Actionable Analytics for Software Engineering

Ye Yang, Stevens Institute of Technology
Davide Falessi, California Polytechnic State University, San Luis Obispo
Tim Menzies, North Carolina State University
Jairus Hihn, Jet Propulsion Laboratory
IN A RECENT article, Lionel Briand and his colleagues emphasized the importance of context-driven software engineering research, to improve the significant disconnect between research and industrial needs.1 Now, after nearly a decade of intensive research on software analytics, industrial practitioners are asking, “And what can we do with all that research?” A repeated complaint is that industrial practitioners have trouble applying the results generated from data science. Some researchers—for example, Ayse Bener and her colleagues2—have taken care to describe how to map their work onto change points for organizations. However, the vast majority of researchers don’t even consider this question important.

The Importance of Actionable Analytics
At a panel at the 2012 International Conference on Software Engineering, industrial practitioners lamented the state of the art in software analytics. The prevailing opinion was, “Prediction is all well and good, but what about decision making?” The panelists were interested more in the interpretations and follow-up that occur after the mining than in just the mining itself. It’s unarguable that in practice, we must be able to take action. Sometimes simple methods are as good as or better than fancy ones, as long as they’re applicable and work well.

When you look at recent software-analytics studies at major software engineering conferences, you can still observe effort in reinventing prediction models or pointlessly perfecting model performance. In today’s software systems, system complexity and the speed of upgrades are ever-increasing. Software analytics must go beyond classic predictability problems such as effort, cost, schedule, and defects, to help us understand and react to constant changes and help us forecast consequences and mitigation actions. In another words, only actionable analytics can increase software development’s agility and success.

This theme issue aims to reflect on the decades of the software-analytics journey and examine the gaps between the proliferation of software-analytics methods and what’s really needed in practices that drive business value. Extending two previous related special issues,3–4 this issue seeks to reflect on questions such as, What types of information do practitioners need in walking the final miles from predictions to plans? How do we link such information needs with software analytics to make them more actionable? How should software analytics continuously change the way we envision, implement, and deliver software? To answer such questions, actionable analytics must be more systematic, practice-oriented, time-sensitive, and context-aware.

In This Issue
In “Hybrid Labels Are the New Measure!” Maleknaz Nayebi and her colleagues discuss a promising approach for improving change triage, in which humans label a change with keywords. The results show that expert and machine labels are complementary and that the hybrid set of labels results in better similarity analysis and consequent prediction.

In “Actionable Analytics for Strategic Maintenance of Critical Software: An Industry Experience Report,” Dan Port and Bill Taber explain how to use a continuous collection of simple metrics, such as the number of bug reports and the time between them, to infer useful intangible system properties such as defect density. Such analytics can help build defensible business cases, such as for the required maintenance staff.

In “How Robust Is Your Development Team?,” Lu Xiao and her colleagues present an approach for visualizing and quantitatively evaluating team structure in a collaborative development environment. This approach helps practitioners evaluate team robustness on the basis of a measure of information loss derived from a developer social network.

In “Deploying Software Team Analytics in a Multinational Organization,” Vinay Augustine and his colleagues tell how the ABB corporation adopted actionable analytics in its software projects. Using business intelligence tools led to the automatic updating of key metrics and new insights and workflows.

In “Using Analytics to Guide Improvement during an Agile–DevOps Transformation,” Barry Snyder and Bill Curtis report how Fannie Mae applied structural-quality metrics to its agile–DevOps transformation. They report examples and analysis of project-level metrics and provide insights on those metrics’ alignment with enterprise-level metrics.

We thank all the authors and reviewers who worked so hard on this theme issue. We also thank Diomidis Spinellis and his team at IEEE Software for all their help throughout this process. ©
References

ABOUT THE AUTHORS
YE YANG is an associate professor in the Stevens Institute of Technology’s School of Systems and Enterprises. Her research interests include software crowdsourcing, software cost estimation, defect prediction, and software process modeling. Yang received a PhD in computer science from the University of Southern California. Contact her at ye.yang@stevens.edu.

DAVIDE FALESSI is an associate professor of software engineering at California Polytechnic State University, San Luis Obispo. His main research interest is in devising and empirically assessing scalable solutions for the development of software-intensive systems. Falessi received a PhD in computer engineering from the University of Rome Tor Vergata. Contact him at d.falessi@gmail.com.

TIM MENZIES is a full professor of computer science at North Carolina State University, where he teaches software engineering, automated software engineering, and the foundations of software science. He’s the director of the RAISE (Real-World AI for Software Engineering) Lab, which explores software engineering, data mining, AI, search-based software engineering, and open-access science. Menzies received a PhD in computer science from the University of New South Wales. Contact him at tim.menzies@gmail.com.

JAIRUS Hihn is a principal member of the engineering staff at NASA’s Jet Propulsion Laboratory, where he’s a subject matter expert on cost estimation, quantitative organizational and project management methods, and early lifecycle risk analysis. Hihn received a PhD in economics from the University of Maryland, College Park. Contact him at jairus.m.hihn@jpl.nasa.gov.

Read your subscriptions through the myCS publications portal at http://mycs.computer.org