

SOFTWARE ENGINEERING *notes*

TABLE OF CONTENTS

LETTERS

- 1** From the Chair: If I Only Had a Brain – *Will Tracz*
- 2** From the Editor – *Michael Wing*
- 3** Top 10 Downloaded Articles
- 4** Top 10 Most Cited Articles
- 5** Bill Riddle, Past ACM SIGSOFT Chair, Dies – *Jack Wileden*
- 5** CAPS Award for ASE 2014 – *Breno Miranda*
- 6** CAPS ASE Experience – *Tony Ohmann*
- 6** New Guide to Computer Science Degrees – *Dan Schuessler*

COLUMNS

- 6** Passages – *Alex Groce*
- 8** Surfing the Net for Software Engineering Notes – *Mark Doernhoefer*
- 17** Risks to the Public – *Peter G. Neumann and contributors*

REPORTS & PAPERS

- 23** The list of Reports is found on page 3
- 28** The list of Papers is found on page 3

WORKSHOP & CONFERENCES

- 29** Calendar of Events – *Daniela Castelluccia*
- 32** ICSE 2015
- 38** FSE 2014



SOFTWARE ENGINEERING NOTES

www.sigsoft.org/SEN/

SIGSOFT EXECUTIVE COMMITTEE

CHAIR:

Will Tracz, Owego, NY 13827, <will.tracz@acm.org>

VICE CHAIR:

Laura K. Dillon, Dept. of CS&E, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824, +1-517-353-4387 (work), <ldillon@msu.edu>

SECRETARY/TREASURER:

Gail C. Murphy, Department of Computer Science, University of British Columbia, 2366 Main Mall, Vancouver BC Canada V6T 1Z4, +1-604-822-5169, <murphy@cs.ubc.ca>

AT-LARGE MEMBERS:

Frances Paulisch, Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, D-80200 Munich, Germany, <frances.paulisch@siemens.com>

Frank Tip, Samsung Research America
1732 N. First Street, 6th Floor
San Jose, CA 95112
<ftip@samsung.com>

Willem Visser, Stellenbosch University, South Africa, <http://www.cs.sun.ac.za/~wvisser/>

MOST RECENT PAST CHAIR:

David S. Rosenblum, School of Computing, National University of Singapore, Computing 1, 13 Computing Drive, Singapore 117417, Republic of Singapore, +65 6516 6230 (work), <david@comp.us.edu.sg>

NEWSLETTER EDITOR:

Michael Wing, <wingacm@swcp.com>

INTERNATIONAL LIAISON:

Shing-Chi Cheung, Dept. of Computer Science and Engineering, The Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong, +852 2358 7016, <sc@cs.ust.hk>

HISTORY LIAISON:

Tao Xie, Department of Computer Science, University of Illinois at Urbana-Champaign, 4237 Siebel Ctr., 201 N. Goodwin Ave., Urbana, IL 61801, +1 217.244.5931 <taoxie@illinois.edu>

PRACTITIONER LIAISON:

Mark Grechanik, Computer Science Department, University of Illinois at Chicago, <DRMark@uci.edu>

CAP COORDINATOR:

Wei Le, Iowa State University, <weile@iastate.edu>

SEN PRODUCTION EDITOR:

Ajay Jain, Adobe Systems, <ajjain@adobe.com>

INFORMATION DIRECTOR:

Greg Cooper, <GC1@acm.org>

Software Engineering Notes is an informal publication of the ACM Special Interest Group on Software Engineering (SIGSOFT) concerned with the cost-effective, timely development and maintenance of high-quality software. Relevant topics include requirements, specification, design and implementation methods, software maintenance, reuse, and re-engineering, quality assurance, measurement and evaluation, software processes, automated tools, and practical experience, etc. SIGSOFT seeks to address research and development issues in these areas and to provide a common ground for both, through sponsorship of conferences, symposia and workshops, the publication of SEN, and the dissemination of information via the SIGSOFT member email distribution list.

SIGSOFT sponsors an annual conference, Foundations of Software Engineering (FSE), in the Fall and co-sponsors the International Conference of Software Engineering (ICSE) in the Spring, in addition to sponsoring several workshops and symposia each year on specialized and timely topics.

