reconhecimento de padrões, reuso de software, arquitetura de computadores, avaliação de de-

sempenho, etc.

De fato, em algumas subáreas da ciência da computação muitas vezes não é possível utilizar *benchmarks*, uma vez que envolvem a interação humana. No entanto, isto não é desculpa para não realizar experimentos. Áreas como medicina e psicologia, que lidam com o fator humano, encontraram várias técnicas para lidar com a variabilidade humana - como: grupos de controle, distribuição aleatória, placebos, pré e pós-teste, balanceamento, baterias de testes estatísticos, etc.

É comum também, por parte dos pesquisadores, pensar que experimentações irão retardar o progresso da pesquisa. No entanto, o aumento da proporção de artigos com boa validação pode acabar acelerando o progresso, uma vez que ideias questionáveis seriam descartadas mais rapidamente, fazendo com que haja mais concentração nas ideias mais promissoras.

Segundo Mudge (1996)¹⁴, o desenvolvimento de novas tecnologias ocorre tão rapidamente que quando os resultados são confirmados por meio de experimentos eles podem não ser mais relevantes. No entanto, se uma questão se torna irrelevante tão rapidamente, talvez seja pelo fato de tal ser efêmera e não valer a pena utilizar. Os pesquisadores deveriam focar nas abordagens fundamentais e promissoras. Além disso, deve-se antecipar tais mudanças e, proativamente, realizar experimentos que permitam explorar as consequências de tais mudanças⁶.

Outro problema que surge ao realizar experimentação na área de computação é que muitos editores e revisores de publicações têm dificuldade em avaliar trabalhos empíricos. No entanto, há muitos periódicos e conferências que os valorizam. É interessante também dividir o trabalho em pequenos passos que possam ser publicados aos poucos, pois melhoram nosso entendimento e levantam novas questões.

Desta forma, podemos sumarizar os principais argumentos utilizados pelos cientistas da computação e profissionais da área para a não realização de experimentos⁶:

- O método científico tradicional não é aplicável;
- O nível atual de experimentação é suficiente;
- Experimentos são caros;
- Demonstrações são suficientes;
- Experimentos são trabalhosos;
- Experimentação atrasa o progresso;
- Tecnologias mudam muito rápido;
- Dificuldade de publicação.

Benefícios da experimentação em engenharia de software

O processo de desenvolvimento de software atual se baseia em boas práticas, experiências pessoais e observações gerais, porém apenas esses fatores não podem assegurar a qualidade e conformidade do artefato resultante. Assim, o processo de desenvolvimento deve ser

seguido imediatamente de uma avaliação para verificar a conformidade com os requisitos iniciais com o objetivo de identificar pontos falhos¹⁵.

Com isso, estudos experimentais possuem como objetivo identificar se o artefato resultante do processo de desenvolvimento obtém os resultados esperados, além de identificar os problemas e dificuldades que as pessoas tiveram ao utilizá-lo na prática¹⁶.

Além de satisfazer seu objetivo principal, a experimentação possui os seguintes benefícios^{6, 17}:

- Construção de uma base confiável de conhecimento reduzindo a incerteza sobre quais teorias, métodos e ferramentas são adequados;
- Propicia novas compreensões sobre áreas que estão sendo pesquisadas;
- Exploração de áreas desconhecidas, estendendo a fronteira do conhecimento, possibilitando o descobrimento de novas áreas de pesquisa;
- Eliminação de abordagens mal fundamentadas, errôneas e modismos, orientando a teoria em direções mais promissoras;
- Demonstração da real utilidade e benefícios de novas técnicas, métodos e tecnologias;
- Finalmente, no processo de formação da engenharia de software como ciência, por meio do aumento de trabalhos científicos com experimentos significativos.

Atualmente, os argumentos para não realizar experimentação não são válidos, uma vez que existem muitos benefícios em sua realização. Notada a importância da avaliação de pesquisa, observa-se também a adesão de ferramentas de software e técnicas para a realização das mesmas, visando à minimização das não ocorrências destas ou a redução do número de falhas, bem como auxiliar e acelerar tal processo.

