

COMPUTING

edge

# SOFTWARE DEVELOPMENT

Also in this issue:

- > **Technology and the Arts:  
Educational Encounters of the Third Kind**
- > **The State of the Cloud**

MAY 2017

[www.computer.org](http://www.computer.org)

 **IEEE**

IEEE  computer society



## Focus on Your Job Search

**IEEE Computer Society Jobs** helps you easily find a new job in IT, software development, computer engineering, research, programming, architecture, cloud computing, consulting, databases, and many other computer-related areas.

**New feature:** Find jobs recommending or requiring the IEEE CS CSDA or CSDP certifications!

Visit [www.computer.org/jobs](http://www.computer.org/jobs) to search technical job openings, plus internships, from employers worldwide.

<http://www.computer.org/jobs>

IEEE  computer society | **JOBS**



The IEEE Computer Society is a partner in the AIP Career Network, a collection of online job sites for scientists, engineers, and computing professionals. Other partners include Physics Today, the American Association of Physicists in Medicine (AAPM), American Association of Physics Teachers (AAPT), American Physical Society (APS), AVS Science and Technology, and the Society of Physics Students (SPS) and Sigma Pi Sigma.



STAFF

**Editor**  
Lee Garber

**Contributing Staff**  
Christine Anthony, Brian Brannon, Lori Cameron, Cathy Martin,  
Chris Nelson, Meghan O'Dell, Dennis Taylor, Rebecca Torres,  
Bonnie Wylie

**Production & Design**  
Carmen Flores-Garvey, Monette Velasco, Jennie Zhu-Mai,  
Mark Bartosik

**Manager, Editorial Content**  
Carrie Clark

**Senior Manager, Editorial Services**  
Robin Baldwin

**Director, Products and Services**  
Evan Butterfield

**Senior Advertising Coordinator**  
Debbie Sims

**Circulation:** ComputingEdge (ISSN 2469-7087) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036.

**Postmaster:** Send address changes to ComputingEdge-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

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

**Reuse Rights and Reprint Permissions:** Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own Web servers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copy-editing, proofreading, and formatting added by IEEE. For more information, please go to: [http://www.ieee.org/publications\\_standards/publications/rights/paperversionpolicy.html](http://www.ieee.org/publications_standards/publications/rights/paperversionpolicy.html). Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). Copyright © 2017 IEEE. All rights reserved.

**Abstracting and Library Use:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

**Unsubscribe:** If you no longer wish to receive this ComputingEdge mailing, please email IEEE Computer Society Customer Service at [help@computer.org](mailto:help@computer.org) and type "unsubscribe ComputingEdge" in your subject line.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit [www.ieee.org/web/aboutus/whatis/policies/p9-26.html](http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html).

## IEEE Computer Society Magazine Editors in Chief

**Computer**

Sumi Helal, *University of Florida*

**IEEE Software**

Diomidis Spinellis, *Athens University of Economics and Business*

**IEEE Internet Computing**

M. Brian Blake, *University of Miami*

**IT Professional**

San Murugesan, *BRITE Professional Services*

**IEEE Security & Privacy**

Ahmad-Reza Sadeghi, *Technical University of Darmstadt*

**IEEE Micro**

Lieven Eeckhout, *Ghent University*

**IEEE Computer Graphics and Applications**

L. Miguel Encarnação, *ACT, Inc.*

**IEEE Pervasive Computing**

Maria Ebling, *IBM T.J. Watson Research Center*

**Computing in Science & Engineering**

Jim X. Chen, *George Mason University*

**IEEE Intelligent Systems**

V.S. Subrahmanian, *University of Maryland*

**IEEE MultiMedia**

Yong Rui, *Lenovo Research and Technology*

**IEEE Annals of the History of Computing**

Nathan Ensmenger, *Indiana University Bloomington*

**IEEE Cloud Computing**

Mazin Yousif, *T-Systems International*

MAY 2017 • VOLUME 3, NUMBER 5

# COMPUTING edge



14

Code Craft

26

The Best Software  
Development  
Teams Might Be  
Temporary

30

The Roles  
of Code in  
Computational  
Science



# 44

## Technology and the Arts: Educational Encounters of the Third Kind

### 8 Spotlight on Transactions: Persistent Memory

DANIEL J. SORIN

### 9 Editor's Note: The Ever-Changing World of Software Development

### 10 A License to Build (Software)

DAVID ALAN GRIER

### 11 Silver Bullet Talks with Jim Manico

GARY MCGRAW

### 14 Code Craft

GERARD J. HOLZMANN

### 18 Application Architecture for the Internet of Cities: Blueprints for Future Smart City Applications

JOHANNES M. SCHLEICHER, MICHAEL VOGLER, SCHAHRAM DUSTDAR, AND CHRISTIAN INZINGER

### 26 The Best Software Development Teams Might Be Temporary

RAFAEL PRIKLADNICKI, MARCELO GATTERMANN PERIN, SABRINA MARCZAK, AND ALESSANDRA COSTA SMOLENAARS DUTRA

### 30 The Roles of Code in Computational Science

KONRAD HINSEN

### 36 The Social Responsibility of Software Development

DIOMIDIS SPINELLIS

### 40 The Means of Production

DAVID ALAN GRIER

### 42 Before Torchi and Schwilgué, There Was White

DENIS ROEGEL

### 44 Technology and the Arts: Educational Encounters of the Third Kind

AISLING KELLIHER

### 48 The State of the Cloud

MAZIN YOUSIF

### Departments

- 4 Magazine Roundup
- 50 Computing Careers: Careers in Software Engineering
- 52 Career Opportunities

Subscribe to **ComputingEdge** for free at  
[www.computer.org/computingedge](http://www.computer.org/computingedge).



# Magazine Roundup

The IEEE Computer Society's lineup of 13 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to cloud migration and microchip design. Here are highlights from recent issues.

## *Computer*

New and innovative technologies enable instructional environments that help students overcome many traditional learning boundaries and constraints. As the classroom becomes more of an abstraction than a physical space, educators and learners are embracing various tech-powered teaching and learning paradigms

that will help students, a trend addressed in *Computer's* May 2017 special issue on **advances in learning technologies**.

## *IEEE Software*

Crowdsourcing is disrupting business models and work practices, and could have a profound impact on software development. *IEEE Software's* March/April 2017 special issue explores various ways that developers and managers could benefit by using **crowdsourcing for software engineering**.

## *IEEE Internet Computing*

With the dawn of the Internet of Things (IoT), small smart devices are becoming ubiquitous. However, they lack conventional interfaces such as

keyboards and mice, and thus can't authenticate and authorize users in familiar ways. Moreover, a one-time authentication at a session's start—as is used in conventional computing settings—won't work for the IoT because application scenarios are dynamic. Thus, users must be continuously authenticated and authorized. This is discussed in "**Continuous Authentication and Authorization for the Internet of Things**," from *IEEE Internet Computing's* March/April 2017 issue.

## *Computing in Science & Engineering*

The US National Cancer Institute and the US Department of Energy (DOE) will use DOE supercomputers to help fight cancer by building sophisticated models based on population, patient, and molecular data. During a three-year pilot project, four participating national laboratories will focus on three key cancer-research problems. This is examined in "**Cancer's Big Data Problem**," from *CiSE's* March/April 2017 issue.

### *IEEE Security & Privacy*

To fully benefit from today's cybersecurity technology, we must complement it with a **mandatory reporting system** that is comprehensive; inexpensive; confidential; and informative about the volume, pattern, and character of digital attacks, according to "Mutual Dependence Demands Mutual Sharing," from *IEEE S&P's* January/February 2017 issue.

### *IEEE Cloud Computing*

Big companies aren't using **platform-as-a-service (PaaS) technology** as much as initially predicted. Offerings vary widely by provider, which has led to major confusion among potential users, according to "PaaS Death Watch?" from *IEEE Cloud Computing's* January/February 2017 issue.

### *IEEE Computer Graphics and Applications*

Two key concerns for wearable devices are how their user interfaces should look in a post-smartphone world and whether they will support sophisticated mobile interactions and information management. The authors of "**Spatial Analytic Interfaces: Spatial User Interfaces for In Situ Visual Analytics**," from *CG&A's* March/April 2017 issue, explore an approach that promises to address these concerns.

### *IEEE Intelligent Systems*

Big data has enabled innovation and advancement in many areas and is

receiving growing attention from researchers and practitioners. However, it still faces many challenges. *IEEE Intelligent Systems' March/April 2017* special issue on **big data** examines these challenges.

