

Open Source Research Ethics

I enjoyed Warren Harrison's From the Editor column "Whose Information Is It Anyway?" (July/August 03). Including open source projects in research studies can provide useful feedback to the open source industry, widen the scope of existing software engineering research, and give us a better understanding of a larger variety of software products. However, using open source data is still a debated issue from an ethical perspective.

Ethics are about what a community accepts to be right and wrong. In this note, I describe what I find to be the right ways of getting involved in the open source world as a researcher. I think that further discussions on this issue will be useful to our community.

I believe that researchers, like other individuals, should have a right to get involved in open source projects and contribute to them in their own unique way. When researchers begin working on an open source project, they should email the project's mailing list(s), introduce themselves, clearly explain the scope and goals of their study, and ask for help.

After that, they should feel responsible for contributing to the project while pursuing their research goals. This includes answering questions, exchanging ideas, contributing to the common knowledge, and keeping the project participants updated about the research results and final publications. By doing this, researchers automatically become active participants in the project. Then, they are entitled to comment on such issues as why the last re-

lease has more bugs, why progress is slow, and so on, just as other participants do.

In addition, I think researchers have a responsibility to make their results publicly accessible, so that many others can question their research methods and results. On the positive side, when the larger community can verify research

results on an open platform, it might be possible to reach healthier conclusions.

Researchers should also avoid naming developers in correspondence and publications, although this doesn't make tracing individual developers totally impossible. In my opinion, open source communities have already accepted the fact that nobody is error-free and integrated bug tracking systems into their processes. Therefore, developers should be open to the

opinions of others.

Developers need to be conscious about the fact that the data and artifacts they leave behind will be publicly available; they can hide their identity if they want. If our research community feels strongly about this issue, we can ask important open source projects and project-hosting Web sites such as sourceforge.net and freshmeat.net to post warnings or disclaimers on their developer Web sites. Then smaller projects can follow what they do. Presently, developers might not be aware of the issues discussed here. Developer information is even made available on some projects' Web sites (see <http://people.kde.org>).

A final note about the quality of data in open source repositories: Although the data might be



We welcome your letters. Send them to software@computer.org. Include your full name, title, affiliation, and email address. Letters are edited for clarity and space.

out there, it might not be suitable to be readily used in empirical studies. For example, if a researcher is interested in using defect data, preliminary studies about consistency in data collection, completeness of defect records, and so on, will be necessary.

A. Güneş Koru
PhD candidate
Southern Methodist University
gkoru@engr.smu.edu

Warren Harrison responds:

I agree that OSS is not only a convenient source of data, as more of it becomes available to the public, but also an important segment to study in itself.

Some studies are statistical in nature; there's no way to identify individual participants. On the other hand, there are tools and studies in which individuals are easily identified. Just because a

tool's author doesn't identify individuals directly, I don't think he or she is off the hook ethically if someone can use the tool (or the study) to identify individuals and it is made available to others.

For instance, suppose you have an

Developers should be aware that the data and artifacts they leave behind will be publicly available; they can hide their identity if they want.

OSS product created by one or two developers. If, in comparing various OSS products, you identify this product (or describe it in a way that makes its identification trivial) as having the highest bug rate, you have for all intents and purposes identified the individuals. Now suppose your analysis includes some judgmental aspect such as "most unmaintainable code," which is really a synthesis of metrics and opinions as opposed to a simple count of artifacts such as lines of code or number of comments. Then you've added insult to injury.

However, I believe the real burden is on those organizations and people that host this kind of data to start with. Either contributors should be given an explicit informed-consent option, or (preferably) the data should be stored in such a way that extracting individual identities is simply not possible.

It would indeed be interesting if aca-



The **B. Thomas Golisano College of Computing and Information Sciences (GCCIS)** is pleased to invite applications for tenure-track Assistant and Associate Professor positions in its Computer Science, Information Technology, and Software Engineering departments starting September 2004. Successful candidates should possess the Ph.D. in one of the specified fields or closely related area; they should have interests in teaching and research, and should have an ability to contribute in meaningful ways to the Institute's commitment to cultural diversity and pluralism. Exceptional candidates will be considered for the rank of Full Professor. Additional departmental requirements include:

- **Computer Science (CS):** Candidates should have an interest in intelligent systems in computer vision or machine learning, security, data mining, or human-computer interfaces.
- **Information Technology (IT):** Candidates should have strengths in one or more of the following areas: network design and protocols, database design and application development, security, gaming, visual object-based development, web and multimedia development, and computer-mediated experiences.
- **Software Engineering (SE):** Candidates must have a deep interest in education at the masters and undergraduate levels as well as a strong desire to conduct research on software process, software architectures and design, software quality assurance, software construction, and secure systems. Professional experience developing software is desirable.