Ferramentas de software e técnicas para a avaliação de pesquisa

Dentre as ferramentas existentes para auxiliar etapas no processo de experimentação, com foco em diversas áreas, não somente na área da Engenharia de Software, temos:

- *GQM (Goal Question Metric)*: Ferramenta que visa facilitar a adoção de princípios sistemáticos e possui importância na obtenção de métricas para o experimento [19]. Baseia-se nos princípios da avaliação orientada a objetivos que, segundo Gladcheff, Sanches e Da Silva (2012)¹⁸, visa servir como uma metodologia genérica para orientar a elaboração e execução de programas de avaliação da qualidade de processos na área de engenharia de software.

- *SGWfC (Sistemas de Gerência de Workflows Científicos)*: Ferramenta que apóia a execução de *workflows* de modo controlado e documentado. Entretanto, tal sistema não oferece recurso para acompanhar as três etapas

do experimento científico como um todo (composição, execução e análise). Esta ferramenta não tem o foco na engenharia de software, visando dar suporte ao processo de experimentação científico em laboratórios, entretanto, pode ser usada na área tecnológica. Como exemplos de tais sistemas temos Taverna, Kepler e VisTrails¹⁹.

É possível citar ainda outras ferramentas de software que auxiliam no processo de tratamento dos resultados obtidos com a experimentação e na aplicação de avaliações estatísticas, tais como:

- Microsoft Excel e o Calc (BrOffice): Ambos programas para a geração de planilhas eletrônicas;

- Statistica, Minitab, MathLab: Programas que lidam com informações estatísticas, permitindo trabalhar com uma grande quantidade de dados e tratá-los com as mais diversas fórmulas de análise e comparação de dados numéricos;

- AVAs (Ambientes Virtuais de Aprendizagem): Ferramentas disponíveis na Internet, objetivando compartilhamento de conhecimento, podendo servir como modo de compartilhamento de resultados e explanação sobre os mais diversos assuntos. Servindo assim, para que equipes possam desenvolver seus trabalhos, mesmo estando geograficamente distribuídas. Como exemplos temos o Moodle, Trello, edX, etc.

Estas ferramentas auxiliam o processo, mas ainda necessitam de certo formalismo, para serem utilizadas, ou seja, a aplicação de técnicas que permitam um controle sistematizado do experimento, desde seu projeto até a apresentação concisa dos resultados obtidos. Assim, podemos mencionar algumas das técnicas existentes hoje, para a realização deste processo:

- Estudo de Caso: É uma técnica poderosa e flexível. Usada principalmente para investigações exploratórias, tanto prospectiva e retrospectiva, que tentam compreender e explicar fenômenos ou construir teorias. Caracteriza-se por ser, geralmente, observacional ou de natureza descritiva, embora possa ser relacional. É aplicada também na avaliação de resultados de investigação²⁰.

- Experimento formal: É mais rigoroso que estudos de caso, por considerar um estado variável - um fator que pode caracterizar a influência do projeto e da avaliação de resultados. Experimentos formais podem ser usados para prover o contexto no qual certos padrões, métodos e ferramentas são recomendadas para uso²¹.

- *Surveys*: Experimentos executados em retrospecto, quando uma técnica ou método já está em uso por alguma empresa e deseja avaliá-la em algum aspecto formal³. As maneiras mais conhecidas e bastante utilizadas são questionários e entrevistas.

- *Quasi-experiments*: Experiências em que as unidades de estudo são aplicadas a grupos experimentais não randomizados. Eles permitem que as investigações de relações causa-efeito em ambientes em que a randomização é inadequada, impraticável, ou muito cara²².

Finalmente, temos que estudos empíricos são compostos por experimentos formais, estudos de caso e *surveys*²¹. Experimentos empíricos são aqueles comprovados pela prática, ou seja, é o estudo de um objeto através de tentativa ou erro, sem nenhum parâmetro estabelecido, apenas utilizando a observação.

4. CONCLUSÃO.

Enfim, é possível perceber que a experimentação é central no processo científico, pois apenas experimentos testam teorias, permitem explorar fatores críticos e trazem à tona novas questões, de modo que teorias possam ser formuladas e corrigidas. De acordo com Tichy (1998)⁶, a ciência da computação está em perigo se os pesquisadores não usarem experimentação para validar suas teorias.