### *IEEE MultiMedia*

Most contemporary Western performing-arts practices restrict **creative audience interaction**. The authors of "Open Symphony: Creative Participation for Audiences of Live Music Performances," from *IEEE MultiMedia's* January–March 2017 issue, discuss a system they developed to explore audience–performer interaction in live music presentations.

### *IEEE Annals of the History of Computing*

"Edge Cryptography and the Co-development of Computer Networks and Cybersecurity," from *IEEE Annals' October–December 2016* issue, looks at the **private line interface (PLI)**, a cryptographic cybersecurity device developed in the early 1970s and used on the Arpanet. It operated with minimal modification of the existing network infrastructure and sat at the edge between the network switches and the connected host computers. The PLI helped establish the historical relationship between cryptography and packet-switched computer networks.

### *IEEE Pervasive Computing*

The authors of "Unleashing the Power of Active Sonar," from *IEEE*

*Pervasive Computing's* January–March 2017 issue, explore various applications that can be enabled by **transforming mobile devices into active sonar systems**.

### *IT Professional*

Researchers and designers have begun developing **fifth-generation (5G) mobile networks**. "Technologies for 5G Networks: Challenges and Opportunities," from *IT Pro's* January/February 2017 issue, reviews potential 5G approaches and concludes that radical network paradigm changes will be necessary.

### *IEEE Micro*

The authors of "Low-Power Automatic Speech Recognition Through a Mobile GPU and a Viterbi Accelerator," from *IEEE Micro's* January/February 2017 issue, propose an **automatic speech-recognition system for low-power devices**. The system combines a mobile graphics processing unit (GPU) for running a deep neural network and a dedicated hardware accelerator for a Viterbi search that generates the sequence of words that a speaker most likely used.

### *Computing Now*

The Computing Now website ([computingnow.computer.org](http://computingnow.computer.org)) features **up-to-the-minute computing news** and blogs, along with articles ranging from peer-reviewed research to opinion pieces by industry leaders. 📍

# PREPARE TO CONNECT



The IEEE Computer Society is launching **INTERFACE**, a new communication tool to help members engage, collaborate and stay current on CS activities. Use **INTERFACE** to learn about member accomplishments and find out how your peers are changing the world with technology.

We're putting our professional section and student branch chapters in the spotlight, sharing their recent activities and giving leaders a window into how chapters around the globe meet member expectations. Plus, **INTERFACE** will keep you informed on CS activities so you never miss a meeting, career development opportunity or important industry update.

**Launching this spring. Watch your email for its debut.**

IEEE  computer society

# INTERFACE

## SkillChoice™ Complete

Now with expanded libraries and an upgraded platform!

Valued at  
**\$3,300!**



**OVER 20x** as many resources as before

# One membership. Unlimited knowledge.

Did you know IEEE Computer Society membership comes with access to a high-quality, interactive suite of professional development resources, available 24/7?

Powered by Skillssoft, the SkillChoice™ Complete library contains more than \$3,000 worth of industry-leading online courses, books, videos, mentoring tools and exam prep. Best of all, you get it for the one low price of your Preferred Plus, Training & Development, or Student membership package. There's something for everyone, from beginners to advanced IT professionals to business leaders and managers.

The IT industry is constantly evolving. Don't be left behind. Join the IEEE Computer Society today, and gain access to the tools you need to stay on top of the latest trends and standards.

Learn more at [www.computer.org/join](http://www.computer.org/join).





SPOTLIGHT ON TRANSACTIONS

# Persistent Memory

*This installment highlighting the work published in IEEE Computer Society journals comes from Computer Architecture Letters.*

**Daniel J. Sorin**, Duke University

**N**onvolatile memory (NVM) technologies—particularly storage class memory (SCM)—offer computer architects exciting new opportunities. Unlike more traditional storage such as hard disks or solid-state drives (SSDs), SCM uses a well-known interface that allows architects to read and write at byte-level granularity. SCM can serve in many roles: as a main memory replacement for DRAM, a new level in the memory hierarchy between the main memory (DRAM) and disk (either hard disk or SSD), or an independent memory region that operates in parallel with the DRAM and disk.

But along with persistent memory's opportunities come challenges. How does one synchronize accesses to volatile and nonvolatile memory? How does one provide atomic semantics for groups of accesses? And, paramount for architects, how does one do all of this efficiently and while placing minimal burden on the programmer?

Libei Pu and her colleagues explore such challenges in “Non-Intrusive Persistence with a Backend NVM Controller” (*Computer Architecture Letters*, vol. 15, no. 1, 2016, pp. 29–32). Through a remarkably simple example—a programmer wishes to write to three variables atomically (either all writes succeed or none)—the authors highlight the ordering and atomicity problems that can arise. What if some, but not all, of the writes have propagated to persistent memory when power fails? How does a core find the most recent copy of a piece of data in a complicated memory hierarchy that includes SCM?

Synthesizing the key innovations of various proposed SCM designs, Pu and her colleagues present an elegant solution for a system in which SCM is the main memory replacement. Programmers simply annotate the regions of code they wish to be atomic and their new NVM controller ensures that accesses obtain the most recently written values, even if those writes haven't propagated to the SCM. It isn't yet clear

whether an all-hardware or a hybrid hardware–firmware solution for the NVM controller is more appropriate; architects and systems designers will likely debate this and other design decisions in the near future.

**T**his is an exciting time for computer architects, with many new memory technologies emerging after years of being restricted to SRAM for caches, DRAM for memory, and hard disks for storage. One of these technologies' most promising aspects is nonvolatility. Pu and her colleagues' paper nicely illustrates some of the challenges—and possible solutions—for architects seeking to incorporate NVM into their systems. **□**

**DANIEL J. SORIN** is the W.H. Gardner Jr. Professor of Electrical and Computer Engineering at Duke University. Contact him at [sorin@ece.duke.edu](mailto:sorin@ece.duke.edu).

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

# The Ever-Changing World of Software Development

Companies and consumers regularly demand new applications with more functionality and thus greater complexity. In the process of meeting this demand, developers regularly look for better and faster ways to create products and avoid bugs, in some cases radically changing the way they work.

These are among the reasons that software development is such a dynamic field. This *ComputingEdge* issue examines current software-development trends.

*Computer's* "A License to Build (Software)" discusses the Malaysian government's proposal to license software engineers.

In "Silver Bullet Talks with Jim Manico," from *IEEE Security & Privacy*, Gary McGraw interviews the founder of Manicode Security about developer training, static analysis, and open source software.

Errors in safety-critical software can have disastrous consequences. *IEEE Software's* "Code Craft" looks at tools that developers can use to thoroughly analyze software subsystems for critical safety properties.

"Application Architecture for the Internet of Cities: Blueprints for Future Smart City Applications," from *IEEE Internet Computing*, outlines three types of smart city applications and identifies key requirements and architectural guidelines.

The authors of "The Best Software Development Teams Might Be Temporary," from *IEEE Software*, studied a large company's software development

projects and found that quality often suffered if team members had worked together previously. In many cases, the authors concluded, temporary—rather than permanent—teams might be best.

In *Computing in Science & Engineering's* "The Roles of Code in Computational Science," the author outlines some of these roles and says it's important for developers to think about them before writing any code.

In *IEEE Software's* "The Social Responsibility of Software Development," the author says developers must realize that much of what they do has ethical, social, and political implications.

Technical leadership remains one of software development's great problems, according to *Computer's* "The Means of Production."

*ComputingEdge* articles on topics other than software development include the following:

- The author of "Before Torchi and Schwilgué, There Was White," from *IEEE Annals of the History of Computing*, says the invention of key-driven calculating machines occurred earlier than previously thought.
- *IEEE MultiMedia's* "Technology and the Arts: Educational Encounters of the Third Kind" reviews integrative technology-arts education.
- "The State of the Cloud," from *IEEE Cloud Computing*, examines important issues related to cloud computing. ●



# A License to Build (Software)

David Alan Grier, George Washington University

*Despite working in a global job market, software engineers must surmount local problems.*

Conference hotels look remarkably similar around the world. This similarity can lead you to believe you understand a problem when you don't. I fell into this trap while trying to answer a question about my opinion on the Malaysian government's proposal to license software engineers. After a pause, I gave a pat speech about risk control. I ran through the standard arguments and talked about the public nature of engineering, the risks borne by communities, the rights of communities to mitigate those risks, and finally, how the software industry has chosen a different strategy to address risk. My words were politely received but were followed by a moment of silence long enough to remind everyone in the room that computer scientists work in a global industry that has to deal with local problems.