Material for publication in SEN should be submitted to the editor electronically as a .txt, .doc, or .tex file. Paper and reports should use the LaTeX or Word format available at <http://www.acm.org/sigsoft/SEN/subpub.html>. CFPs and Advance Programs for SIGSOFT sponsored or in cooperation events may be submitted as .pdf files with all fonts embedded. Because only the abstracts of papers and reports will appear in the print issue of SEN, authors should review their abstract to make sure it reflects the key points addressed in the document, which will appear in full in the ACM Digital Library.

Submissions in the form of letters, abstracts, book reviews, specialized bibliographies, notices, news reports, etc. are all welcome. All contributions are considered personal rather than organizational; technical contributions are considered as unrefereed working papers. Authors should note that by submitting an article for distribution in SEN they grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library and in any Digital Library related services
- to allow users to make a personal copy of the article for non-commercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will make every effort to refer requests for commercial use directly to you.

Additional copies of SEN will be provided to authors only if requests are received before the issue is published.

Notice to Past Authors of ACM-Published Articles: ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that has been previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

Members wishing to participate in SIGSOFT activities may contact the Chair or Vice Chair. To join SIGSOFT, contact ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701, 1-212-869-7440. Annual dues are \$25 for ACM members and associates and \$12 for student members. Institutional or library subscriptions to SEN are \$35 per year. Back issues are available from ACM Order Dept., P.O. Box 12114, Church Street Station, New York, NY 10257. For other SIGSOFT related business, contact Fran Spinola, SIG Services, ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701 (1-212-626-0603, <spinola@hq.acm.org>).

Software Engineering Notes (ISSN 0163-5948) is published six times a year in January, March, May, July, September, and November by ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodical postage paid at New York, NY 10121-0701 and additional mailing offices. POSTMASTER: Send address changes to Software Engineering Notes, ACM, 2 Penn Plaza, Suite 701, New York, NY 10121-0701.

Letter from the Chair If I Only Had a Brain¹

Will Tracz
will@tracz.org

*I could wile away the hours - Conferrin' with the flowers
Consultin' with the rain. And my head I'd be scratchin'
While my thoughts were busy hatchin' - If I only had a brain*

If you happened to read the September issue of SEN's Letter from the Chair, your head may be scratching wondering if I am a Wizard of Oz fan. While the answer to that question is yes, I believe it is only a coincidence, as I have been consulting with colleagues on tons of SIGSOFT matters that made me question if it was necessary – if I had a brain.

Conferring/Conferencing FSE/ICSE

For those of you not having the “pleasure/opportunity” of being part of a conference organizing committee and witnessing first-hand the significant logistical effort required to plan a conference, it may not come as a surprise to find out that it requires a lot of work. Easily there are thousands of emails that are exchanged; countless hours spent pouring over minute details necessary to have a successful conference². I would like to thank, in advance, Shing-Chi Cheung from the Hong Kong University of Science and Technology for his tireless efforts in organizing this year's FSE (<http://fse22.gatech.edu/>). Needless to say, the venue required exponentially more coordination with service providers than most FSE's. I will spare you details of the progress Laurie Dillon at Michigan State University, the general chair of ICSE 2016 in Austin, has made and encourage you to go to the website³ (<http://2016.icse.cs.txstate.edu/>) and subscribe to the Facebook and Twitter feeds to keep abreast of progress. This said, I would encourage you to start making your plans for ICSE 2015 in Florence/Firenze, Italy (<http://2015.icse-conferences.org/>) – Antonia Bertolino has been busy working out the logistics for a technically and historically rewarding ICSE. Finally, my hat is off to the Tom Zimmermann at Microsoft, the general chair of FSE 2016, which will be held in Seattle, Washington, for nailing down the conference hotel⁴.

Hatching New Ideas

We are still working out the details for the new **SIGSOFT Webinar Series**. Speakers are being selected from the ICSE 2014 Foundations of Software Engineering (FOSE) talks. (<http://2014.icse-conferences.org/fose>) featuring topics such as

Certiably Safe Software-dependent Systems, Engineering, Big Data, Software Engineering for Mobility, and Software Testing.

Another idea that is being hatched is the establishment of an ACM Policy for Retired Members. While this is still a work in progress, it would appear that recommendation will be to offer membership and conference registration rates for retired members that are the same as student members.

I'd unravel any riddle, for any individ'le, in trouble or in pain

What will Software Engineering be like in 10 years?

This is the riddle I have been burning brain cells on, wondering about the vector and velocity Software Engineering research and practice is taking. Virtualization, big data, GPUs, scale, open source, IDEs, theorem provers, patterns, etc. – what, where and when is the next disruptive technology/game changer? What research topics have reached the point of diminishing returns? What old research topics are now poised for advancement given the progress in previously lacking technologies? Finally, what software engineering lessons learned are being re-discovered, re-invented, or renamed by application domains searching for ways to leverage software in advancing their technology?

