

SSC5906 Engenharia de Software Experimental

Profs. Rosana T. Vaccare Braga e Paulo C. Masiero
1o. Semestre de 2017

MOTIVAÇÃO

A Engenharia de Software Experimental objetiva o aprimoramento da Engenharia de Software a partir da aplicação de uma abordagem científica, incorporando princípios de Engenharia na construção de novos processos, métodos e técnicas para apoiar o desenvolvimento de software. É uma área de pesquisa da Engenharia de Software e, portanto, a disciplina busca fornecer uma visão da área de Engenharia de Software Experimental, com ênfase na condução de experimentos em engenharia de software.

ESTRUTURA

O Curso será desenvolvido em três frentes:

- aulas expositivas, presenciais, nas quais os principais conceitos sobre ESE serão apresentados;
- leitura e discussão de artigos da área;
- desenvolvimento de um projeto experimental, no qual o aluno deverá escolher um tema, desenvolver o trabalho experimental, descrevê-lo na forma de um artigo e apresentá-lo como numa conferência.

MATERIAL

Todo material será disponibilizado na página do curso no Moodle/Stoa da USP. Slides, vídeos, artigos, discussões no fórum da disciplina são alguns dos materiais a serem utilizados. A referência básica será o livro: Wohlin, C. Experimentation in Software Engineering, Kluwer Academic Publishers, 2012. Versão pdf está disponível em:

<http://link.springer.com/book/10.1007%2F978-3-642-29044-2>

ARTIGOS

Serão lidos e discutidos vários artigos relacionados com ESE. A lista de artigos está no final deste documento, e desta lista serão escolhidos 1 ou 2 artigos no final de cada aula para serem lidos por todos os alunos nas semanas seguintes.

A discussão deve envolver todos os alunos e será feita de forma assíncrona, ou seja, por meio do fórum da disciplina na Internet. No dia da aula serão sorteados dois alunos para avaliar cada artigo escolhido. Um dos alunos deve fazer o papel de relator do artigo e o outro de detrator (crítico). O relator do artigo deve fazer um resumo de acordo com o gabarito sugerido. Os demais alunos,

incluindo o detrator, devem participar da discussão emitindo sua opinião, levantando dúvidas, fazendo comentários sobre o artigo e sobre a avaliação inicial dos colegas. Os responsáveis pelas revisões têm 48 horas após o término da aula para postar no fórum as suas revisões. Os demais têm até às 6h da manhã do dia anterior à aula seguinte para postar seus comentários. Durante a aula, o relator terá de 5 a 10min para fazer um resumo do artigo, enfatizando seus pontos principais e sua contribuição. Os demais alunos podem complementar a discussão e trazer a tona pontos interessantes ou polêmicos.

PROJETO

Cada dupla de alunos vai planejar, executar e relatar (de forma escrita e oralmente) um experimento na área de Engenharia de Software. O projeto deve ser feito ao longo da disciplina. Alguns temas poderão ser definidos pelo docente ou pelos estudantes, que podem também propor temas de seu interesse, desde que sejam aprovados pelo docente. No início do semestre o aluno deve descrever o tema e como pretende conduzir o experimento e os colegas poderão criticar e dar sugestões.

AVALIAÇÃO

As notas – entre 0 e 10 – serão baseadas nos seguintes elementos, com os respectivos pesos:

1. participação na discussão dos artigos (como relator, detrator ou comentarista em geral) e em todas as discussões, perguntas e revisões lançadas no fórum da disciplina. Espera-se que os alunos participem ativa e energeticamente das discussões, contribuindo com comentários pertinentes e relevantes. “Eu concordo com o Zé” não é uma contribuição relevante. “O Zé disse que o número de objetos utilizado no experimento é pequeno, mas nesse caso específico acho que é suficiente porque blá blá blá...” é uma contribuição relevante e será considerada como participação. Participação em sala de aula também será considerada da mesma forma (15%);
2. projeto experimental, que será avaliado considerando-se, o planejamento, a operação análise de dados e o artigo produzido, bem como a apresentação oral. Eventualmente, poderão ser consideradas replicações de experimentos já realizados (55%);
3. artigo do projeto experimental, que será avaliado pelo docente e por dois colegas, no final do semestre (30%)*