The licensing proposal came as the Malaysian government was preparing a new five-year economic plan. Unlike the economic plans of the old Soviet Union or those once common in India, the 2016–2020 Malaysian economic plan is closer to a corporate strategic plan. It articulates goals, identifies infrastructure investments, catalogs best practices, and outlines policy changes. It's an aggressive document with subtle

references to the successes of India and China, two countries that serve as both models and competitors for Malaysia.

This proposal was originally part of a plan to expand Malaysia's technology workforce—which had already doubled in the prior five years—by about 30 percent between 2016 and 2020. The policymakers, or at least some of them, reasoned that requiring licensure would make software engineering jobs more prestigious. They imagined hundreds if not thousands of young workers would flock to the field, similar to the structural engineers who built the famous Petronas Towers in Kuala Lumpur.

Time and again, software developers report that they don't have the same status afforded to product managers or financial managers. So anyone who has spent time in the software field knows that there's a desperate truth behind the Malaysian reasoning, though it might not be enough to achieve the desired results. Indeed, a license might elevate a software developer's status, but it could just as easily burden the job with additional costs that would discourage people from entering the field.

In passing judgment on this proposal, I needed to move from theory to practice, from speculative reasoning to cold, hard facts. The facts came in the form of Information Technology Professional Examination Council (ITPEC) certificates. These certificates aren't licenses, but they were developed in the mid-1990s (as the software outsourcing

market began to grow) to give programmers access to the Japanese software market. Run by a Japanese nonprofit, the certificate programs are offered in Malaysia, Vietnam, Mongolia, Thailand, Myanmar, the Philippines, and Bangladesh, but they've never been popular. Over 20 years, they've brought about 3,500 programmers to Japanese firms. Less than five came from Malaysia.

By the time the final document was published, the Malaysian government had abandoned its plan to license software developers. As the idea passed from office to office and meeting room to meeting room, policymakers recognized that it could actually incentivize people to look for jobs in other fields. However, the government still needed a plan to expand the software-developer workforce, so it embraced a set of ideas that have been discussed in almost every meeting room in the world. They decided to promote early STEM education, build more technology incubators, fund nonstandard training programs, and encourage more joint programs between business and education.

To connect their technical workforce with those of North America and Europe, the government also proposed a Malaysian professional body that would govern professional competencies and ethics within the country but strive to ensure that "professionals' competencies are on par with global standards." It wouldn't be a radically new kind of organization; instead, it would be the kind that could be created in the meeting rooms of a large conference hotel. 

DAVID ALAN GRIER is an associate professor of International Science and Technology Policy at George Washington University. Contact him at [grier@gwu.edu](mailto:grier@gwu.edu).

*This article originally appeared in Computer, vol. 50, no. 2, 2017.*

# Silver Bullet Talks with Jim Manico

Gary McGraw | Cigital

Hear the full podcast at [www.computer.org/silverbullet](http://www.computer.org/silverbullet). Show links, notes, and an online discussion can be found at [www.cigital.com/silverbullet](http://www.cigital.com/silverbullet).



Jim Manico is the founder of Manicode Security, where he trains software developers on secure coding and security engineering. He's also the founder of Brakeman Security, which produces a Ruby on Rails security scanner. Jim is a frequent speaker on software security practices and specializes in the notion of building as opposed to breaking. In addition, he's a volunteer and former board member of the Open Web Application Security Project (OWASP) and authored the book *Iron-Clad Java: Building Secure Web Applications* (McGraw-Hill 2014).

**It seems to me that static analysis, which flourished just a few years ago, might be stagnating in the enterprise in some respects. Do you see that in your work?**

I see some people simply not using it, which to me in 2016 is a shock. I don't see static analysis as advanced or even interesting—it's something that everyone should do early in the systems development lifecycle [SDLC]. If anyone I work with isn't using it, I'll flag it as a major problem. Even worse, they'll use it but in very ineffectual ways—like only looking for critical issues or not letting it block anything. I see a multitude of problems: not using it, using it ineffectively, or not understanding the importance of doing it very early in the SDLC.

**Do you think lightweight approaches that are now being developed at a number of places will help?**

We have a lot of problems with moving to DevOps too quickly. People often end up using static analysis to only show critical bugs

or remove certain security areas to make it work well within an automated DevOps environment, and then major bugs—that SAST [static analysis security testing] could have found—slip through. I just want it used effectively. The trend I do like is that technology folks are integrating and building static analysis that's live in the IDE [integrated development environment]. People talked about this 10 years ago, but I see the beginnings of companies who are building it in very effective ways, so it's like a spell-check for a coder. As I'm writing code, static analysis rules will fire up in real time with advice on how to fix it. That speaks to me.

**Tell us about Brakeman, which is a Ruby on Rails static analysis tool.**

Justin Collins built this open source tool from the ground up and has been maintaining it over time. A lot of people build open source solutions, celebrate it for marketing, and then walk away from it. That infuriates me. Collins has been adding rules to Brakeman consistently for many years now, which is why I'm so impressed with him. I told him, "You should consider a commercial fork of this. I'm not saying stop the open source version, but there's a need for a pro version." A year later he did it. Along with Collins and Neil Matatall, I'm a small partial owner and guide of the project. I'm very honored to have a small part in it.

**Do you think Brakeman is making a difference in the Ruby community?**

I do. Brakeman Pro doesn't have a lot of customers, but those who jumped in are giant. Overnight we were doing business with multi-billion dollar



## About Jim Manico

Jim Manico is the founder of Manicode Security, where he trains software developers on secure coding and security engineering. He's also the founder of Brakeman Security, an open source vulnerability scanner designed for Ruby on Rails applications, and is an investor/advisor for Signal Sciences. A frequent speaker on secure software practices and a member of the JavaOne speaker community, Manico is also a volunteer and former board member of the Open Web Application Security Project (OWASP) Foundation. He's the author of *Iron-Clad Java: Building Secure Web Applications* (McGraw-Hill 2014).

communications companies that are depending on Brakeman Pro. There's not a lot out there in the commercial world that does really good Ruby on Rails analysis, and this tool does Ruby on Rails analysis really well. Frankly, I'd like to see more tools that do a couple languages extremely well instead of doing a lot of languages in a less-robust fashion.

**You're a major OWASP contributor so you must have a lot of thoughts about open source and security and so on. Can you share some of them?**

That's a big question. I think from an open source point of view, OWASP does a few things exceptionally well and a lot of stuff that we'll put in the research category. My world is mostly Java, so I like to highlight the Java tools that have been extremely disruptive from OWASP. The main one is the Dependency Check tool by Jeremy Long. It's destroyed several companies that offer similar services commercially by taking serious income away from them, and that's what positive disruption should do. It's innovated very well, it's free, it works, and it takes out products that normally cost six or seven figures. There's also a tool by John Melton called AppSensor, which is a way to embed intrusion detection alerts inside a Java application.

Rather than be supported by a large community, I want open source tools to be supported by

one or two stellar experts—like the OWASP Java encoder and the OWASP HTML Sanitizer. There's no good encoding library to stop XSS in Java, so we have Jeff Ichnowski, who wrote the OWASP Java Encoder. There's also no good HTML sanitizer for third party-authored HTML snippets, so Mike Samuel from Google wrote the OWASP HTML Sanitizer.

I see these four Java tools as either being production quality or close to production quality, disruptive, or helpful in some way where I could use them in my own projects. I want to point out those tools compared to the hundreds of other “research quality” tools that don't have that level of production quality. That's the state of the union at OWASP.

**What about all the piles of open source that security people have to contend with that are outside the security tools range?**

It's scary because there's trillions and trillions of insecure lines of code that we depend on right now. Somewhere at this moment a developer is looking for a third-party library to satisfy a programming task on their plate. They find it, they download it, they mess with it for a little while, it works, and now it's live in their project. But they miss security analysis, even due diligence, in looking at the security history of this library. This developer didn't look at the code

and compound this by millions of times, and that leaves us where we are today. Dependency Check gives me a little something. It's not perfect, but at least I'm now doing my due diligence as a developer or server administrator.

**Let's talk a little bit about the evolution of Java. Scala has in some sense served as a training ground for new language features. It seems the way they're trying stuff out in Scala is causing Java to adopt some of those features more quickly these days. Do you think that's true?**

Let me pivot on you. Java 8 and 9 have an extraordinary number of new security features to help Java developers build more secure apps. That's what excites me about Java. This is the talk that [I gave at JavaOne in September 2016], where I highlighted the security features of the newest versions of Java. They're trying to improve the cryptography and TLS and get internal validation. So I think they're learning from the hipster languages, the componentization world, the need for security deeper in the framework. I'm very happy with the movement Java's making. They're not as far as .NET, which is another 10 years ahead of the game when it comes to LangSec [language-theoretic security], but Java is getting there and it's putting in a lot of serious effort.