Oh, I would tell you why, the ocean's near the shore

Awards? The good, bad and the ugly

I could tell you why, we don't get more submissions for our SIGSOFT awards (<http://www.sigsoft.org/awards>). Yes, it is time for SIGSOFT to find worthy candidates for consideration. Try as we may, there always seems to be a lack of submissions. So if anyone has questions, don't hesitate to contact me or our awards chair, Frank Tip (<http://www.sigsoft.org/about/execComm.htm>).

*I could think of things I never think before
And then I'd sit and think some more*

What I think about thinking about often falls under the category of not knowing what I don't know. I don't know if there are any Software Engineering areas that SIGSOFT needs to be supporting in the form of a co-located conference or symposium – or a new standalone conference? Maybe this would start out as a track or a panel in an existing conference? Should SIGSOFT be exploring possible joint conferences with other SIGS (e.g., SIGPLAN, SIGCHI, SIGCSE, etc.)?

I would not be just a nuffin'- My head all full of stuffin'

My heart all full of pain

I would dance and be merry - Life would be a ding-a-derry

If I only had a brain

Woodrow Wilson once said “I not only use all the brains that I have, but all that I can borrow.” I am a firm believer that you, the members of SIGSOFT, are an untapped resource and together we can identify, explore, refine, and implement new ways to improve our profession and the impact it has on our world. “My dull brain was wrought with things forgotten.” — William Shakespeare. Help me out! If you have any new ideas, or old ones whose day has come – please fire at...

Will

¹ Music by Harold Arlen and lyrics by E.Y. Harburg – don't ask me why we are on a Wizard of Oz kick.

² I realize this may sound discouraging to future OC volunteers, but it really is a rewarding opportunity to contribute to the software engineering community.

³ You have to check out the website if you are in the mood – no bull (sorry, I couldn't pass up the puns).

⁴ I should mention ESEC/FSE 2015 will be held Bergamo, Italy. <http://esec-fse15.dei.polimi.it/> - Yes, two chances to enjoy Italy next year! General Chair Elisabetta Di Nitto's efforts in the historical medieval city is making this a conference you won't want to miss.

Letter from the Editor

Michael Wing

editors_sigsoft@acm.org

Welcome to the November 2014 issue of SEN.

This month, Mark writes about autonomous systems, Alex writes about the book *The Pragmatic Programmer*, and Peter writes about - wait for it - computing risks. We have two CAPS reports, two workshop reports (FLOSS and the General Theory of Software Engineering), and seven papers. Also we have Daniela's calendar.

Also, "big congratulations to Mary Shaw for National Medal of Technology and Innovation, the USA's highest honor for achievement in the field of technology, innovation and invention!!!"

I have now been the editor of SEN for 2 years, which means that I have personally reviewed about 1000 drafts of maybe 400 different papers, about 10 per week. If you will indulge me, I want to share some 12 observations about researching and publishing.

One: Research takes a long time and a lot of work.

The single biggest problem in the papers I review is that they are unfinished. In most reviews, I ask authors to work on various language issues, because they need a few hints, but my real goal is to have them put more work into their papers. Writing a research paper takes longer than anyone wants. I find that when I get sick and tired of my own papers and they seem nearly complete, then I am maybe 1/3 of the way done. I read through this letter around 100 times over 2 weeks, got two colleagues to review it, and discussed the content with a number of other people, and this is just a short letter. There are no shortcuts to good work, and it never gets easy.

Two: Conflict is the main source of innovation.

Research is about innovation, and conflict exposes situations where new ideas are needed. Finding an issue where two researchers disagree exposes an opportunity to compare or unify their positions, which is pretty rare. Finding an issue where practice contradicts the prevailing theory exposes an opportunity for better theory, which happens a lot. The Waterfall was clear and simple, but nobody could live up to it. Eventually, people concluded, "this just isn't working and there must be a more realistic process." Note that Agile doesn't do requirements or designs or estimates any better, it just lets you change your mind.

Three: Consulting is a great place to find conflicts.

Companies only hire consultants when they are desperate. You see people at their best: hard-working and cooperative. You see people at their worst: panicked, floundering, deluded, and demanding. Consultants who cannot spot the chasms between theory and practice in these situations have no business doing research. Any workplace that does real development will expose similar chasms. Find any place where good people consistently struggle and you will find an important research topic.

