PROVENIÊNCIA DE DADOS EM EXPERIMENTOS DE SOFTWARE: UMA PROPOSTA DE MELHORIA SOB A ÓTICA DA QUALIDADE DE DADOS

DATA PROVENIENCE IN SOFTWARE EXPERIMENTS: A PROPOSAL OF IMPROVEMENT FROM THE PERSPECTIVE OF DATA QUALITY

Claudia de O. Melo[1], Sidney Viana[2], Judith Pavón[2], Jorge R. Almeida Jr[3].

[1]Universidade de São Paulo, Departamento de Ciência da Computação.
claudia.melo.prof@gmail.com
[2]Universidade Salvador, NUPERC - Núcleo de Pesquisa em Redes e Computação
sidney.viana@gmail.com, judithpm@gmail.com
[3]Universidade de São Paulo, Departamento de Engenharia de Computação e Sistemas Digitais.
jorge.almeida@poli.usp.br

Resumo

Experimentação em Engenharia de Software é fundamental para a evolução dessa ciência. Um dos maiores problemas enfrentados pela comunidade de Engenharia de software experimental é a ameaça da validade dos resultados dos experimentos. Um dos motivos é a falta de qualidade dos dados. Diversos estudos abordam conceitos e métricas de qualidade de dados, mas poucos exploram a proveniência de dados como elemento de um modelo de qualidade de dados para experimentos de software. O objetivo deste trabalho é apresentar um modelo de qualidade de dados baseado em dimensões voltadas para experimentos de software, com aplicação de proveniência de dados para o cálculo das métricas.

Palavras-chave: Engenharia de Software Experimental; Proveniência de dados; Qualidade de dados; Dimensões de qualidade; Modelo de qualidade.

Abstract

Experimentation in Software Engineering is fundamental to the science evolution. One of the largest issues faced by the experimental software engineering community is the threat to validity of the experimental results. One reason is the lack of data quality. Several studies have been published on data quality concepts and metrics, but few studies explore data provenance as a key part of data quality model for software experiments. This study aims to present a quality data model based on software experiments dimensions using data provenance to calculate metrics.

Keywords: Experimental Software Engineering; Data Provinance; Data Quality; Quality Dimensions; Quality Model.

1 INTRODUÇÃO

Ciências que estudam fenômenos do mundo real, isto é, ciências empíricas, necessitam de métodos que consistem mais da obtenção de informações a partir de observação e experimentação, do que da lógica dedutiva ou matemática. Portanto, se a pesquisa em Engenharia de software deve ser científica, então é necessário usar métodos empíricos. Abordagens empíricas para avaliar tecnologias de Engenharia de software, inclusive com colaboração da indústria, iniciaram em larga escala desde os anos de 1970 [15]. Essa preocupação refletiu-se na criação de diversos eventos científicos na área Engenharia de software experimental (ESE), como *Journal of Empirical SE* (EMSE) - desde 1996, *IEEE International Symposium on Software Metrics* (METRICS) - desde 1993, *Empirical Assessment & Evaluation in SE* – EASE - desde 1997, e *IEEE International Symposium on Empirical SE* (ISESE) - desde 2002.

Segundo Sjøberg et al. [15], para evoluir a área de Engenharia de software são necessários mais estudos empíricos de qualidade, relevância e maior foco em sintetizar evidências e construir teorias. Precisa-se, portanto, de estudos nos quais podemos confiar, principalmente em relação a sua validade. Aplicações científicas (como as usadas em experimentos de software) se caracterizam por possuírem grandes quantidades de dados complexos [17]. Price e Shanks [16] mostraram que informações sobre a qualidade dos dados influenciam não apenas a de tomada de decisão sobre tais dados, mas também o processo de tomada de decisão. Por isso, propuseram um processo de anotação da qualidade de dados para melhorar o planejamento e a validade de estudos experimentais. Nesse contexto, dentre as diversas preocupações para melhorar a validade de experimentos em engenharia de software, a qualidade de dados representa um tópico importante de pesquisa.