GCCIS (www.rit.edu/~gccis) is RIT's newest college at the 1,300-acre suburban university located in Rochester, New York, just 90 minutes from Niagara Falls, 3 hours from Toronto, and 6 hours from NYC. In addition to CS, IT, and SE, the college is home to the **Lab for Applied Computing**, the research arm of the college. The college, with 90+ faculty members, is housed in a new 126,500 square foot state-of-the-art building with 13 classrooms, student team rooms, and laboratories, including specialty labs for security, vision, embedded systems, networking, systems administration, and streaming multimedia. Our 2,300 undergraduate students are enrolled in BS programs in IT, Applied Networking and System Administration, IT New Media, Computer Science, and Software Engineering. Our 600 graduate students are enrolled in IT, Software Development & Management, and Computer Science MS programs.

Interested applicants should submit the following information to the email address below by January 30, 2004: a summary of education and professional background, a current list of publications, a summary of teaching and research experience, the names of three references, and a brief description of future research plans. Please indicate for which department(s) you would like your application to be considered.

B. Thomas Golisano College of Computing and Information Sciences
Dean's Office - Faculty Search Committee
Rochester Institute of Technology
20 Lomb Memorial Drive, Rochester, NY 14623
Email: Faculty_Search@gccis.rit.edu

R·I·T
"providing career education over a lifetime"
RIT is an Affirmative Action/Equal Employment Opportunity Employer.

democratic studies of OSS began being reviewed by campus Human Subjects boards, as more traditional survey and controlled experiments already are.

Balancing both worlds

I enjoyed Bob Glass's last two Loyal Opposition columns. When I browse through so-called "software engineering" books, I find that authors rarely have much real-world experience. Of course there are a few exceptions, but rarely under the software engineering umbrella.

About the academic-versus-industry debate, I've found that balancing the two worlds is hard. Usually for professors, people in industry are just developers who write buggy code. And for industry people, professors only care about writing conference papers, not code. It's a pity. At the end of the day, we

must remember that we are basically part of the same business—like it or not.

Also, if you're trying to go back to school, most computer science departments don't care about your industrial experience; you're just another student. It would be nice if universities could offer two PhD tracks: industrial and academic. If you chose the industrial one, you should be able to work on more real-world problems. Perhaps that's the one for teaching software engineering.

Finally, I believe it's possible to balance both worlds. This fall I'll start my PhD in the field as a part-time student, much like James Cain said in the last issue's Letters to the Editor: one day a week. It's probably a long road, but I'm looking forward to it.

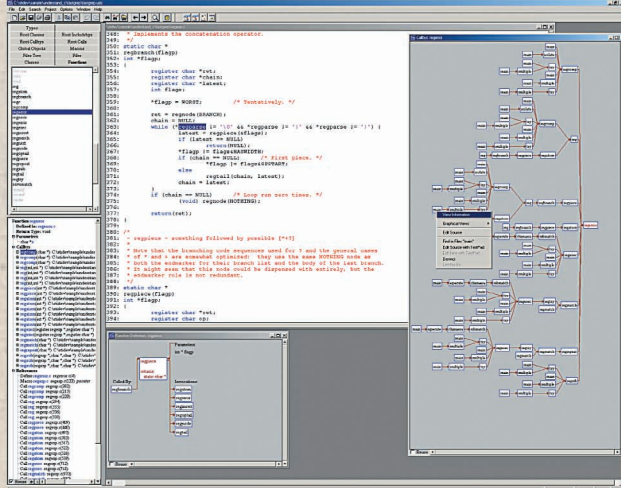
Omar Alonso
San Mateo, CA
oralonso@yahoo.com

The agile sweet spot

While it's always nice to be noticed, I fear that the July/August Manager column "Scaling Agile Methods" by Don Reifer, Frank Maurer, and Hakan Erdogan misrepresented my position on this topic. Naturally, I intended my sound byte "scaling agile methods is the last thing you should do" to be startling, but I also intended it to be precise. I'm not against scaling agile methods, but I think we should try other things first (<http://martinfowler.com/bliki/LargeAgileProjects.html>). In particular, we've seen a lot of evidence that large projects often show alarming levels of overstaffing and feature bloat. Dealing with these problems often turns a large project into one that fits nicely into the agile sweet spot.

Martin Fowler
Chief scientist, ThoughtWorks
<http://martinfowler.com>

Reverse Engineering Tools



Our tools help developers understand, document, and maintain impossibly large or complex amounts of source code.

They parse **Ada 83, Ada 95, FORTRAN 77, FORTRAN 90, FORTRAN 95, K&R C, ANSI C and C++, Java, JOVIAL, & PASCAL** source code to reverse engineer, automatically document, calculate code metrics, and help your engineers understand, navigate and maintain source code that has grown too large for one person (or even a group) to know.

Big projects aren't a problem. The tools can parse and later manipulate very large amounts of code. 1,000,000 SLOC projects (and larger) are common among our customers.

We also focus on exceptional customer support .based on rapid response from a real engineer and bug fixes and new features incorporated into weekly builds.