Four: Most innovation is accidental.

If a topic is cool or inspired by a famous researcher or result, then probably many other researchers are already working on it and progress will be slow. I always wanted to do programming languages, but I don't seem to have the knack for it. Most of the research I have worked on was about process, even though I never wanted to do that. I like to whine about my bosses, so I do process (see "Burt Doesn't Manage"). You will have to work and work and work and then stumble over something that you can make your own. It won't be what you originally intended.

Five: A survey of the literature is not innovation.

A literature survey will tell you what is known and what is agreed upon, or not, but it cannot tell what is missing or wrong. Researchers need to find out what is known, so that they can relate their work to the literature, but researchers also need to find out what is unknown, so that they can innovate. As a rule, comparisons to related work are the weakest parts of papers. They are seldom systematic or fair. Reducing years of someone else's research to a single sentence or paragraph does everyone an injustice. Historical analysis is the only way to appreciate someone else's work.

Six: Implementation is not research.

Explaining the functions or classes in your app is boring.

Seven: Research proposals are not research.

They are proposals. Nobody cares what you want to do. One paragraph about future work is enough.

Eight: Standards are about the past.

It takes ten years for a good idea to become widely-accepted and well-understood, before it can be put into a standard. This means that most standards are historical reenactments. Doing contemporary software engineering means doing what is appropriate for today.

It also means that research papers should never agree with a standard. The only reason to mention a standard in a research paper is to argue.

Nine: Presume that everyone is acting in good faith.

Practitioners are not unethical, just because they don't follow your theory or standard. Researchers are not clueless, just because they don't describe your situation. In my experience, everyone is busy and worried about their own problems. Authors only go wrong when they presume that anyone else should do something their way, which is politics rather than innovation. To quote Charles McCabe, "Nobody has to do anything."

Ten: Everything in a paper is the author's opinion.

The author chooses what content he or she considers important and then writes corresponding sentences. Using passive voice to make all your opinions seem self-evident is a common tactic in technical writing that does not work. Take responsibility for your ideas.

Write "this is my opinion." Mostly, nobody cares what the facts are, because "any clod can have the facts," to quote Charles McCabe again. Tell me what you really believe is important and why it makes sense. I believe that technical essays are more honest and less contrived than any other form.

Eleven: Real accomplishment in research is developing and expressing a new idea.

Gaming the system is not hard. Apparently, people love to play games and to beat the system. According to anecdotal evidence provided by our associate editors, 1/3 of the papers SEN publishes are plagiarized, and this is after using serious plagiarism detection tools. Over 1/4 of my time is spent dealing with plagiarism. We simply do not know where content comes from. Given the scope of the problem, we don't expect it to go away. Sites like essaytyper.com suggest that it will only get worse. But, beating SEN is not nearly as much fun as contributing to software engineering.

Twelve: SEN is an informal publication, intended for works in progress, incomplete papers, surveys, proposals, and anything else you find interesting.

If you write such a paper, just make it clear in the title: "A Preliminary Report of ...," "A Proposal for ...," "A Survey of ...". Even so, all papers should say something new that the SE community might care about. SEN is a forum to practice, experiment, and learn. We do not require perfection; we only require a good-faith effort and readable text. We want our authors move on and to write even more ambitious papers for refereed journals. I hope the reason you want to publish in SEN is to share your interesting new idea or experience. If that is the reason for your paper, we will be happy to work with you and you will get published.

As stated above, over the past 2 years, I have edited about 10 papers per week, and this workload usually takes about 2 hours per day. However, starting a month before ICSE 2014 in Hyderabad, the number of submissions from India more than doubled. This is very good news: it means researchers from India are trying harder and doing more. Their papers are better than ever. Because of holidays and vacation over the summer, I was able to cope. But in the shuffle, I also made several big mistakes, and I failed to get at least three accepted papers into issues.

In the last month, the number of submissions has fallen back a bit, but it is still an unsustainable amount of work for me and, I have fallen a month behind. I feel bad, but there is not much I can do. For the next few months, response times to authors will be much worse than anyone wants. SEN does not have the editing resources to review all the papers that are being submitted.

Things need to change and they will. It is hard to say exactly what will happen, but we do know a few things. Will Tracz and SigSoft have been developing a new author submission website that will make it possible to crowd-source the reviewing, but it is not ready. We may have to stop doing complete reviews or we may need to find an editor who doesn't need a real job. If anyone has any thoughts, please pass them along.