Segundo Mattoso et al. [17], fazer ciência hoje implica, dentre outros aspectos, ubiquidade e distribuição, visando ao desenvolvimento e execução de soluções com alto desempenho, baseadas, por exemplo, em reutilização, gerência de dados e experimentos.

Esse tipo de ciência hoje é conhecido como e-ciência (ou *e-science*). Na e-ciência é vital gravar o processo experimental para uso futuro, portanto, é necessário interpretar os resultados, verificar a corretude do processo e rastrear a origem dos dados [21]. Surge então o conceito de proveniência de dados, que auxilia o rastreamento da origem dos dados e, por consequência, auxilia o gerenciamento dos dados em experimentos. Quanto à gerência de dados de proveniência, Marinho [28] propõe uma solução que facilita a captura, o armazenamento e a análise integrada de informações de proveniência em cenários de ambientes heterogêneos e distribuídos. No entanto, poucos trabalhos relacionam os conceitos de qualidade de dados, proveniência de dados e sua aplicação na melhoria de experimentos [31], especificamente em qualidade dos dados e engenharia de software. Este artigo tem como objetivo apresentar um modelo de qualidade de dados em experimentos de software utilizando proveniência de dados.

O artigo está dividido da seguinte forma: na Seção II são apresentados os conceitos da Engenharia de software experimental e seus principais problemas e desafios. Na Seção III são apresentados os conceitos, aplicações e benefícios da proveniência de dados para o uso geral. Na Seção IV é apresentado o modelo de qualidade de dados para experimentos de software aplicando a proveniência de dados. Na Seção V são apresentadas as discussões, conclusões e trabalhos futuros.

2 ENGENHARIA DE SOFTWARE EXPERIMENTAL

Experimentação em Engenharia de software é necessária [4]. Um experimento controlado é uma forma de estudo experimental na qual o investigador tem controle sobre os principais aspectos e variáveis em estudo. Ele pode também ser visto como uma operação realizada sob circunstâncias controladas a fim colocar à prova uma hipótese em observação [9].

A Engenharia de Software Experimental (ESE) é uma subárea da Engenharia de Software que visa investigar teorias, métodos e técnicas pela experimentação. Esta experimentação deve envolver investigadores, ambientes, domínios de forma

colaborativa [5]. O processo de experimentação segue, em geral, as atividades [9, 24, 12] ilustradas pela Figura 1.

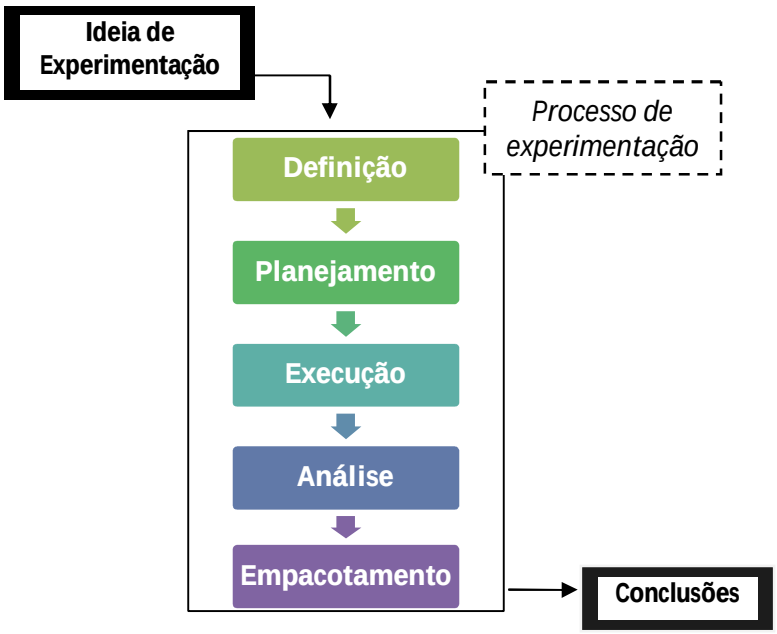

Figura 1 - Processo de experimentação de software (Adaptado de Hholinoo, 2000)