Algumas teorias da ciência da computação, como programação funcional, orientada a objetos e métodos formais, nunca foram avaliadas para se verificar sua real veracidade. Muitos destas acabam sendo, de certa forma, experimentadas diretamente pelo usuário final, como em empresas e também no meio acadêmico. Isto pode agregar muitas vezes gastos e riscos desnecessários, pois o uso de técnicas e métodos não comprovados podem mais prejudicar do que colaborar com o desenvolvimento de um projeto.

Ainda falta visão quanto ao impacto de um estudo carente de avaliação, principalmente quando falamos de tecnologia. Com o advento da Internet, em poucos minutos, novos conhecimentos são difundidos de um lado do globo a outro, e muitas empresas se baseiam no conhecimento divulgados sem se preocupar em identificar a veracidade dos resultados e estudos disponibilizados. Assim, acabam muitas vezes investindo em algo que, muitas vezes, não gerará o retorno esperado.

REFERÊNCIAS

- [1] Denning PJ. Is computer science science? *Communications of the ACM*. 2005; 48(4):27-31.
- [2] Juristo N, Moreno AM. *Basics of software engineering experimentation*. Springer Publishing Company, Incorporated. 2010.
- [3] Amaral EAGG. *Empacotamento de experimentos em engenharia de software*. Dissertação (Mestrado), Programa de Engenharia de Sistemas e Computação - COPPE/UFRJ, Rio de Janeiro. 2003.
- [4] Kerlinger FN, Lee HB. *Foundations of behavioral research: Educational and psychological inquiry*. New York: Holt, Rinehart and Winston. 1964.
- [5] Basili VR, Selby RW, Hutchens DH. *Experimentation in software engineering*. *Software Engineering, IEEE Transactions on*. 1986; 7:733-43.
- [6] Tichy WF. Should computer scientists experiment more? *Computer*. 1998; 31(5):32-40.
- [7] Travassos GH, Gurov D, Amaral EAGG. *Introdução à en-*

- genharia de software experimental. UFRJ. 2002.
- [8] Wohlin C, et al. Experimentation in software engineering. Springer. 2012.
- [9] Travassos GM, Shull F, Fredericks M, Basili VR. Detecting defects in object-oriented designs: using reading techniques to increase software quality. *ACM SigplanNotices*. 1999; 34(10).
- [10] Malhotra NK. Pesquisa de marketing: uma orientação aplicada. Bookman. 2012.
- [11] Zelkowitz MV, Wallace DR. Experimental models for validating technology. *Computer*. 1998; 31(5):23-31.
- [12] Tichy WF, Lukowicz P, Prechelt L, Heinz EA. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*. 1995; 28(1):9-18.
- [13] Feitelson DG. Experimental computer science: The need for a cultural change. 2006.
- [14] Mudge T. Report on the panel: how can computer architecture researchers avoid becoming the society for irreproducible results. *Computer Architecture News*. 1996; 24(1):1-5.
- [15] Babar MA, Gorton I. Comparison of scenario-based software architecture evaluation methods. *Software Engineering Conference, 2004. 11th Asia-Pacific. IEEE*. 2004.
- [16] Ciolkowski M, Shull F, Biffl S. A family of experiments to investigate the influence of context on the effect of inspection techniques. NA. 2002.
- [17] Travassos GH, Shull F, Carver J. Working with uml: A software design process based on inspections for the unified modeling language. *Advances in Computers*. 2002; 54:35-98.
- [18] Gladcheff AP, Sanches R, Silva DM. Um instrumento de avaliação de qualidade de software educacional: Como elaborá-lo. *Pensamento & Realidade. Revista do Programa de Estudos Pós-Graduados em Administração-FEA*. 2002; 11.
- [19] Mattoso M, *et al.* Desafios no apoio à composição de experimentos científicos em larga escala. *Seminário Integrado de Software e Hardware, SEMISH*. 2009; 9:36.
- [20] Perry DE, Sim SE, Easterbrook SM. Case studies for software engineers. *Software Engineering. ICSE 2004. Proceedings. 26th International Conference on. IEEE*. 2004.
- [21] Fenton NE, Pfleeger SL. *Software metrics: a rigorous and practical approach*. PWS Publishing Co. 1998.
- [22] Kampenes VB, Dyba T, Hannay TE, Sjoberg DIK. A systematic review of quasi-experiments in software engineering. *Information and Software Technology*. 2009; 51(1):71-82.