For now, please enjoy the issue.

Michael Wing

Mike

	SEN Volume 39 Number 6 Report Title/Author	Pg
1	How to Develop a General Theory of Software Engineering: Report on the GTSE 2014 Workshop Paul Ralph, Iakov Exman, Pan-Wei Ng, Pontus Johnson, Michael Goedicke, Alper Tolga Kocataş and Kate Liu Yan	23
2	FLOSS Research Track at the 10th International Symposium on Open Collaboration (OpenSym 2014) Gregorio Robles, Jesús M. González-Barahona and Dirk Riehle	26

Table 1: SEN Volume 39 Number 6 Reports

	SEN Volume 39 Number 6 Paper Title/Author	Pg
1	Mitigation of SQL Injection Attacks using Threat Modeling Navdeep Kaur and Parminder Kaur	28
2	A Distributed Load-balancing Scheme Based on a Complex Network Model of Cloud Servers Narander Kumar, Shalini Agarwal, Taskeen Zaidi and Vipin Saxena	28
3	A Novel Approach to Component-Based Software Testing Lata Nautiyal, Dr. Neena Gupta and Dr. Sushil Chandra Dimri	28
4	Measurement of the Reliability of a Component-Based Development using a Path-Based Approach Lata Nautiyal, Dr. Neena Gupta and Dr. Sushil Chandra Dimri	28
5	Measuring Change-Readiness of Classes by Mining Change-History Anshu Parashar and Jitender Kumar Chhabra	29
6	Identifying Interactions for Combinatorial Testing using Data Flow Techniques Sangeeta Sabharwal and Manuj Aggarwal	29
7	Performance Estimation of Static Step Topology Across Distributed Networks through Simulation Tool Taskeen Zaidi and Vipin Saxena	29

Table 2: SEN Volume 39 Number 6 Papers

Top 10 Downloaded Articles

(past 12 weeks)

<http://dl.acm.org/sig.cfm?id=SP950>

1. **Lightweight Automated Detection of Unsafe Information Leakage via Exceptions** – 2014, Benwen Zhang, James Clause. Downloaded 149 (new) times.
2. **Software Engineering Issues for Mobile Application Development** – 2010, Anthony I. Wasserman. Downloaded 122 (new) times.
3. **Requirements Engineering in the Year 00: a Research Perspective** – 2000, Axel van Lamsweerde. Downloaded 118 (144) times
4. **UML in Action: a Two-layered Interpretation for Testing** – 2011, Bernhard K. Aichernig, Harald Brandl, Elisabeth Jöbstl, Willibald Krenn. Downloaded 100 (99) times
5. **Requirements Engineering: a Roadmap** – 2000, Bashar Nuseibeh, Steve Easterbrook. Downloaded 97 (187) times
6. **Software Development Lifecycle Models** – 2010, Nayan B. Ruparelia. Downloaded 86 (new) times
7. **CoREBench: Studying Complexity of Regression Errors** – 2014, Marcel Böhme, Abhik Roychoudhury. Downloaded 84 (new) times

8. **A Requirements-based Approach for the Design of Adaptive Systems** – 2012, Vítor E. Silva Souza. Downloaded 79 (89) times
9. **Core Java Volume I: Fundamentals, 9th edition** by Cay S. Horstmann and Gary Cornell – 2013, Igor Gvero. Downloaded 78 (new) times
10. **View of 20th and 21st Century Software Engineering** – 2006, Barry Boehm. Downloaded 72 (104) times