Na **Definição** ocorre a identificação dos objetivos do estudo. A abordagem GQM (*Goal, Question, Metric*) [30] é frequentemente usada no estabelecimento de objetivos, formulação de questões e métricas para processos de experimentação. No **Planejamento** é feito o projeto do estudo experimental, produção, coleta e preparação de todo o material necessário para a condução do experimento. Dentro deste escopo estão a formulação de hipóteses, identificação de variáveis dependentes e independentes, seleção de métodos de análise e instrumentos, além da análise de riscos.

A **Execução** do experimento é realizada de acordo com o planejamento e coleta de dados. Nesta fase diversas ferramentas podem ser utilizadas para dar suporte à automatização de parte ou todo o processo. Na **Análise** os resultados são analisados de acordo com os métodos selecionados no planejamento, possivelmente a aplicação de técnicas estatísticas. Comparar tais dados com outros obtidos a partir de mineração dos repositórios de software tem sido uma abordagem utilizada para validar empiricamente

novas ideias e técnicas e entender a evolução do software [19, 20]. Dados qualitativos (por exemplo, obtidos por meio de entrevistas e questionários) são frequentemente usados tanto para encontrar as causas ou justificar os resultados quantitativos, quanto para estabelecer novas hipóteses que possam ser medidas quantitativamente.

Um dos objetivos da ESE, é construir uma base de conhecimento empírico consolidada que identifique os benefícios e custos de várias técnicas e ferramentas para suporte à Engenharia de Software [6]. Para envolver diversos investigadores, ambientes e domínios são necessárias estratégias de replicação de experimentos. A replicação de experimentos tem sido abordada pela comunidade científica de ESE em diversos trabalhos, alguns muito recentes [12, 7, 8, 13, 14]. No entanto, um estudo realizado por Hannay et al. [8] mostrou que em 113 experimentos pesquisados, apenas 18% deles realizaram replicação. Para Miller [7], parte do problema está em tornar as replicações externas (realizadas por outros grupos) mais atrativas. Para criar evidências é necessário repetir experimentos diversas vezes. Pacotes de laboratório são uma importante ferramenta para a replicação de experimentos [6,12]. A repetição de experimentos permitem aos pesquisadores responder perguntas que estão além do escopo dos experimentos individuais [4]. O **Empacotamento** descreve o estudo, seus artefatos e resultados de modo que a comunidade externa seja capaz de compreendê-los, bem como replicá-lo em contextos diferentes.

A. Métodos de Coleta de dados em Experimentos de Software

Os dados coletados durante os experimentos de software podem ter diversas fontes. De acordo com Lethbridge et al. [22], as técnicas de coleta de dados podem ser classificadas de acordo com o grau de contato humano que ela requer. A Tabela I apresenta os diferentes graus de contato e seus tipos de técnica de obtenção de dados.

Tabela 1 - Principais técnicas de coleta de dados em pesquisa em engenharia de software (adaptado de [22])

Categoria	Técnica
Primeiro grau (envolvimento direto com engenheiros de software)	*Técnicas inquisitivas :* • Brainstorming; • Entrevistas; • Questionários; • Modelagem Conceitual. *Técnicas observacionais:* • Diários de trabalho; • Participação observadora.
Segundo grau (envolvimento indireto com os engenheiros de software)	Sistemas de instrumentação; Participantes gravando seu trabalho.
Terceiro grau (estudo apenas dos artefatos de trabalho)	Análise das bases eletrônicas de trabalho realizado; Análise de logs de ferramentas; Análise de documentação; Análise estática e dinâmica dos sistemas.