**What are your thoughts about instructor-led security training versus computer-based training? What's the most effective way to teach developers secure coding?**

I think a combination of both is going to be important. Instructor-led training [ILT], especially by someone who's a developer and speaks the language of developers and has good up-to-date material, is a great way to get everyone on the same page. But once the trainer leaves, developers will have questions. So we need to have a center of excellence and

other things available to developers beyond just the training.

My favorite customers go through every slide, make change requests, and have their own standards to add to the material. I think those kinds of initiatives where the customer really owns the material and digs into it and asks for changes is how you get the most out of an ILT.

Computer-based training [CBT] scales extremely well. You can't bring in a hundred trainers for a large team, so we often need CBT to supplement what's done in a normal ILT environment. In the past, some of the different CBT stuff I've seen is absolutely horrible.

I'm working with one of your companies, Codiscope, right now to build high-quality CBT training material. It's part of the new trend in microlearning: it's a way for developers to get on-the-spot snippets of training while they're working. Imagine a scenario where an instructor shows up, works with the company to make sure the material is appropriate for what they're trying to accomplish, and then starts using a tool where developers get alerted with these same ideas while they're working and three-to five-minute CBT trainings pop up as they're going. That's going to change how developers write code, and it's the direction I believe the industry should be moving.

**Let's go back to OWASP to close things out. Now that the OWASP Top 10 has entered the zeitgeist, what's next for OWASP? What would you like to see?**

My recommendation is that everybody is allowed to read the Top 10 once and then they have to be banned from ever looking at it again. Anytime someone is building a software assurance program, they shouldn't be allowed to use the term "OWASP Top 10" at all. Anyone who is delivering training

should never be allowed to mention the OWASP Top 10 because there are more than 10 things. I want to replace all global mentions of the OWASP Top 10 with the OWASP Application Security Verification Standard [ASVS]. It's not perfect but it's a 200-item, three-tiered standard on how to achieve basic web-application and (to some degree) mobile- and web-service security. People pay big money to get secure coding standards for their group—the ASVS can help that process. I would never use the ASVS off the shelf as it is. I would fork it and make sure developers own it.

When I'm trying to build a secure coding standard for a group of developers, I don't hand them a standard from security and say, "You have to do this." I sit with them and go over the ASVS one at a time and see which ones the developers want to accept and which ones they don't. Then it becomes a list of standards that they decided was good for their organization. That's a great first step in establishing a standard.

**T**he Silver Bullet Podcast with Gary McGraw is cosponsored by Cigital and this magazine and is syndicated by SearchSecurity. ■

**Gary McGraw** is Cigital's chief technology officer. He's the author of *Software Security: Building Security In* (Addison-Wesley 2006) and eight other books. McGraw received a BA in philosophy from the University of Virginia and a dual PhD in computer science and cognitive science from Indiana University. Contact him via [garymcgraw.com](http://garymcgraw.com).

myCS

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

This article originally appeared in IEEE Security & Privacy, vol. 14, no. 6, 2016.

got  
flaws?



Find out more  
and get involved:

[cybersecurity.ieee.org](http://cybersecurity.ieee.org)

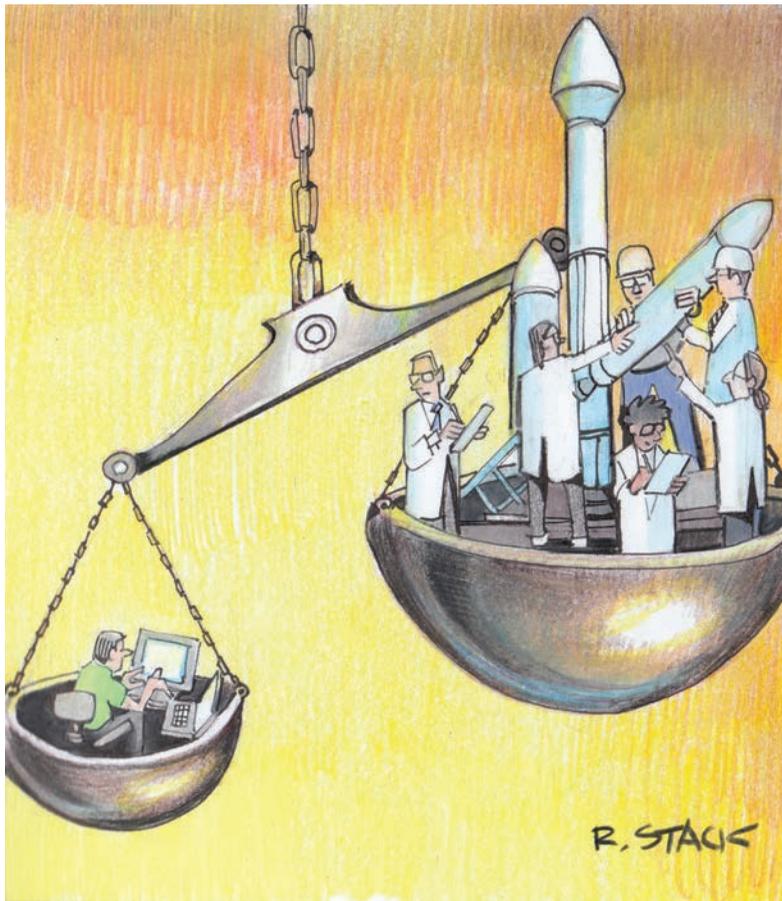


IEEE computer society



# Code Craft

Gerard J. Holzmann



**WHICH IS HARDEST:** overcoming the many physical challenges of autonomously landing a spacecraft on the surface of a distant planet, building the

many interacting pieces of hardware to execute such a mission, or writing the software that controls the whole process? To me, the first two tasks look impossibly hard. But, if we look at a series of failed attempts to accomplish this feat, it might well be that writing the software correctly is still the hardest problem.

The recent loss of the Schiaparelli spacecraft from the European Space Agency, now thought to have been caused by a software defect, is a powerful reminder of this fact. However, it's certainly not the only one, nor is it likely to be the last. Why is writing bug-free code so hard? Over half a century ago we hopefully started calling our discipline "software engineering," but how much do we really have in common with more traditional engineering disciplines?

Quite a few people have tried to compare the design processes used in software engineering with those used in civil engineering.<sup>1</sup> The comparison is usually somewhat accusatory, questioning why software engineers make such scant use of tools that could help them model and analyze software systems before they're implemented. It would be unthinkable for a civil engineer to propose building a structure before its design was carefully vetted in analyti-

cal models, audited for compliance with all the relevant standards, and checked for respecting proper safety margins. Why is this still so different in software engineering?

I'll cross my fingers that not too many actual civil engineers will read this column, because they would probably disagree with the following statement: Modeling and analyzing the design of a physical structure might be easier than modeling and analyzing the reliability of a complex software system. I say this because a large bridge or tall building is unlikely to come tumbling down if a single rusty nail hiding somewhere deep in its internal structure unexpectedly breaks. Similarly, paint peeling off at one end of a bridge is unlikely to cause the roadway to collapse at the other end. Yet this type of thing can happen in large software systems.

The often-made analogy between constructing a bridge or house and constructing a software system is therefore flawed. Building software is perhaps closer to constructing a sand castle at the beach or playing a game of sticks. Building sand castles instead of concrete structures requires different skills and tools.

### Robustness

Even though many of us feel that hardware systems can be more robust than software systems, there's an enduring trend to move functionality gradually away from hardware and into software, even in safety-critical systems. Software is clearly simpler to update or extend than hardware, and we all want to believe that, with sufficient care, software can be just as reliable. There are many reminders that justifying this might require more evidence than is often assumed. Building ro-

bust software really does require a different approach.

### Therac-25

The Therac-25 radiation therapy machine is perhaps one of the best-known examples of the danger of blindly moving critical safety protection mechanisms from hardware to software.<sup>2</sup> The Therac-25 was introduced in the early '80s as an improvement over two successful earlier machines, the Therac-6 and Therac-20. Those machines had been developed with minimal software support, relying on solid hardware interlocks for their safety features. The new machine was instead designed to be software controlled.

Between June 1985 and January 1987, the Therac-25 accidentally delivered six massive overdoses, resulting in serious injuries and in some cases death. Although these accidents had several causes, a few stand out. One cause was a race condition in the code that allowed the machine to apply radiation at a much higher level than the operator intended. The effect of the race condition depended on the speed with which proficient operators could enter settings into the machine and on the ease with which they could override inscrutable error codes.

