Call for papers

Journal of Automated Software Engineering
Special Issue: Learning to Organize Testing

EDITORS:
Prof Ayse Bener (TRSITM, Ryerson University) ayse.bener@ryerson.ca
Prof Tim Menzies (West Virginia University, US) tim@menzies.us

At the start of the decade, two publications[1,2] described the start-of-the art in defect reduction. Since then, there has been considerable research into data mining of defect data; e.g. [3]. The data mining work has become less about *defect reduction*, and more about how to *organize a project's test resources* in order to *improve product quality* by (say) defining a procedure such that the modules most likely to contain defects are inspected first [4].

After a decade of intensive work into data mining to make best use of testing resources, it is time to ask: what have we learned from all that research? Some of that research offers success stories with (e.g.)

- Reducing the costs to find defects [4];
- Generalizing defect predictors to other projects [5];
- Tuning those predictors to different business goals [6].

But other research offers the cautions that:

- defect predictors may not generalize to other projects [7];
- Despite much effort on data mining and defects, most of that work achieves similar conclusions [8];
- Data mining data is fundamentally less important than discussing those effects with the users [9]

The above references sample just a small subset of the research performed this decade on data mining and software defects. We seek papers that document, review, and extend this work. Do the insights from the start of the decade still hold? Has anything extra really been learned in the meanwhile? If we wrote an article "What We Have Learned About Organizing Testing Resources" in 2010, what would we write in such an article, that has been *verified using publicly available data sets*?

For this special issue, we seek papers about or progress (or lack of progress) in using data mining to organize test resources in order to (say) fight defects. Papers are required to offer verifiable results; i.e. they must be based on **public-domain data sets**. Submissions should come with an attached note offering the URL of the data used to make the paper's conclusions. A condition of publication for accepted papers is that their data must be transferred to the PROMISE repository (http://promisedata.org/data) prior to final acceptance.

DATES

Dec 15 2010:	submission
Feb 2011:	reviews, round 1
April 2011:	resubmit revised papers
May 2011:	notifications of acceptance

SUBMISSION

Submit to http://www.editorialmanager.com/ause/, adhering to the instructions for authors at http://www.springer.com/computer/ai/journal/10515. On submission, please include a note saying "For the special issue on Learning to Organize Testing".

REFERENCES

1. Forrest Shull, Victor R. Basili, Barry W. Boehm, A. Winsor Brown, Patricia Costa, Mikael Lindvall, Daniel Port, Ioana Rus, Roseanne Tesoriero, Marvin V. Zelkowitz: What We Have Learned About Fighting Defects. IEEE METRICS 2002: 249. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.1413
2. B. Boehm,V. Basili, "Software Defect Reduction Top 10 List,"IEEE Computer,34(1): 135-137, Jan 2001. http://www.cs.umd.edu/projects/SoftEng/ESEG/papers/82.78.pdf
3. Tim Menzies, Jeremy Greenwald, Art Frank: Data Mining Static Code Attributes to Learn Defect Predictors. IEEE TSE 2007, January, 2-13
4. Defect prediction from static code features: current results,imitations, new approaches, Tim Menzies, Zach Milton, Burak Turhan, Bojan Cukic, Yue Jiang, Ayse Bener, Journal Automated Software Eng, Dec 2010.http://menzies.us/pdf/10which.pdf
5. AI-Based Software Defect Predictors:Applications and Benefitsin a Case Study (Deployed). Ayse Tosun, Ayse Bener, Resat Kale. Innovative Applications of Artificial Intelligence, 2010
6. On the relative value of cross-company and within-company data fordefect prediction, B Turhan, T Menzies, A B Bener, J D Stefano, Journal of Empirical Software Engineering, 2009. Pages 278-290. http://menzies.us/pdf/08ccwc.pdf
7. T. Zimmermann, N. Nagappan, H. Gall, E. Giger, Brendan Murphy: Cross-project defect prediction: a large scale experiment on data vs. domain vs. process. ESEC/SIGSOFT FSE 2009:91-100
8. Stefan Lessmann, Bart Baesens, Christophe Mues, Swantje Pietsch:Benchmarking Classification Models for Software Defect Prediction: A Proposed Framework and Novel Findings. IEEE TSE, July 2008. 485-496
9. N. E. Fenton, M. Neil, W. Marsh, P. Hearty, Lukasz Radlinski, Paul Krause: On the effectiveness of early life cycle defect prediction with Bayesian Nets. Journal of Empirical SE, October 2008, p499-537