3 PROVENIÊNCIA DE DADOS

Proveniência (também conhecida como linhagem ou *pedigree*) de dados é definida como a descrição da origem de um pedaço (*stream*) de dado e o processo pelo qual este chegou ao banco de dados [10]. A origem de um dado é o dado fonte que foi utilizado como entrada para geração de um determinado produto a partir de um processo. A captura da proveniência pode se dar tanto em nível de dado quanto em nível de processo.

Segundo Freire et al. [1], a proveniência pode ser capturada por meio de dois caminhos, prospectiva e retrospectiva. A forma prospectiva captura os passos que devem ser seguidos para a geração de um dado produto (por exemplo, os passos que devem ser seguidos para a execução de um conjunto de processos relacionados ao *build* de uma aplicação), permitindo desta forma o registro da especificação de tarefas computacionais (por exemplo, conjunto de processos, um script). A forma de captura de proveniência

retrospectiva captura os passos executados por uma tarefa computacional assim como a informação sobre o ambiente utilizado para derivar um dado produto específico, ou seja, um log detalhado sobre a execução da tarefa (por exemplo, os dados produzidos por um sistema computacional para busca por similaridade entre sequências de DNA, além de todos os parâmetros envolvidos na execução desse sistema). Essas duas formas de captura de proveniência são independentes, ou seja, para a captura da proveniência prospectiva, a captura da proveniência retrospectiva não é estritamente necessária. O modelo conceitual de proveniência define como as informações de proveniência são representadas ou modeladas. Um modelo de proveniência pode representar informações de proveniência tanto prospectiva quanto retrospectiva. Além disso, esses modelos podem incluir anotações para prover mais semântica aos dados de proveniência [29].

A forma de captura da proveniência pode ser em maior ou menor nível de detalhe e pode ser classificada em dois níveis: "grão grosso" e "grão fino" [10]. A granularidade de proveniência chamada "grão grosso" está relacionada à proveniência de um conjunto de atividades, chamado *workflow*. Ela descreve a história da execução de um *workflow* e/ou da derivação de um conjunto de dados. A Figura 2 mostra um exemplo de *workflow* científico onde os dados e seu processamento podem ser gravados e rastreados por meio da proveniência. Elipses representam os componentes do workflow, quadrados denotam dados e o cilindro representa fontes de dados.

Já a proveniência tendo granularidade "grão fino" está relacionada à proveniência de dados, pois descreve a história da proveniência de um item de dado de um conjunto de dados, de forma a capturar a origem e a movimentação de um dado (ou seja, "onde" foi originado) que pode estar relacionado a bancos de dados integrados e descrever a importância da presença de um item de dado na composição de uma informação.

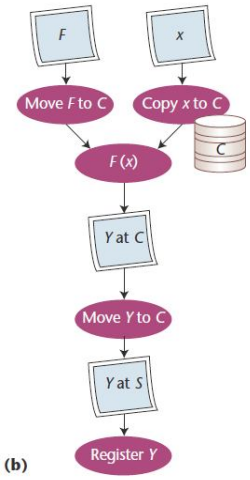

Figura 2 - Exemplo de workflow científico [31].

De acordo com Simmhan [17], a proveniência de processos envolve a descrição da execução de um simples processo, ou seja, das tarefas que dele fazem parte. Em determinados processos, como os *workflows* científicos, para que se tenha a informação dos dados geradores de um dado produto, é importante que se registre cada dado consumido pelo processo. Assim os dados não se perdem quando for necessário investigar a proveniência de dados. Já para que seja capturada a proveniência de um processo, é necessário capturar a descrição da execução de um processo, ou seja, todas as tarefas que foram executadas para que se tenha a informação de todas as tarefas que obtiveram sucesso ou não durante suas execuções e assim, permitir que sejam constatadas todas as tarefas que influenciaram no resultado obtido na execução de um processo.