It was later found that the Therac-6 and Therac-20 also had some of the critical design defects in their software but that their independent hardware interlocks had prevented accidents. Replacing the hardware interlocks with software versions, combined with misleadingly simple software overrides for warnings, had inadvertently introduced serious new risks. The manufacturer of the Therac-25 initially denied that the machine, which it believed had been well-tested, could

have possibly caused the overdoses. It later transpired that the actual testing had been minimal and that virtually no meaningful analysis of the safety of the reused parts of the software had been performed.

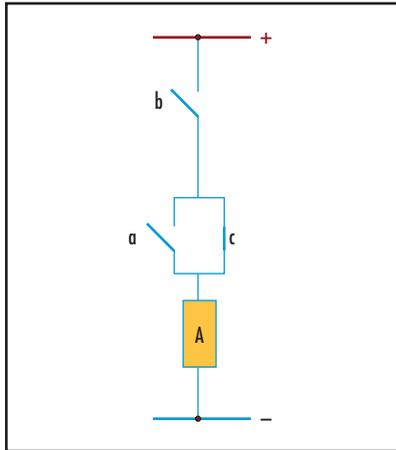
Tools for detecting race conditions and data corruption errors in software systems weren't generally available when the Therac-25 software was developed, but that changed soon after. Next, I'll show how tools that have been available for about a quarter century now can be used to thoroughly analyze both hardware and software subsystems for critical safety properties.

### Hardware Interlocks

Peter van Eijk, a former colleague of mine, once published a fairly straightforward model-checking analysis of a hardware lockout circuit used in trains.<sup>3</sup> The relay circuit was meant to ensure that trains with a driver cabin at each end couldn't be operated from both cabins at the same time.

A relay is basically an electromagnet that can open or close switches when it's connected to a power source. By using a combination of relays in an electrical circuit, you can design fairly complex systems that aren't all that easy to analyze with traditional means. Van Eijk first explained how a simple Boolean formula can describe the working of a system of switches and relays.

For example, the circuit in Figure 1 contains three switches—*a*, *b*, and *c*—and a single relay *A*. In Figure 1, *c* is closed, and *a* and *b* are open. The horizontal line at the top represents a power source; the horizontal line at the bottom represents the ground. Relay *A* will be powered when *b* is closed and either *a* or *c* is closed. I can express that as  $A = b \text{ and } (a \text{ or } c)$ .



**FIGURE 1.** A simple relay corresponding to  $A = b \text{ and } (a \text{ or } c)$ , where  $A$  indicates the relay and  $a$ ,  $b$ , and  $c$  indicate switches.  $A$  will be powered when  $b$  is closed and either  $a$  or  $c$  is closed.

I now add one final wrinkle to the functioning of this circuit. I'll use the convention that every switch controlled by a relay carries the same letter as that relay, but in lowercase. So, in Figure 1,  $A$  controls  $a$ : if  $A$  is powered, then  $a$  will close. This ensures that once  $A$  is powered it can remain so, independently of the position of  $c$ .

Figure 2 shows a version of this simple circuit extended to implement a hardware lockout between the front and rear train cabins. The figure shows two copies of a small extension of the circuit from Figure 1, both connected to the same power and ground. The circuit adds `in_service`, an additional horizontal wire that's powered when at least one of  $a$  and  $b$  is closed. The notation  $!a$  indicates that a switch controlled by  $A$  is closed instead of open in the unpowered state.

To avoid confusion, I'll distinguish the various switches in the two copies of the circuit with an index. For example,  $a[0]$  refers to  $a$  in the

left-hand copy, and  $a[1]$  refers to the same switch in the right-hand copy.

The new circuit has four relays, two on each side. There are also four switches marked with either `key` or `!key`. These switches are controlled by an operator inserting a key in the front or rear train cabin. Figure 2 shows all switches in the unpowered state, with neither key inserted. When the  $B$  relay on either side is powered, that side of the train will control the train. So, it's important that both  $B$  relays can't be powered simultaneously.

## Driving the Train

When the driver in the front cabin inserts a key to take control of the train, switch `key[0]` closes and, simultaneously, switch `!key[0]` opens. That means that relay  $B$  receives power and causes switch  $b[0]$  to close. Relay  $A[0]$  stays unpowered because both switches  $a[0]$  and `!key[0]` are now open. But, `in_service` is now powered, which means that relay  $A[1]$  in the other cabin will be powered as well. This in turn closes switch  $a[1]$  and opens `!a[1]`. Because `!a[1]` is now open, inserting the driver's key in the rear cabin will have no effect. Similarly, because  $a[1]$  is now closed,  $A[1]$  will stay powered independently of whether the second key is present. So, all seems to work as intended with this circuit. But is it really reliable?

It's not terribly complicated to capture the definitions of the state of each switch from the new version of the circuit in a Boolean formula like I did for Figure 1. For  $A[0]$ , for instance, I can write

$$A[0] = (b[0] \text{ or } \text{in\_service}) \text{ and } (a[0] \text{ or } !key[0])$$

It's straightforward to build a small Spin model that mimics the

functionality of the entire circuit, including the possible actions of the train operators inserting or removing their key. Figure 3 shows the model, which defines two processes called `cabin`. The two copies are distinguished by a unique process identification number, `_pid`, that's either 0 or 1 in this case. I use `_pid` (assigned to variable `s`) to set the correct value for each of the  $a$  and  $b$  switches and the state of `in_service`.

The fourth line in the `do` loop in this small model indicates that the driver in either cabin can arbitrarily insert or remove his or her key. The fifth line, finally, contains an assertion that states that the  $b$  switches can't both be closed at the same time, which would happen if the  $B$  relays were powered simultaneously.

To analyze the correct functioning of the circuit, all I have to do is run the model checker and ask it whether any possible scenario exists that could lead to the failure of the assertion. (Source and binary versions of the Spin model checker are freely available at [spinroot.com](http://spinroot.com).) The following command runs the analysis, using a breadth-first search to get the shortest-possible trace to an assertion violation, in case one is possible:

```
$ spin -run -bfs train.pml
assertion violated ( !(b[s])|!(b[(1-s])) (at depth 4)
wrote train.pml.trail
```

The search completes virtually instantly, finding an assertion violation after just four steps. I can replay that scenario as follows, printing every process step that's executed and ending with the resulting settings of all switches:

```
$ spin -p -replay train.pml
using statement merging
```

This article originally appeared in IEEE Software, vol. 34, no. 2, 2017.

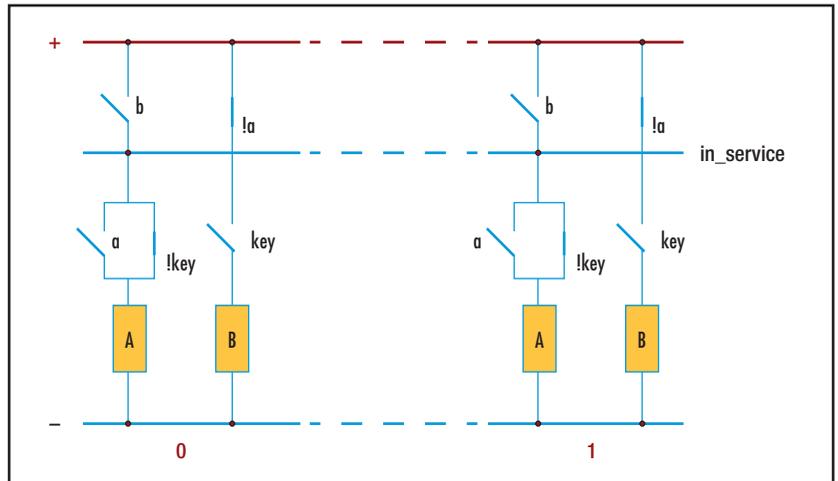
```

1: proc 1 (cabin:1) train.pml:8 [key[s] = !(key[s])]
2: proc 1 (cabin:1) train.pml:7 [b[s] =
  (key[s]&&!a[s])]
3: proc 0 (cabin:1) train.pml:8 [key[s] = !(key[s])]
4: proc 0 (cabin:1) train.pml:7 [b[s] =
  (key[s]&&!a[s])]
5: proc 1 (cabin:1) train.pml:10[assert()]
spin: train.pml:10, Error: assertion violated
spin: text of failed assertion:
assert(!!(b[s])||!(b[1-s]))
spin: trail ends after 5 steps
#processes: 2
  a[0] = 0
  a[1] = 0
  b[0] = 1
  b[1] = 1
  key[0] = 1
  key[1] = 1
  in_service = 0
5: proc 1 (cabin:1) train.pml:5 (state 6)
5: proc 0 (cabin:1) train.pml:5 (state 6)
2 processes created

```

In this scenario, first driver 1 inserts a key, thereby closing switch `key[1]` and opening `!key[1]`. Next, relay `B[1]` is powered, which closes switch `b[1]`. That in turn will apply power to `in_service`, but that step isn't recorded in the scenario yet. Before the circuit can stabilize further, driver 0 now inserts the second key, closing `key[0]` and opening `!key[0]`. This powers `B[0]` and closes `b[0]`. If `!key[0]` opens before `b[0]` closes, `A[0]` remains unpowered and `!a[0]` remains closed. Similarly, on the right side of the circuit, `A[1]` remains unpowered and `!a[1]` remains closed, leaving the circuit in a stable state with both drivers in control of the train.