Key Features:

- Fast on big projects
- Quick and easy to use—no complicated or fussy setup, immediately useful
- PERL/C/C++ API for custom reports/documentation
- Automatic creation of graphics and documentation
- Export to common graphics formats and Visio
- Cross reference everything in source
- Variety of hierarchical and graphical views (including With Trees, Call Trees, Include Trees, Extended-By Trees, Ada Structure Graphs, and many others)
- Code colorizing source editor and printing
- Rapid code navigation and editing

Supported Languages:

- Ada 83 and Ada 95
- Java
- ANSI C, K&R C, and C++
- FORTRAN 77, 90, 95
- JOVIAL
- Pascal
- Can create custom languages on request

Supported Languages:

- Windows 95, 98, NT 4.0, 2000, XP
- Linux (Intel)
- Solaris
- HP-UX
- SGI Irix
- Alpha (OSF)

Download and try on your code:
<http://www.scitools.com/>

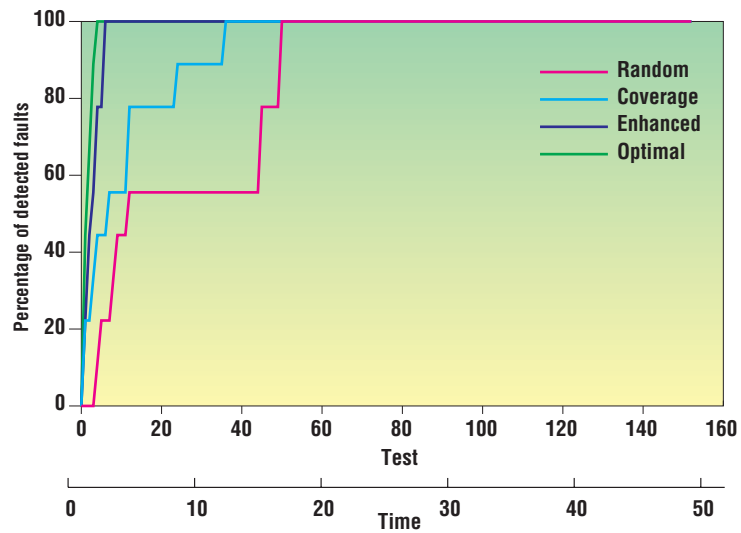
All downloads are fully functional, just time limited.

STI
Scientific Toolworks, Inc.
info@scitools.com
(802) 763-2995

More recent references

In the article "On the Declarative Specification of Models" (March/April 2003), the reference to E.R. Gansner et al. on drawing diagrams seems to suppose there are no approaches newer than 1993. See Springer-Verlag's *Formal Systems Specification: The RPC-Memory Specification Case Study*, 1996, and *Graph Drawing: 9th International Symposium, 2001; 1st International Workshop on Visualizing Software for Understanding and Analysis*, 2002; and the *ACM Symposium on Software Visualization*, 2003.

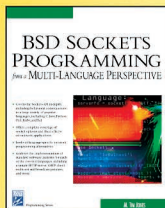
Holger Eichelberger
University of Würzburg
eichelberger@informatik.uni-wuerzburg.de



Correction

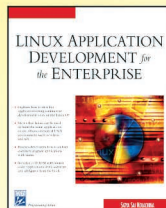
In our Nov./Dec. Quality Time column ("Putting Your Best Tests Forward" by Gregg Roethermel and Sebastian Elbaum), we erroneously switched the lines for "Op-timal" and "Coverage" in the figure depicting fault detection rates for four prioritized test suites. The correct figure is shown here.

Computer Books for Computing Success



1-58450-268-1 \$49.95

A practical handbook for developing applications with the BSD Sockets API using multiple languages, including C, Java, and Perl.



1-58450-253-3 \$54.95

The complete guide to developing commercial-quality applications for the Linux OS.



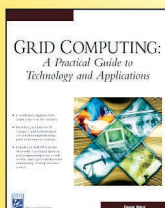
1-58450-246-0 \$49.95

The "all-in-one" guide to creating business applications with Apache Jakarta Commons and other open source tools.



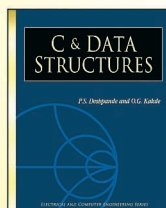
1-58450-278-9 \$54.95

A tutorial-based guide to adding *useful* artificial intelligence techniques to software projects using C.



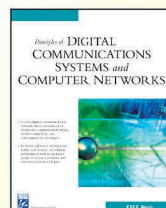
1-58450-276-2 \$49.95

A clear, readable, and pragmatic overview to all aspects of grid computing technology for programmers, engineers, and IT personnel.



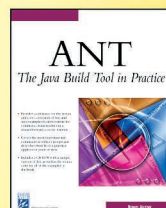
1-58450-338-6 \$59.95

A complete guide and reference to the C language and how to implement it effectively in data structures such as linked lists and trees.



1-58450-329-7 \$59.95

A complete resource for digital communications systems, data communication protocols, computer networking, and mobile computing.



1-58450-248-7 \$39.95

A desktop guide and reference for this Java-based application building tool.



Available at Amazon, Borders, Barnes & Noble, and other fine retailers

(800) 382-8505

www.charlesriver.com

20% OFF
Web Orders