Top 10 Most Cited Articles

1. **Foundations for the Study of Software Architecture** – 1992, Dewayne E. Perry, Alexander L. Wolf. Cited 485 (485) times
2. **Bandera: Extracting Finite-State Models from Java Source Code** – 2000, James C. Corbett, Matthew B. Dwyer, John Hatcliff, Shawn Laubach, Corina S. Păsăreanu, Robby, and Hongjun Zheng. Cited 389 (388) times
3. **N Degrees of Separation: Multi-dimensional Separation of Concerns** – 1999, Peri Tarr, Harold Ossher, William Harrison, Stanley M. Sutton, Jr. Cited 382 (382) times
4. **CUTE: a Concolic Unit Testing Engine for C** – 2005, Koushik Sen, Darko Marinov, and Gul Agha. Cited 350 (348) times
5. **Dynamo: a Transparent Dynamic Optimization System** – 2000, Vasanth Bala, Evelyn Duesterwald, Sanjeev Banerjia. Cited 345 (343) times
6. **Interface Automata** – 2001, Luca de Alfaro, Thomas A. Henzinger. Cited 342 (341) times
7. **Patterns in Property Specifications for Finite-State Verification** – 1999, Matthew B. Dwyer, George S. Avrunin, James C. Corbett. Cited 299 (299) times
8. **Program Slicing** – 1981, Mark Weiser. Cited 258 (257) times
9. **Automatically Validating Temporal Safety Properties of Interfaces** – 2001, Thomas Ball, Sriram K. Rajamani. Cited 248 (248) times
10. **Software Processes are Software Too** – 1987, L. Osterweil. Cited 232 (232) times
11. **Pick-and-Drop: a Direct Manipulation Technique for Multiple Computer Environments** – 1997, Jun Rekimoto. Cited 232 (232) times

SEN Reviewers Wanted

The following table lists books that are available free to SIGSOFT members for review. Go to <http://www.sigsoft.org/SEN/bookreview.html> for an up-to-date list, as the list changes frequently, with new reviewer requests, and new books being sent from publishers.

Processes & Engineering	
<i>A Practical Approach to Large-Scale Agile Development: How HP Transformed LaserJet Future Smart Firmware</i>	Gary Gruver, Mike Young and Pat Fulghum
<i>Agile and Lean Service-Oriented Development: Foundations, Theory, and Practice</i>	Xiaofeng Wang, Nour Ali, Isidro Ramos, and Richard Vidgen
Design, Architecture, and Patterns	

<i>Architecture and Patterns for IT Service Management, Resource Planning, and Governance, 2nd Edition: Making Shoes for the Cobbler's Children</i>	Charles T. Betz
<i>Building Enterprise Systems with ODP: An Introduction to Open Distributed Processing</i>	Peter F. Linington, Zoran Milosevic, Akira Tanaka and Antonio Vallecillo
<i>Design Research Through Practice: From the Lab, Field, and Showroom</i>	Ilpo Koskinen, John Zimmerman, Thomas Binder, Johan Redstrom, & Stephan Wensveen
<i>Guerrilla UX Research Methods: Thrifty, Fast, and Effective User Experience Research Techniques</i>	Russ Unger & Todd Zaki Warfel
<i>Information Systems Transformation Architecture-Driven Modernization Case Studies</i>	William M. Ulrich and Philip H. Newcomb
<i>Software Architecture in Practice, 3rd edition</i>	Len Bass, Paul Clements and Rick Kazman
<i>User Experience in the Age of Sustainability: A Practitioner's Blueprint</i>	Kem-Laurin Kramer
Data	
<i>Next Generation Search Engines: Advanced Models for Information Retrieval</i>	Christophe Jouis, Ismail Biskri, Jean-Gabriel Ganascia and Magali Roux
<i>Using Open Source Platforms for Business Intelligence: Avoid Pitfalls and Maximize ROI</i>	Lyndsay Wise
Programming, Tools, & Web Development	
<i>Deploying Rails</i>	Tom Copeland and Anthony Burns
<i>Dreamweaver CS6: The Missing Manual</i>	David Sawyer McFarland
<i>Drupal 7 Explained: Your Step-by-Step Guide</i>	Stephen Burge
<i>Introduction to Combinatorial Testing</i>	D. Richard Kuhn, Raghu N. Kacker, and Yu Lei
<i>Java Reflection LiveLessons (video training)</i>	Paul Anderson
<i>Letting Go of the Words - Writing Web Content that Works 2nd Edition</i>	Janice (Ginny) Redish
<i>Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks, 2nd edition</i>	David Josephsen
<i>Practical Object-Oriented Design in Ruby: An Agile Primer</i>	Sandi Metz
<i>Rails Recipes: Rails 3 Edition</i>	Chad Fowler
<i>Sams Teach Yourself Node.js in 24 Hours: 24 Proven One-Hour Lessons</i>	George Ornbo
<i>Scala Fundamentals LiveLessons (Video Training)</i>	Presented by Dan Rosen
<i>The CSS3 Anthology: Take your sites to New Heights</i>	Rachel Andrew