*OBS: o trabalho de avaliação dos artigos dos colegas não entra nesse item, mas no item 1 acima

Os conceitos serão atribuídos de acordo com a seguinte tabela:

Nota	Conceito
9,1 a 10,0	A
7,6 a 9,0	B
5,0 a 7,5	C
< 5,0	D

CALENDÁRIO

Semana 1 – 22/3	
Apresentação da disciplina e do material. Apresentação individual dos alunos, aprovação do calendário. Discussão sobre o projeto experimental. Apresentação da lista de artigos básicos dos quais um subconjunto será lido ao longo do bimestre. Aula-1 Introdução. Aula-2 Identificação e Definição	
Semana 2 – 5/4	
Discussão sobre o projeto experimental Discussão sobre o artigo 1 e 2 Aula-3 Planejamento Entregar até 17/4 uma 1ª versão de proposta de trabalho experimental * (alunos)	
Semana 3 – 26/4	
Discussão sobre o artigo 3 Breve apresentação das propostas dos alunos (5 a 10min) Aula-4 Planejamento (cont)/Execução Entregar até 5/5 a 2ª versão do planejamento do trabalho experimental (alunos) **	
Semana 4 – 10/5	
Discussão do artigo 4 Aula-5 Execução (cont)/Análise de dados Discussão sobre as propostas de experimentos	
Semana 5 – 31/5	
Discussão do artigo 5 Aula-6 Análise de dados (cont)/Estatística Aula-7 Apresentação e Empacotamento Discussão sobre o andamento dos experimentos Entregar artigo sobre os experimentos até 18/6	
Semana 6 – 21 /6	
Discussão do artigo 6 Apresentação dos artigos dos experimentos Discussão dos resultados	

* os professores revisarão a primeira versão e farão comentários.

** de 5/5 até 18/6: executar o experimento, analisar os dados e escrever o artigo

Artigos para leitura e discussão

GRUPO I – Métodos de experimentação

1. Norman Fenton, Shari Lawrence Pfleeger and Robert L. Glass, "Science and Substance: A Challenge to Software Engineers", IEEE Software, 11(4):86-95, July 1994.
2. Walter F. Tichy, "Hints for Reviewing Empirical Work in Software Engineering", Empirical Software Engineering, 5(4):309-312, December 2000.
3. Andreas Zendler, "A Preliminary Software Engineering Theory as Investigated by Published Experiments", Empirical Software Engineering, 6(2):161-180, June 2001.
4. Warren Harrison, "Editorial: Open Source and Empirical Software Engineering", Empirical Software Engineering, 6(3):193-194, September 2001.
5. Forrest Shull, Manoel G. Mendonça, Victor Basili, et al. "Knowledge-Sharing Issues in Experimental Software Engineering", Empirical Software Engineering, 9(1-2):111-137, March 2004.
6. Per Runeson, and Martin Höst, "Guidelines for conducting and reporting case study research in software engineering", Empirical Software Engineering, 14(2):131-164, April 2009.
7. Forrest Shull, Jeffrey Carver, Sira Vegas and Natalia Juristo, "The role of replications in Empirical Software Engineering", Empirical Software Engineering, 13(2):211-218, April 2008.
8. Barbara Kitchenham, "The Role of Replications in Empirical Software Engineering—A Word of Warning", Empirical Software Engineering, 13(2):219-221, April 2008.
9. Jeff Offutt, "Editorial: Standards for reviewing papers", Journal of Software Testing, Verification and Reliability, 17(3):135-136, September 2007.

GRUPO II – Métricas e complexidade

1. Burak Turhan, Tim Menzies, Ayse B. Bener and Justin Di Stefano, "On the Relative Value of Cross-Company and Within-Company Data for Defect Prediction", 14(5):540-578, October 2009.
2. Denys Poshyvanyk, Andrian Marcus, Rudolf Ferenc and Tibor Gyimóthy, "Using Information Retrieval Based Coupling Measures for Impact Analysis", 14(1):5-32, February 2009.
3. Lionel Briand and J. Wuest, "Empirical Studies of Quality Models in Object-Oriented Systems", Advances in Computers, vol. 56, 2002.
4. Norman Fenton and Niclas Ohlsson, "Quantitative Analysis of Faults and Failures in a Complex Software System", IEEE Transactions on Software Engineering, 26(8):797-814, August 2000.

