

## Whose Information Is It Anyway?

Warren Harrison

I just returned from ICSE, the International Conference on Software Engineering. This is the premier academic software engineering conference and is cosponsored by the IEEE Computer Society and the ACM. For those of you who do not regularly attend ICSE, I strongly urge you to take the opportunity if you are ever fortunate enough to have it come to your town (this year it was held in Portland, Oregon, my home base).



### Harvesting software developer information

While at ICSE, I had a chance to hear an interesting and well-done presentation describing a software engineering tool that harvests information from email logs, revision control systems, and other development information repositories.

When this information is linked together, you can make inferences to do everything from identifying artifacts for reuse to discovering developers' favorite approaches to making modifications. To test the tool's viability, the presenters harvested information from an open source software product.

Although I can see how this aggregated information is useful, I had an uneasy feeling about the wider ramifications. On many (if not most) open source software projects, developer logs, change requests, and so on are all readily available for harvesting. I can imagine such systems being used to extract and synthesize information about individual software en-

gineers, answering questions such as "Which contributor had the most bugs?" or "Which contributor took the longest to make repairs?" But such systems, if available via the Internet, could also let a nosey manager find out that her employee had submitted a bug report to the Starship Galactica game site at 9:45 a.m. on Monday when he was supposed to be at an important meeting.

Some might argue that if this information is being harvested in a proprietary work environment, the manager rightfully should have access to it (of course, others might argue that this information is no one's business). However, when resources available on the Internet such as open source code and bug repositories are being harvested, this information is accessible to anyone who chooses to run the tool.

### Open source repositories

The tool described at ICSE is not the first research use of open source information repositories. Many applied software engineering researchers need to try their tools and methods on real software systems to see how well they work. Software metrics researchers were perhaps the first to encounter a general reluctance by industry to provide useful data and information for knowledge discovery, and they consequently turned to harvesting data from open source projects.

And the data is available. For instance, if you browse through the Mozilla bug repository at <http://bugzilla.mozilla.org>, you can run queries to retrieve lists of reported defects, including the resolution and the identification of

### DEPARTMENT EDITORS

**Bookshelf:** Warren Keuffel,  
wkeuffel@computer.org

**Construction:** Andy Hunt and Dave Thomas,  
{Andy, Dave}@pragmaticprogrammer.com

**Design:** Martin Fowler,  
fowler@acm.org

**Loyal Opposition:** Robert Glass,  
rglass@indiana.edu

**Manager:** Don Reifer,  
d.reifer@ieee.org

**Quality Time:** Nancy Eickelmann,  
nancy.eickelmann@motorola.com  
and Jane Hayes, hayes@cs.uky.edu

**Requirements:** Suzanne Robertson,  
suzanne@systemsguild.com

**Software Engineering Glossary:** Richard H. Thayer,  
thayer@csus.edu

### STAFF

Senior Lead Editor  
**Dale C. Strok**  
dstrok@computer.org

Group Managing Editor  
**Crystal Shif**

Associate Editors  
**Shani Murray and Dennis Taylor**

Assistant Editors  
**Rebecca Deuel and Denise Kano**

Editorial Assistant  
**Joan Hong**

Magazine Assistant  
**Pauline Hosillos**, software@computer.org

Art Director  
**Toni Van Buskirk**

Cover Illustration      Technical Illustration  
**Dirk Hagner**              **Alex Torres**

Production Assistant      Production Artist  
**Monette Velasco**        **Carmen Flores-Garvey**

Executive Director  
**David Hennage**

Publisher  
**Angela Burgess**

Assistant Publisher  
**Dick Price**

Membership/Circulation Marketing Manager  
**Georgann Carter**

Advertising Assistant  
**Debbie Sims**

### CONTRIBUTING EDITORS

Candace English, Kirk Kroeker, and  
Beth Wilson

**Editorial:** All submissions are subject to editing for clarity, style, and space. Unless otherwise stated, bylined articles and departments, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *IEEE Software* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society.