<i>The Official Joomla! Book, 2nd Ed.</i>	Jennifer Marriott and Elin Waring
<i>The Rails View: Create a Beautiful and Maintainable User Experience</i>	Bruce Williams and John Athayde
Operating Systems	
<i>Switching to the Mac: The Missing Manual, Lion Edition</i>	David Pogue
<i>Windows 8.1: The Missing Manual</i>	David Pogue
Mobility	
<i>iPad: The Missing Manual, 4th Edition</i>	J.D. Biersdorfer
Cloud Computing, SOA, and SaaS	
<i>Is Your Company Ready for Cloud?: Choosing the Best Cloud Adoption Strategy for Your Business</i>	Pamela Isom with Kerrie Holley
<i>Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments</i>	Anca Daniela Ionita, Marin Litoiu, and Grace Lewis
Management, Industry Affairs, and Enterprises	
<i>Cases on ICT Utilization, Practice and Solutions: Tools for Managing Day-to-Day Issues</i>	edited by Mubarak S. Al-Mutairi et al.
<i>Competition, Strategy, and Modern Enterprise Information Systems</i>	Madjid Tavana
<i>Content Strategy at Work: Real-world stories to strengthen every interactive project</i>	Margot Bloomstein
<i>Dynamic Models for Knowledge-Driven Organizations</i>	Murray E. Jennex
<i>Enterprise Software Delivery: Bringing Agility and Efficiency to the Global Software Supply Chain</i>	Alan Brown
<i>Intersection: How Enterprise Design Bridges the Gap between Business, Technology, and People</i>	Milan Guenther
<i>Knowledge Driven Service Innovation and Management: IT Strategies for Business Alignment and Value Creation</i>	Eng K. Chew and Petter Gottschalk
<i>Mobile Strategy: How Your Company Can Win by Embracing Mobile Technologies</i>	Dirk Nicol

Table 3: SEN books needing to be reviewed

Bill Riddle, Past ACM SIGSOFT Chair, Dies

Jack Wileden
jack@cs.umass.edu

William E. Riddle, an early leader of the software engineering community and an active member for over 40 years, died on September 26, 2014 in Santa Fe, NM due to complications from a recent fall. He was 72 years old. Bill earned his bachelors and masters degrees from Cornell and his PhD in Computer Science from Stanford in 1972. Early in his career he served on the

faculties of the University of Michigan and the University of Colorado and was a visiting research fellow at the University of Newcastle upon Tyne. Late in his career he held visiting scientist and senior scientist appointments at the Fraunhofer Institute for Experimental Software Engineering. The majority of his professional life, however, was devoted to providing software engineering leadership in a variety of roles at a variety of organizations: Software Design and Engineering Manager at Cray Laboratories, Chief Technical Officer and Executive Vice President at the Software Productivity Consortium, Senior Member of Technical Staff at the Software Engineering Institute, Solution Architect at Borland Software (TeraQuest Metrics), President of Software Design & Analysis Inc., Director of the Rocky Mountain Institute of Software Engineering, and Senior Solution Architect at Solution Deployment Affiliates.

Bill was instrumental in the early development of the SIGSOFT community, serving as vice-chair of the SIGSOFT Executive Committee (1979-1981), as the third chair of ACM SIGSOFT (1981-85), as general chair of the first SIGSOFT Software Engineering Symposium (1981) and as general chair of the 1987 International Conference on Software Engineering. His many contributions were recognized by his election to IEEE Fellow (1991), by a 1996 ICSE Most Influential Paper Award (for *Sam Redwine Jr., William Riddle: Software Technology Maturation, ICSE-8, 1985*) and by an ACM SIGSOFT Distinguished Service Award (1999). An obituary appeared in the Boulder Daily Camera:

<http://www.legacy.com/obituaries/dailycamera/obituary.aspx?n=william-riddle&pid=172666735>

CAPS Award for ASE 2014

Breno Miranda

University of Pisa

brenoafmiranda@gmail.com

The purpose of this report is to share my experience in attending the Automated Software Engineering conference (and its co-located Doctoral Symposium) held in Västerås, Sweden from 15th to 19th of September 2014.

It was my first time in a doctoral symposium and I think it was a very enriching experience. I received very good feedback about what I have done so far and good directions on how to proceed in the next steps of my research. I am currently working on the topic of Coverage Testing and the most important feedback I received was related to the scope of my research. One of the committee members mentioned that I may possibly be working on a very big scope and that I should consider revising it with my advisor in order to tailor it to avoid having problems to complete everything on time.