A. Importância e aplicação da proveniência de dados

A proveniência pode ser uma métrica de qualidade importante já que processo de derivação dos dados tem implicações significativas na qualidade de dados e nos erros introduzidos por dados falhos que aumentam na medida em que se propagam em outras derivações [2].

Em experimentos científicos, a proveniência nos ajuda a interpretar e entender resultados: examinando a sequência de passos que levaram ao resultado, podemos ter ideias sobre a cadeia de raciocínio utilizada na sua produção, verificar que o experimento

foi realizado de acordo com procedimentos aceitáveis, identificar as entradas do experimento e, em alguns casos, reproduzir o resultado [1]. Segundo Buneman e Tan [14], manter um registro completo de como o cálculo ou o processamento foram realizados é essencial para: (a) assegurar a repetitividade, (b) catalogar o resultado, (c) evitar a duplicação de esforços, e (d) recuperar as fontes de dados a partir dos dados de saída.

Segundo Bose e Frew [3], os principais benefícios da proveniência para a qualidade de dados são:

- Comunica a qualidade de dados: confiabilidade, adequação, acurácia, atualidade, redundância;
- Melhora a interpretação do dado em função do reconhecimento da fonte;
- Contribui para a justificativa do uso de um determinado dado;
- Reduz a possibilidade de erros no juízo da precisão do dado;
- Permite que usuários não especialistas em dados entendam os passos do processamento;
- Permite identificar o processo utilizado para a condução da criação de dados científicos;
- Permite atualização de dados a partir de visões relacionais;
- Permite a modificação de schemas de visões relacionais;
- Possibilita o uso de fontes de dados históricas.

Tais benefícios revelam a aplicabilidade direta da proveniência de dados para obtenção de métricas de qualidade de dados.

B. Benefícios da proveniência para experimentos de software

Bose e Frew [3] apresentam os principais benefícios da proveniência para o processamento científico. Uma das tarefas da Engenharia de software experimental é definir e tratar experimentos controlados acerca do software e neles podem ser usadas diversas aplicações científicas para capturar, armazenar e analisar dados. A Tabela II

apresenta os benefícios da proveniência de dados para o processamento científico e seu mapeamento com as fases de experimentos de software, indicando em que fases eles podem ser obtidos.

Tabela 2 - Benefícios da proveniência para experimentos de software e as fases onde são obtidos

Benefícios da proveniência para o processamento científico [3]	**Fases do experimento de software onde são obtidos**
Grava o histórico de processamento para gravações internas, auditoria e controle de qualidade.	Análise Replicação
Grava o histórico computacional para julgamento da validade estatística de operações futuras	Análise Replicação
Provê uma documentação consistente para conjuntos de dados distribuídos	Empacotamento Replicação
Encontra as fontes de falhas e saídas de processamento anômalas	Análise
Encontra as saídas afetadas por entradas falhas, de processamentos anômalos	Análise
Salva as "receitas" de processamento; modifica e re-executa sequências de processamento	Execução Replicação
Otimiza o tamanho da base de dados ao criar produtos intermediários sob demanda.	Execução
Auxilia as medidas de propagação de erros durante o processamento	Planejamento Análise
Dá suporte ao versionamento de objetos para modelagem colaborativa de bases de dados científicas.	Definição Planejamento Execução Análise Empacotamento
Permite um sistema de monitoramento identificar a fonte de dados falhos coletados em redes.	Planejamento Análise Replicação
Permite um centro de informação notificar suas fontes de dados após operações de limpeza de dados.	Replicação

4 UM MODELO DE QUALIDADE DE DADOS PARA EXPERIMENTOS DE SOFTWARE

O modelo de qualidade de dados para experimentos de software é baseado na noção de dimensões de qualidade e de proveniência. Dimensões de qualidade são atributos utilizados para medir a qualidade dos dados [25]. Cada dimensão selecionada para este modelo é definida e depois justificada no contexto de experimentação de software. Além disso, uma proposta de dados de proveniência é associada à dimensão. As dimensões podem ser aplicadas em todas as categorias de técnicas de coleta de dados.