GRUPO III – Teste

1. Mark Hennessy and James F. Power, "Analysing the Effectiveness of Rule-Coverage as a Reduction Criterion for Test Suites of Grammar-Based Software", 13(4):343-368, August 2008.

2. Yu-Seung Ma, Jeff Offutt and Yong Rae Kwon, "MuJava: An Automated Class Mutation System", Journal of Software Testing, Verification and Reliability, 15(2):97-133, June 2005.
3. Mats Grindal, Birgitta Lindstrom, Jeff Offutt and Sten F. Andler, "An Evaluation of Combination Strategies for Test Case Selection", Empirical Software Engineering, 11(4):583-611, December 2006.

GRUPO IV – Segurança (Proteção)

1. Toan Huynh and James Miller, "An Empirical Investigation Into Open Source Web Applications' Implementation Vulnerabilities", 15(5):556-576, October 2010.

GRUPO V – Confiabilidade

1. Carina Andersson, "A Replicated Empirical Study of a Selection Method for Software Reliability Growth Models", Empirical Software Engineering, 12(2):161-182, April 2007.

GRUPO VI – Manutenção e evolução

1. Daryl Posnett, Christian Bird and Prem Dévanbu, "An Empirical Study on the Influence of Pattern Roles on Change-Proneess", 16(3):396-423, June 2011.
2. Stephen R. Schach, Bo Jin, Liguu Yu, Gillian Z. Heller and Jeff Offutt, "Determining the Distribution of Maintenance Categories: Survey versus Measurement", Kluwer's Empirical Software Engineering, 8(4):351-365, December 2003.

GRUPO VII – Projeto e modelagem

1. Anne Martens, Heiko Koziulek, Lutz Prechelt and Ralf Reussner, "From monolithic to component-based performance evaluation of software architectures: A series of experiments analysing accuracy and effort", 16(5):587-622, October 2011.
2. John C. Knight and Nancy G. Leveson, "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming", IEEE Transactions on Software Engineering, SE-12(1):96-109, January 1986.
3. Wojciech James Dzidek, Erik Arisholm and Lionel C. Briand, "A Realistic Empirical Evaluation of the Costs and Benefits of UML in Software Maintenance", IEEE Transactions on Software Engineering, 34(3):407-432, May/June 2008.

GRUPO VIII – HCI

1. Richard J. Miara, Joyce A. Musselman, Juan A. Navarro and Ben Shneiderman, "Program Indentation and Comprehensibility", Communications of the ACM, 26(11):861-867, November 1983.
2. Simone S. Borges et all. Reduced Gui for na interactive geometry software: Does it affect student's performance? Computers in Human Behavior, 2016, pages 124-33.

GRUPO IX – Requisitos

1. Carol K. Gonzales and Gony Lero, "Eliciting User Requirements using appreciative Inquiry", 16(6):733-772, December 2011.

GRUPO X – Processos e gerenciamento

1. Mats Grindal, Jeff Offutt and Jonas Mellin, "On the Testing Maturity of Software Producing Organizations", Testing: Academia & Industry Conference – Practice And Research Techniques, TAIC PART 2006, pages 171-180, August 2006 Windsor, UK.
2. M. Pikkariainen, J. Haikara, O. Salo, P. Abrahamsson and J. Still, "The Impact of Agile Practices on Communication in Software Development", 13(3):303-337, June 2008.

GRUPO XII – Outros

1. Adam Zajac, Jaroslaw Cholewa, Stanislaw Poprzecki, Zbigniew Waszkiewicz and Jozef Langfort, Effects of sodium bicarbonate ingestion on swim performance in youth athletes, Journal of Sports Science and Medicine (2009) 8, 45-50
2. David Budgen et al. Presenting Software engineering results using structured abstracts: a randomized experiment. Empirical Software Engineering, 13,4, 2008, pages 435-68.
3. Marco Mello. Que teste estatístico devo usar? Blog.