The scenario reveals a race condition in the circuit that would be hard to spot without using tools. It's not too hard to fix the circuit to prevent such race conditions from having ill effects,



**FIGURE 2.** A relay circuit that once was used as a hardware lockout for trains. In trains that can be controlled from both the front (left) and rear (right) cabins, only the first key inserted by a driver should succeed in powering the `B` relay on that side, which then supplies power to the `in_service` wire.

once we know the circuit isn't quite right yet.

The use of software analysis tools such as the one I used here still seems to be somewhat of an exception, even for safety-critical systems. Would it not be fair to say that if we really want software development to be considered a true engineering discipline, the use of these tools should become standard? For both hardware and software design, not using the right tools is just not cool. ☹

## References

1. L. Lamport, "Who Builds a House without Drawing Blueprints?," *Comm. ACM*, vol. 58, no. 4, 2015, pp. 38–41.
2. N. Leveson and C. Turner, "An Investigation of the Therac-25 Accidents," *Computer*, vol. 26, no. 7, 1993, pp. 18–41.

```

bool a[2], b[2], key[2], in_service
active [2] proctype cabin()
{
  bool s = _pid
  do
  :: a[s] = (b[s] || in_service) && (a[s] || !key[s])
  :: b[s] = key[s] && !a[s]
  :: in_service = b[s] || b[1-s]
  :: key[s] = !key[s]
  :: assert(!b[s] || !b[1-s])
  od
}

```

**FIGURE 3.** A small Spin verification model to check the correctness of the circuit in Figure 2. The model checker reveals a race condition that would be hard to spot otherwise.

3. P. van Eijk, "Verifying Relay Circuits Using State Machines," *Proc. 3rd Spin Workshop*, 1997; spinroot.com/spin/symposia/ws97/eijk.pdf.

**GERARD J. HOLZMANN** works on developing stronger methods for the design and analysis of safety-critical software as a consultant and researcher at Nimble Research. Contact him at gholzmann@acm.org.



# Application Architecture for the Internet of Cities: Blueprints for Future Smart City Applications

Johannes M. Schleicher, Michael Vögler,  
and Schahram Dustdar • TU Wien, Austria

Christian Inzinger • University of Zurich, Switzerland

As smart cities evolve towards the Internet of Cities, it's essential to consider common problems and how to address them. Three representative types of smart city applications are outlined here, identifying key requirements and architectural guidelines for implementation.

**T**he rapid rise and pervasive adoption of the smart city paradigm in conjunction with the Internet of Things (IoT) is reshaping the world's metropolises. Today's cities have evolved to become behemoths of connected devices cutting through all vital domains, starting with building management and operations to smart grids and multimodal traffic management up, to social media-driven citizen-participation processes. To deal with this increasing complexity, experts and stakeholders alike must be able to utilize this plethora of information to ensure optimal city planning, operations, and management. It's vital to provide stakeholders with capabilities for understanding the massive amounts of data generated, while also supporting the efficient operation and management of increasingly complex infrastructures.

This new city concept also comes with the ability to let citizens participate – this is an integral element in advancing the city as a whole. Thus, in previous work, we introduced the Smart City Application Ecosystem (SCALE) to provide citizens with a toolset to build novel kinds of applications.<sup>1</sup> (For others' work in this area, please see the related sidebar.) Figure 1 shows SCALE's architecture; beyond helping practitioners build applications with novel capabilities emerging through the IoT, it also helps them efficiently access the mas-

sive amounts of data. These applications in turn become composable, interchangeable abstractions of capabilities, much like the applications known from today's smartphones, but on a much larger scale. This evolution promises to be a vital stepping stone to reach the *Internet of Cities*,<sup>2</sup> an open market place where applications can interact and be exchanged between cities. The Internet of Cities will let practitioners use best practices and solutions from other cities to solve common problems in their own environment.

Such an ambitious move forward logically requires forethought about possible scenarios along the way. In order for practitioners to understand where and how to apply SCALE in such a capacity, we identified common application cases that today's smart cities are facing. Here, we present best practices (blueprints) for these common cases, show how they can be addressed, and explain which elements of SCALE – or more specifically, SCALE's core, the Smart City Operating System (SCOS) – must be applied to enable them.

## Blueprints for Smart City Applications

Here, we outline blueprints for three different cases we identified as relevant throughout our smart city research. We identify their integral requirements

## Related Work in Designing Smart Cities

The broad coverage of the smart city paradigm in research initiatives around the globe has led to several approaches to address the inherent complexities. These approaches mainly focus either on the Internet of Things (IoT) aspect or on general complexity and integration issues surrounding smart cities.

SmartSantander<sup>1</sup> proposes a city-scale experimental research facility that supports applications and services in the smart city context. The primary focus is the IoT element of a smart city, and SmartSantander aims to provide a large-scale testbed that helps with issues arising from connecting and managing IoT infrastructure elements.

Jiong Jin and colleagues<sup>2</sup> present a framework that focuses on the IoT aspect for creating smart cities. Another approach in the context of IoT comes from Nathalie Mitton and colleagues,<sup>3</sup> where the authors propose the combination of a cloud and sensors in a smart city environment. In their work, they focus on pervasive infrastructures, where services interact with their surrounding environment.

In terms of integration, Jiafu Wan and colleagues<sup>4</sup> present an event-based architecture for machine-to-machine (M2M) communications for smart cities based on the Stratospheric Observatory for Infrared Astronomy (SOFIA) project, with a case study in the vehicular context. On a more abstract level, Hafedh Chourabi and colleagues<sup>5</sup> present a framework to understand the concepts of smart cities. The authors introduce a conceptual framework to comprehend the vital elements in a smart city by identifying critical factors. This approach is similar to that of Taewoo Nam and Theresa Pardo,<sup>6</sup> who try to identify how a city can be considered smart by aligning strategic principles in the context of technology, people, and institutions, which represent the main dimensions of a smart city. In this broader context, Catherine Mulligan and Magnus Olsson<sup>7</sup> discuss the architectural implications of smart city business models. The authors specifically discuss the architectural evolution required for ensuring a smooth rollout and deployment of smart city technologies.

In the context of citizen participation, Zaheer Khan and Saad Liaquat Kiani<sup>8</sup> introduce a cloud-based architecture for context-aware citizen services in smart cities. The authors argue that cloud computing can provide a suitable computing infrastructure for smart city applications and also emphasize the importance of security considerations in this context. In a similar manner, Roozbeh Jalali and colleagues<sup>9</sup> present a smart city architecture for community-level services through IoT. Their smart city architecture enables community service providers and citizens alike with access to real-time data, which has been gathered through IoT as a basis for decision processes and future planning.

### References

1. L. Sanchez et al., "SmartSantander: IoT Experimentation over a Smart City Testbed," *Computer Networks*, vol. 61, no. 11, 2014, pp. 217–238.
2. J. Jin et al., "An Information Framework for Creating a Smart City Through Internet of Things," *IEEE Internet of Things J.* vol. 1, no. 2, 2014, pp. 112–121.
3. N. Mitton et al., "Combining Cloud and Sensors in a Smart City Environment," *Eurasip J. Wireless Comm. and Networking*, vol. 2012, no. 1, pp. 247–257.
4. J. Wan et al., "M2M Communications for Smart City: An Event-Based Architecture," *Proc. IEEE 12th Int'l Conf. Computer and Information Technology*, 2012, pp. 895–900.
5. H. Chourabi et al., "Understanding Smart Cities: An Integrative Framework," *Proc. 45th Hawaii Int'l Conf. System Sciences*, 2012, pp. 2289–2297.
6. T. Nam and T.A. Pardo, "Conceptualizing Smart City with Dimensions of Technology, People, and Institutions," *Proc. 12th Ann. Int'l Digital Government Research Conf. Digital Government Innovation in Challenging Times*, 2011, pp. 282–291.
7. C.E.A. Mulligan and M. Olsson, "Architectural Implications of Smart City Business Models: An Evolutionary Perspective," *IEEE Comm.*, vol. 51, no. 6, 2013, pp. 80–85.
8. Z. Khan and S.L. Kiani, "A Cloud-Based Architecture for Citizen Services in Smart Cities," *Proc. 2012 IEEE/ACM 5th Int'l Conf. Utility and Cloud Computing*, 2012, pp. 315–320.
9. R. Jalali, K. El-khatib, and C. McGregor, "Smart City Architecture for Community-Level Services through the Internet of Things," *Proc. 18th Int'l Conf. Intelligence in Next Generation Networks*, 2015, pp. 108–113.

and present approaches for each case that demonstrate how best practices for each case should look like.