A. Credibilidade

Definição: Credibilidade (*Believability*) representa o quanto o dado é considerado verdadeiro.

Importância para experimentos de software:

1. *Planejamento*: durante o planejamento de um experimento as fontes de dados podem ser selecionadas de acordo com sua credibilidade.
2. *Análise*: nesse momento os resultados do experimento são verificados e sua credibilidade calculada. Dessa forma é possível encontrar problemas na geração dos dados, como a consistência ao longo do tempo ou ao longo das fontes de dados.
3. *Empacotamento*: ao empacotar o experimento, deve-se também publicar os dados utilizados, assim como os dados e procedimentos para o cálculo da credibilidade por terceiros.
4. *Replicação*: na replicação de experimentos, dados de estudos anteriores podem ser usados para compor uma base de conhecimento acerca do objeto de estudo. Poder medir sua credibilidade é importante para a validação dos resultados gerais do experimento.

A falta de credibilidade dos dados tem efeito geral sobre os experimentos de software, pois ameaçam a validade do experimento e desencorajam a replicação dos experimentos. Cabe ressaltar que, em geral, os dados em experimentos de software são produzidos no próprio experimento ou obtidos de replicações anteriores. Não é comum

(ainda) obter dados de terceiros, como ocorre nas ciências biomédicas em seus experimentos.

B. Livre de Erros

Definição: Essa dimensão mede o quanto o dado é correto e confiável.

Importância para experimentos de software:

1. *Análise*: durante a análise dos dados gerados pelo experimento (que pode ocorrer várias vezes no experimento se ele for iterativo), deve-se calcular a corretude e confiabilidade dos dados para evitar problemas de propagação de erros que ameaçam a validade dos experimentos.
2. *Empacotamento*: ao exportar os dados para um pacote de laboratório, publicar as medidas dessa dimensão e sugerir o que pode ser feito para melhorar o processo de experimento e, consequentemente, seus dados.

C. Segurança

Definição: O quanto o dado é apropriadamente restrito para manter sua segurança.

Importância para experimentos de software:

1. *Planejamento*: em experimentos com humanos, uma das questões é assegurar a confidencialidade dos participantes do experimento, tanto para preservá-los, quanto para não influenciar a análise dos resultados. O grau de segurança deve ser definido neste momento.

2. *Análise:* durante esta fase, é importante verificar se os dados sensíveis, como identidade dos participantes, nome dos projetos analisados etc., estão protegidos como planejado.

3. *Empacotamento:* ao exportar os dados para um pacote de laboratório, deve-se assegurar que os dados sensíveis estão mascarados ou protegidos. A verificação dessa dimensão auxiliará nesta tarefa.

4. *Replicação:* ao utilizar pacotes de laboratório de terceiros e/ou dados de terceiros, verificar a dimensão de segurança para verificar se atende as definições do pacote de experimento.

A Tabela III sumariza as dimensões sugeridas e possíveis métricas de proveniência de dados associadas.

Tabela 3 - credibilidade e possíveis métricas de proveniência de dados

Dimensão	**Possível métrica de proveniência**
Credibilidade	Calcular a proveniência de acordo com as métricas [27]: • *Trustworthiness of source*, • *Reasonableness of data*, • *Temporality of data.* Escala varia entre 0 (total ausência de qualidade) e 1 (qualidade perfeita).
Livre de erros	Calcular o número de erros total dividido pelo número total de unidades de dados subtraído de 1. Determinar o que é uma unidade de dados e o que é um erro requer critérios bem definidos e uma precisão associada. Eles que podem variar caso a caso, dependendo do experimento.
Segurança	Checagem estática: Desenvolver um checklist de itens checando as propriedades de segurança que o experimento deve ter (encriptação, níveis de acesso etc). Inspecionar o projeto do experimento, calcular o número de itens positivos e comparar com a taxa mínima de segurança aceitável (definida pela equipe do experimento, universidade ou centro de pesquisa). Checagem dinâmica: Calcular o número total de acessos não autorizados e comparar com a taxa máxima aceitável.