**To Submit:** Access the IEEE Computer Society's Web-based system, Manuscript Central, at <http://cs-ieee.manuscriptcentral.com/index.html>. Be sure to select the right manuscript type when submitting. Articles must be original and not exceed 5,400 words including figures and tables, which count for 200 words each.

```
$bug="99999";
$param="bugzilla.mozilla.org/show_bug.cgi?id=$bug";
$page=`lynx -dump http://$param`;
$page=~ /Reporter: ([\S]*)/;
print "$1\n";
```

**Figure 1. A simple Perl script can identify who reported the bug.**

the person who reported the bug. Building such harvesting engines is quite simple. For example, given the bug ID, the simple Perl script shown in Figure 1 will identify the person who reported the bug by first making a shell call to lynx to retrieve the Bugzilla Web page and then using a regular expression to extract the contributor's name.

You can write other queries with little more effort, allowing automated harvesting of all sorts of data. For instance, is a particular individual constantly reporting bugs with the ultimate resolution "WORKSFORME"? Although a researcher might want to harvest this information to characterize a software development environment, a hiring manager might want to screen out prospective quality assurance engineers who are too fast on the trigger in reporting bugs.

### Use versus intention

When I contribute a bug report to an open source software system, I do it for a specific reason—so that someone will fix a bug or add some functionality. If I contribute a software component, my intention is to provide some functionality to the whole. Is it reasonable to expect me to also anticipate other uses that might be made of my contributions? Should I expect that someone will critique my programming style or ability to write a bug report for the public at large, or that some day all the code contributors in an open source project will be ranked by bugs per KLOC?

I'm not a lawyer, but I think this after-the-fact harvesting of information without the contributor's consent is probably legal. Nevertheless, when you have 10 programmers, one is going to be the best and one is going to be the worst. So legality notwithstanding, I

might not appreciate other people poking around my code, finding stylistic faults, and comparing my programming ability to others' without my permission. Egoless programming is one thing in a small development team with a single focus, but it is an entirely different matter when your ranking appears on the World Wide Web.

Many of my rabid open source friends tell me that this isn't really an issue. They say that anyone doing open source must know that their work product is wide open for anyone to see. After all, that's the meaning of open source. They suggest that anyone concerned about such retroactive analysis use a pseudonym when contributing code or filing a bug report.

### Who is responsible?

A dozen years ago, few of us would have thought such widespread, systematic harvesting of information would even be possible, much less the kinds of uses to which it could be put. Many of us still don't truly understand how widely accessed the things we say and do on the Internet are, and how persistently that information remains available.

Virtually every piece of information you leave behind, from postings to newsgroups, to email sent to customer support sites, open source discussion lists, and item tracking systems, is maintained for an indefinite period of time. For instance, if you search for "poor support" using Google's Groups search option, you end up with hits dating from February 1996 discussing a database vendor's poor support practices. Over seven years old and still coming out at the top of 1.2 million hits! So don't be surprised if some day a bug report, a rant about customer service, or a piece of code you

**Correction**

As many readers have informed me, in my last column, I erroneously referred to Walker Royce as the author of “Managing the Development of Large Software Systems: Concepts and Techniques,” the famous 1970 article often cited as the genesis of the waterfall lifecycle model. The author actually was Winston Royce. My thanks to all who took the time to contact me regarding my mistake.

whipped up that ended up in an open source archive comes back to bite you—unless someone takes responsibility for making sure it doesn’t, that is.

One thought is that responsibility should fall to a repository’s “owner” to spell out exactly how widely accessible its artifacts are, if not monitor how the information is being used on others’ behalf. We expect such behavior from Yahoo and Amazon; why not from our favorite open source sites? Better yet, it would be nice if future repositories simply cloaked individual identities by default. Some open source sites, such as SourceForge.org, allow user names that appear to make users anonymous but then provide search tools that can associate real names with user names.

We should also expect researchers and others who harvest information from open source repositories to be sensitive to the custom of informed consent by asking participants to opt-

in if data that might identify them will be used. Perhaps the only thing worse than being identified as the least talented programmer associated with an open source effort is to be identified as such and not know it.

Ultimately, however, just like your personal safety, your personal privacy is your responsibility. Be aware what personal information you have out there and what kinds of things it says about you. If you have some thoughts about the issue of harvesting information from open source repositories, please write to me at warren.harrison@computer.org.

**Other reading**

You can read more about using data from open source software repositories in *Empirical Software Engineering’s* special issue on research ethics organized by Janice Singer and Norman Vinson (vol. 6, no. 4, Dec. 2001). ☞

**Software Engineering Glossary**

One of the major benefits of good software engineering practices is the improvement in communication that results from a common vocabulary. I’m delighted that *IEEE Software* will be doing its part in facilitating this common vocabulary among practitioners by adding a new feature called the Software Engineering Glossary. You’ll find definitions of various software engineering terms dispersed throughout the pages of each issue.