### Case I: Smart City Planning

Today's cities and their respective stakeholders want to provide their citizens with a sustainable, secure, affordable, and livable city. In order to provide stakeholders in a smart city environment with the optimal foundations for a reflected planning and

decision process to support such a lasting evolution, it's essential to enable a holistic view of the city. To achieve this, we must integrate every important aspect of the city, seen as a complex system, and provide an understandable view for the stakeholders. This can be achieved by enabling domain experts from relevant fields such as energy, mobility, sociology, or buildings, among others, to integrate their findings and views with one another. Such

an integration allows for novel views as well as more meaningful integrated insights. For example, understanding the social milieu of a district could help derive certain mobility patterns, which in turn enable novel aspects of multimodal transport planning, and so on. The fuel for these integrated insights are the domains experts' different models that express certain facets of their respective domains. To enable these models, we must supply

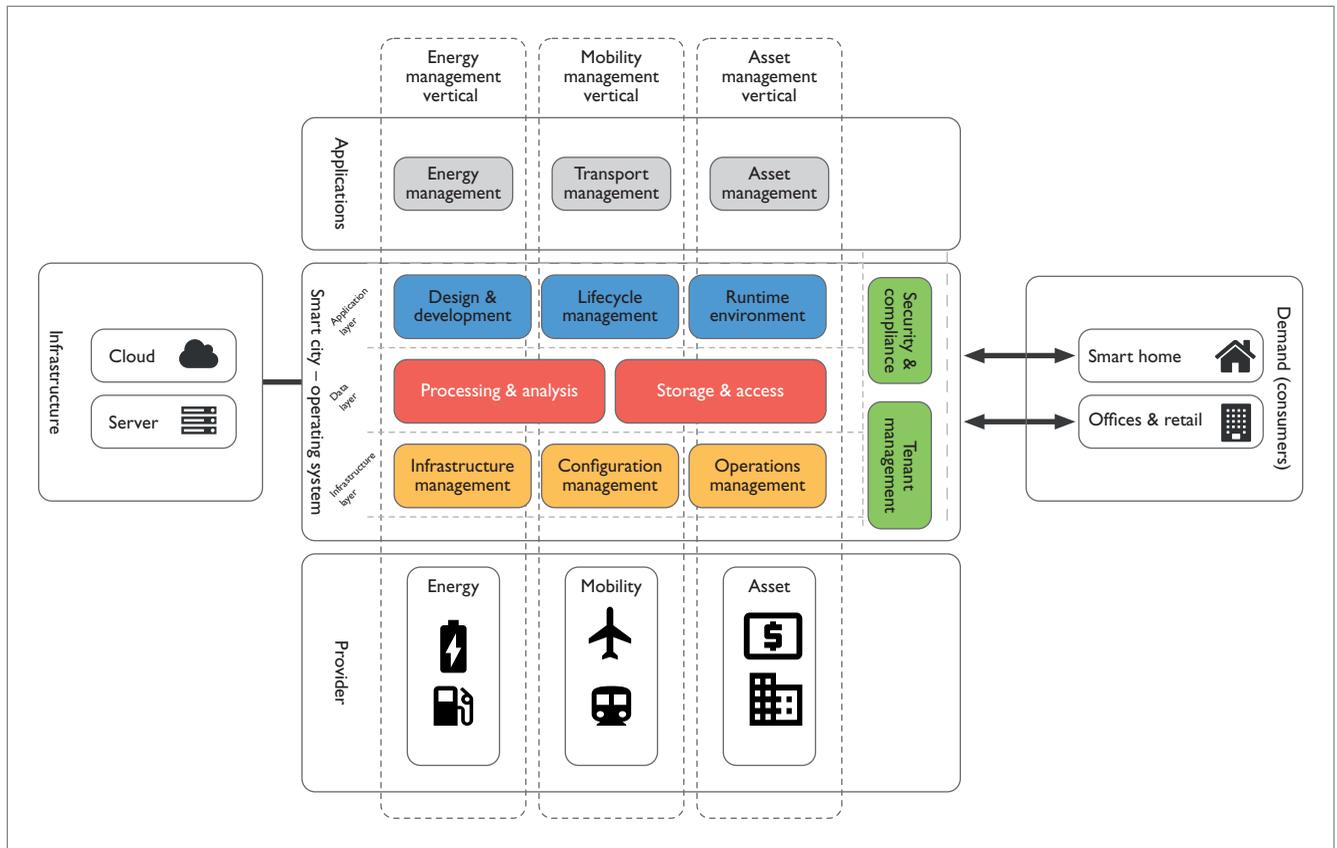


Figure 1. Smart City Application Ecosystem (SCALE) architecture. Beyond helping practitioners build applications with novel capabilities emerging through the Internet of Things (IoT), SCALE also helps them efficiently access the massive amounts of data generated.

them with the most accurate and novel data about the city. Then they can provide a more precise analysis of a city’s current situation, while also supporting more accurate forecast and prediction models.

**Requirements.** To enable this case, the following requirements must be met: For the domain experts to be able to use the most current and accurate data in their respective models, they must be able to access all the relevant data provided by the city. This massive amount of data comes from a wide variety of sources, ranging from the IoT, to open data sources, up to old documents. This means that there needs to be a way to manage massive amounts of diverse data while also integrating heterogeneous data sources. Because the experts rely on

proven tool stacks to implement their models, which in turn consist of a number of legacy systems combined with older mature system stacks, it’s vital to support legacy systems and tool stacks for model building. Additionally, all these models need the computational capacity to provide their results in a reasonable amount of time, which calls for infrastructure resource provisioning for efficient model execution. Finally, the smart city domain with its many stakeholders and data providers has a highly complex set of security and compliance constraints. Therefore, to optimally use all the available data, it’s essential to provide measures for compliance and security enforcement.

**Approach.** To enable the presented case, we must address the outlined

requirements. Thus, we introduce a blueprint here that accomplishes this by utilizing several components of the SCOS. We show the blueprint for this smart city planning in Figure 2, which consists of four main parts. In order to support the stakeholders in understanding and exploring the complex aspects of the city in a holistic way, we introduce a visualization component. It enables geospatial visualizations of different aspects of the city by integrating the results of the domain experts’ models.

The second component that’s relevant for stakeholder interaction is the decision-support component, which provides detailed analytics and statistics based on the aforementioned integration of different domain model results. Both components rely on SCOS support in two main areas. First,