5 CONCLUSÃO

A proveniência de dados auxilia o rastreamento da origem dos dados e processos subsequentes pela associação entre origem e estado em que o dado se encontra. Incorporar um modelo de qualidade de dados e proveniência no processo de experimentação pode nos ajudar a aumentar a validade dos experimentos, uma vez que a confiabilidade dos dados será monitorada. Este trabalho apresentou um mapeamento

entre a proveniência de dados e as fases de experimentos de software, e um modelo de qualidade de dados baseado em proveniência que pode ser aplicado nas fases da Engenharia de software experimental. Como trabalho futuro vamos criar um modelo concreto de proveniência de dados e avaliá-lo por meio de múltiplos estudos de caso.

REFERÊNCIAS

[1] Freire, J., Koop, D., Santos, E., Silva, C.T.. "Provenance for Computational Tasks: A Survey," *Computing in Science and Engineering*, vol. 10, no. 3, pp. 11-21, May/June, 2008.

[2] Veregin, H and Lanter, D. P.. Data-Quality Enhancement Techniques In Layer-Based Geographic Information Systems. Computers, Environment and Urban Systems, 19(1):23–36, January/ February 1995. Elsevier Science Ltd., Oxford, UK. ISSN:0198-9715, doi/10.1016/0198-9715(94)00032-8.

[3] Bose, R. and Frew, J. 2005. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.* 37, 1 (Mar. 2005), 1-28

[4] Basili, V. R., Shull, F and Lanubile, F. Building knowledge through families of experiments. IEEE Transactions on Software Engineering, 25(4):456–473, 1999.

[5] Victor R. Basili. The past, present, and future of experimental software engineering. Journal of the Brazilian Computer Society (JBCS), 12(3):7–12, 2006.

[6] Forrest Shull, Manoel G. Mendoncça, Victor Basili, Jeffrey Carver, José C. Maldonado, Sandra Fabbri, Guilherme Horta Travassos, and Maria Cristina Ferreira. Knowledg-sharing issues in experimental software engineering. Empirical Software Engineering, 9(1-2):111–137, 2004.

[7] Miller, J.. Replicating software engineering experiments: a poisoned chalice or the holy grail. Information and Software Technology, 47(4):233–244, March 2005.

[8] Jo E. Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal. A survey of controlled experiments in software engineering.IEEE Transactions on Software Engineering, 31(9):733–753, 2005. Member-Sjoberg„ Dag I. K.

[9] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, Björn and Wesslén, A.. Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[10] Buneman, P., Khanna, S. and Tan, W.C.. Why and where: A characterization of data provenance. In: 8th International Conference on Database Theory, London. p. 4-6, 2001.

[11] Buneman, P.; W.Tan. Provenance in databases. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, China, 2007.

[12] Shull, F., Basili, V.R., Carver, J., Maldonado, J.C., Travassos, G.H., Mendonça, M. and Fabbri, S.C.F.P.. Replicating software engineering experiments: Addressing the tacit knowledge problem. In Proceedings of the 2002 International Symposium on Empirical Software Engineering (ISESE'02), page 7, Washington,DC, USA, 2002. IEEE Computer Society.

[13] Mendonça, M. G., Maldonado, J. C., Oliveira, M.C.F., Carver, J., Fabbri, S.C.F.P., Shull, F. Travassos, G.H., Höhn, E.N., Basili, V. R.. A framework for software engineering experimental replications. In Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'08), pages 203–212, Washington, DC, USA, 2008. IEEE Computer Society.

[14] Travassos, G. H., Santos, P.S.M., Mian, P.G., Neto, A.C.D. and Biolchini, J.. An environment to support large scale experimentation in software engineering. In Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'08), pages 193–202, Washington, DC, USA, 2008. IEEE Computer Society.