This year's edition of the doctoral symposium included 10 students working on many different topics encompassed by software engineering and all the students have been assigned to some tasks prior to the doctoral symposium: first, we were asked to read one given paper using the "reviewer's mindset" (from another student attending the doctoral symposium) and elaborate at least one good question related to his/her research; second, we were asked to read a second paper from another student and act as the "session chair" for that work (introduce the students' work and lead the Q&A session). All these tasks were very important

to give us an idea about what to expect if we decide to continue in the academia after finishing our PhD.

The doctoral symposium also included two panels (“Finding a Job with your PhD” and “Finishing your Thesis”) and two group discussions (“Student Research Direction” and “Publishing and what makes a Good Paper”). I really appreciated the idea of having something more than just giving feedback about the students’ research. The panels and the group discussions provided many good tips not only for the time we are pursuing our PhD, but also for what can come next.

The participation in the main conference was also very good. The conference’s program was very attractive. Many talks (especially the ones in the symbolic execution and testing tracks) were related to my research topic and it was a good opportunity to make questions and have ideas about other directions that I could follow in the future.

Finally, attending the conference was a great opportunity to be in touch with other researchers that I met in previous academic events and to meet new ones. Staying in touch with other researchers and planning future collaboration is definitely very important in the academic life.

I would like to reiterate my gratitude to the SIGSOFT-CAPS program and the SIGSOFT CAPS Coordinator, Wei Le, for supporting my participation at the Automated Software Engineering conference and its co-located Doctoral Symposium.

CAPS ASE Experience

Tony Ohmann

University of Massachusetts
ohmann@cs.umass.edu

Thanks in large part to the funding provided by a CAPS travel grant, I was able to attend and present my work at Automated Software Engineering (ASE) 2014 in Västerås, Sweden. I attended a tutorial on Tuesday, September 16 and the main ASE conference sessions Wednesday through Friday.

I attended a tutorial and a session on bug localization, a field with which I was not previously familiar. This was particularly impactful for me because it helped me realize that my current work in model inference might be an excellent complement to developments in the bug localization field. As a result of attending the conference, I am considering extending and combining my work with current advances in this field.

At ASE, I had the opportunity to meet a number of important figures in my current field and related fields. The one-on-one discussions I had with other researchers were invaluable in helping me understand better ways to present or discuss my work. I also discussed promising future research directions that I hadn't previously considered. Further, I learned of many new projects and upcoming techniques that other researchers are developing, which helped me broaden my understanding of the field and its future directions.

Already in the first session of the conference, I listened to a talk on a performance prediction and modeling technique that I recognized as being a potentially great complement to my model inference work. Through discussions with the primary author, an industry researcher working at IBM, I had the opportunity to explore ideas working towards collaboration. These discussions continued beyond that week and are ongoing over digital mediums, but this potential collaboration would never have been

possible had I not been able to travel to this conference, see the presentation, and start the discussion in person.

Finally, attending ASE 2014 granted me the opportunity to attend presentations on extremely important and current research in software engineering in an organic way. Unfortunately, few researchers have the time to truly and fully explore the literature of their broad computer science area, and this conference was extremely valuable in keeping me aware of the current advances in the software engineering field. I attended many excellent talks on innovations in sub-areas outside of my current research such as mining information about software, inferring and evaluating test suites, detecting optimal software configurations, and others. It is typically difficult to stay up to date on a diverse range of areas like these, and attending this conference has afforded me a firsthand survey of present work in many of them.

Attending ASE gave me insight into some current and future directions of software engineering research, opened the door to a potential collaboration, helped me consider new angles of my current research, and provided me with fresh ideas for my future work. It was an invaluable experience made possible by funding from the CAPS travel grant, for which I am very grateful.

New Guide to Computer Science Degrees

Dan Schuessler

Computer Science Online

dan@computerscienceonline.org

Computer Science Online recently published an in-depth guide to computer science degrees and careers. Fueled by knowledge and insight from some of today's leading computer science minds, this new resource breaks down the value of computer science degrees at each academic level, how to find the right program, and new specializations that lie on the cutting edge of an already high-tech field.

Seven computer science experts contributed to the guide, including professors from Carnegie Mellon, Cornell and Ohio State. Each expert discusses the current landscape of degrees in CS, as well as various offshoots driving career growth: cloud computing, robotics, machine learning, software development and more.

Computer science degree programs:

<http://www.computerscienceonline.org/degree-programs/>

Support computer science students:

With the new school year in full swing, our mission is to get this new guide in front as many current and future computer science students as possible.