Richard Thayer, with his long and distinguished background in the software engineering community, has graciously agreed to edit this department. (See inside the back cover of this issue.) Thayer is a fellow of the IEEE, a member of the IEEE Computer Society Golden Core, a Certified Software Development Professional, and a member of the IEEE Software Engineering Standards Committee. He is a principal author of the Concept of Operations (ConOps) Document standard (IEEE Std 1362-1998) and a principal author of the Software Project Management Plans standard (IEEE Std 1058-1998).

I hope you will find these definitions as useful as I do.

EDITOR IN CHIEF:

**Warren Harrison**  
10662 Los Vaqueros Circle  
Los Alamitos, CA 90720-1314  
warren.harrison@computer.org

EDITOR IN CHIEF EMERITUS:  
Steve McConnell, Construx Software

**ASSOCIATE EDITORS IN CHIEF**

**Education and Training:** Don Bagert, Rose-Hulman Inst. of Technology; don.bagert@rose-hulman.edu

**Design:** Maarten Boasson, Quaerendo Invenietis  
boasson@quaerendo.com

**Requirements:** Christof Ebert, Alcatel  
christof.ebert@alcatel.com

**Management:** Ann Miller, University of Missouri, Rolla  
millera@ece.umar.edu

**Quality:** Jeffrey Voas, Cigital  
voas@cigital.com

**Experience Reports:** Wolfgang Strigel,  
Software Productivity Center; strigel@spc.ca

**EDITORIAL BOARD**

- Nancy Eickelmann, Motorola Labs
- Richard Fairley, Oregon Graduate Institute
- Martin Fowler, ThoughtWorks
- Robert Glass, Computing Trends
- Jane Hayes, University of Kentucky
- Andy Hunt, Pragmatic Programmers
- Warren Keuffel, independent consultant
- Brian Lawrence, Coyote Valley Software
- Karen Mackey, Cisco Systems
- Deependra Moitra, Infosys Technologies, India
- Don Reifer, Reifer Consultants
- Suzanne Robertson, Atlantic Systems Guild
- Richard H. Thayer, Calif. State Univ. Sacramento
- Dave Thomas, Pragmatic Programmers

**INDUSTRY ADVISORY BOARD**

- Robert Cochran, Catalyst Software (chair)
- Annie Kuntzmann-Combelles, Q-Labs
- Enrique Draier, MAPA LatinAmerica
- David Hsiao, Cisco Systems
- Takaya Ishida, Mitsubishi Electric Corp.
- Dehua Ju, ASTI Shanghai
- Donna Kaspersen, Science Applications International
- Pavle Knaflic, Hermes SoftLab
- Wojtek Kozaczynski, Microsoft
- Tomoo Matsubara, Matsubara Consulting
- Masao Matsumoto, Univ. of Tsukuba
- Dorothy McKinney, Lockheed Martin Space Systems
- Stephen Mellor, Project Technology
- Susan Mickel, AgileTV
- Dave Moore, Vulcan Northwest
- Melissa Murphy, Sandia National Laboratories
- Kiyoh Nakamura, Fujitsu
- Grant Rule, Software Measurement Services
- Girish Seshagiri, Advanced Information Services
- Chandra Shekaran, Microsoft
- Martyn Thomas, Praxis
- Rob Thomsett, The Thomsett Company
- John Vu, The Boeing Company
- Simon Wright, Integrated Chipware
- Tsuneo Yamaura, Hitachi Software Engineering

**MAGAZINE OPERATIONS COMMITTEE**

- Jean Bacon (chair), Thomas J. Bergin, Pradip Bose,
- Doris L. Carver, George Cybenko, John C. Dill,
- Frank E. Ferrante, Robert E. Filman,
- Forouzan Golshani, David A. Grier
- Rajesh Gupta, Warren Harrison,
- Mahadev Satyanarayanan, Nigel Shadbolt,
- Francis Sullivan

**PUBLICATIONS BOARD**

- Rangachar Kasturi (chair), Jean Bacon,
- Laxmi Bhuyan, Mark Christensen,
- Thomas Keefe, Deependra Moitra,
- Steven L. Tanimoto, Anand Tripathi