[15] Sjøberg, D. I., Dybå, T., and Jørgensen, M. 2007. The Future of Empirical Methods in Software Engineering Research. In *2007 Future of Software Engineering* (May 23 - 25, 2007). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 358-378.

[16] Price, R. and Shanks, G. 2008. Data Quality Tags and Decision-making: Improving the Design and Validity of Experimental Studies. In *Proceeding of the 2008 Conference on Collaborative Decision Making: Perspectives and Challenges* P. Zaraté, J. P. Belaud, G. Camilleri, and F. Ravat, Eds. Frontiers in Artificial Intelligence and Applications, vol. 176. IOS Press, Amsterdam, The Netherlands, 233-244.

[17] Mattoso, M., Werner, C., Travassos, G., Braganholo, V., Murta, L., 2008, "Gerenciando Experimentos Científicos em Larga Escala". In: SEMISH - CSBC, Belém, Pará -Brasil.

[18] Simmhan, Y. L. Provenance framework in support of data quality estimation. 2007. 350 f. Doctoral thesis, Indiana University, Indianapolis, USA, 2007.

[19] D'Ambros, M., Gall, H. C. Lanza, M. and Pinzger. M.. Software Evolution, ch. Analyzing software repositories to understand software evolution, pages 37–67. SpringerLink, 2008.

[20] Lile Hattori, Gilson dos Santos Jr, Fernando Cardoso, and Marcus Sampaio. Mining software repositories for software change impact analysis: a case study. In SBBD '08: Proceedings of the 23rd Brazilian symposium on Databases, pages 210–223, Porto Alegre, Brazil, Brazil, 2008. Sociedade Brasileira de Computação.

[21] Miles, S., Groth, P. M. Branco, and L. Moreau, "The requirements of recording and using provenance in e-Science experiments," in Technical Report, Electronics and Computer Science, University of Southampton, 2005.

[22] Lethbridge, T. C., Sim, S. E., and Singer, J. 2005. Studying Software Engineers: Data Collection Techniques for Software Field Studies. *Empirical Softw. Engg.* 10, 3 (Jul. 2005), 311-341.

[23] Hartig, O.: Provenance Information in the Web of Data. Proceedings of the 2nd Workshop on Linked Data on the Web (LDOW2009).

[24] Freimut, B., Punter, T., Biffl, S., Ciolkowski M., 2002. State-of-the-art in empirical studies. IESE-Report No. 017.02/E & ViSEK Report No. 007/02. Kaiserslautern, Fraunhofer IESE.

[25] Wang, R. Y. A. Product Perspective on Total Data Quality Management. Communications of the ACM, V. 41, no 2., february, 1998.

[26] Lee, Y. , Pipino, L., Funk, J. and Wang, R.. Journey to Data Quality, MIT Press, Cambridge, MA, 2006.

[27] Prat, N. and Madnick, S.. "Measuring Data Believability: a Provenance Approach", Proceedings of HICSS-41, Big Island, HI, USA, January 2008.

[28] Marinho, A. S., PROVMANAGER: Uma Abordagem para Gerenciamento de Proveniência de Experimentos Científicos. Dissertação de Mestrado. UFRJ/COPPE, Rio de Janeiro, RJ, 133pp., 2011.

[29] ProvChallenge, Fourth Provenance Challenge, http://twiki.ipaw.info/bin/view/Challenge/FourthProvenanceChallenge, acesso em (2011).

[30] Basili, V. R. and D. Rombach. "The TAME Project: Towards Improvement-Oriented Software Environments," IEEE Trans. on Software Engineering, 14 (6), June 1988, pp. 758-773.

[31] Miles, S., Groth, P., Deelman, E., Vahi, K., Mehta, G., Moreau,L.. "Provenance: The Bridge Between Experiments and Data," Computing in Science and Engineering, pp. 38-46, May/June, 2008