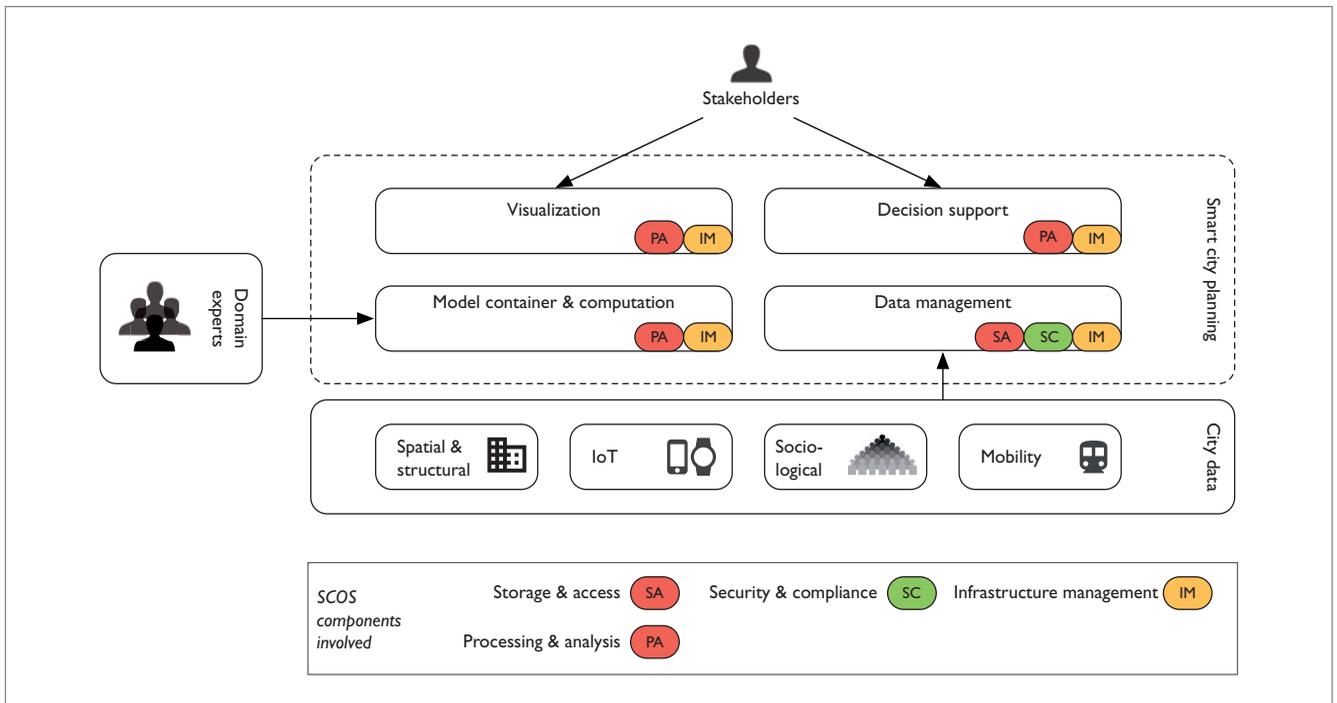


Figure 2. Smart city planning blueprint. SCOS stands for Smart City Operating System.

they need processing and analysis’s pre-built services in order to be able to effectively prepare the data for visualization and decision support. Additionally, they rely on infrastructure management to provide them with the necessary infrastructure resources to efficiently perform their tasks.

Both components in turn rely on the model container and computation component, which is responsible for executing the domain experts’ models. This component is tightly integrated with the data management component, which ensures that the experts are able to access all the diverse city data that’s in turn used as the basis for their models. Additionally, it facilitates the standard and pre-built services provided by the SCOS’s processing and analysis to support the model building when feasible.

The data management component acts as a domain-specific data managing entity, and is used by all other components. It ensures that the complex compliance and security requirements originating from the dif-

ferent data providers are being met by utilizing SCOS’s security and compliance. Additionally, it utilizes the storage and access, as well as the infrastructure management components to ensure efficient and effective data handling of the massive amount of city data.

### Case 2: Smart City Infrastructure Management and Operation

A city consists of a network of complex systems that interfuse all layers of its infrastructure. From building and traffic management up to grid management, controlling all these systems becomes more data-intensive and complex, but also more capable. Increasingly, this is the case with the rise of the IoT, where more and more sensor and devices become critical elements of said infrastructure.

Thus for stakeholders, it’s imperative to have processes that enable the effective management and operation of these complex systems in a transparent and efficient way. First and foremost, stakeholders need means to configure and control this cyber-physical infra-

structure, so they can manage it in a logically centralized way (which is a vital factor for cost efficiency).

The second important facet is the distributed autonomous and timely control of this infrastructure, to be able to act and react to fast-changing circumstances. This is not only essential in areas such as building management – where heating, ventilation, and air conditioning (HVAC), lighting, or access control systems need to be managed – it becomes even more important in areas such as traffic and grid management. The ability to have fine-grained, instant control is crucial for enabling multimodal transport management or paving the way for the advent of autonomous cars, where crossroad management becomes a highly sophisticated task. Integrating smart metering facilitates smart management of grid use and leads to significant cost reductions for operators and consumers alike.

Finally, all this information needs to be composed in a smart manner, so that operators can control the systems and physical infrastructure efficiently.

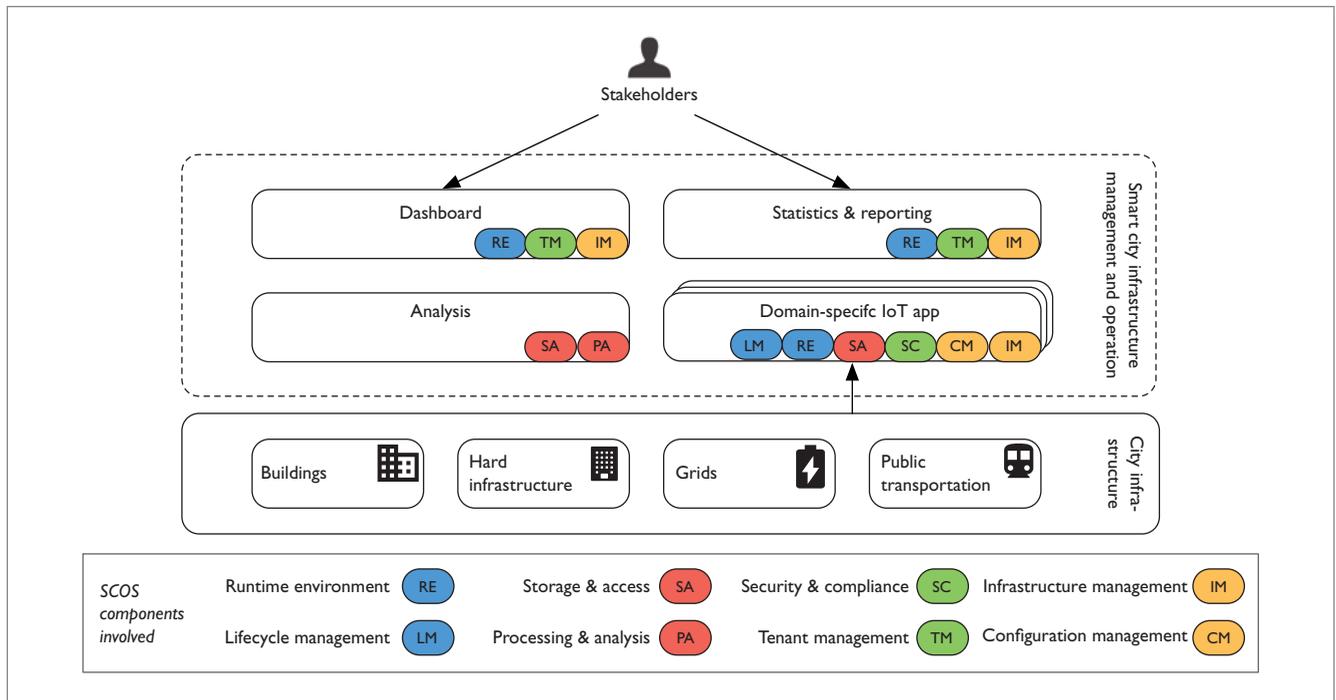


Figure 3. Smart city management and operation blueprint. The blueprint consists of four main parts, supported by various components of SCOS, to manage and operate available smart city infrastructures such as buildings, hard infrastructure (traffic lights, bridges, and roads), energy grids, or public transportation.

**Requirements.** In order to allow such smart infrastructure management and operation, several requirements need to be met. First, there needs to be a way to provide IoT device lifecycle management. This includes ways for device provisioning, management, and integration. Next it is vital to provide IoT cloud application lifecycle management, which calls for means to provide the necessary cloud infrastructure resources to enable these IoT devices. Additionally, means for IoT applications lifecycle management need to be covered to ensure applications can be deployed and updated including the vital element of fault handling. The large amounts of data these systems generate call for a way to manage this massive amount of diverse data. Last but not least, these integrations touch many security critical domains so it is essential to provide compliance and security enforcement.

**Approach.** To address the identified requirements for the presented case,

we introduce the Smart City infrastructure management and operation blueprint in Figure 3. In essence, the blueprint consists of four main parts, which are supported by various components of SCOS, to manage and operate available smart city infrastructures like buildings, hard infrastructure (traffic lights, bridges, roads), energy grids, or public transportation.

First, to allow stakeholders to easily gain insights of the managed smart city infrastructure, we introduce a dashboard component. Dashboards support displaying both real-time, processed, and historical performance data (such as energy consumption) of the infrastructure. Furthermore, dashboards are specifically built for non-technical users, to avoid the need for custom programming, while supporting great flexibility by providing visual designers that allow stakeholders to create visualizations that are tailored to the available data and demands.

Second, next to dashboards, another important aspect in infrastructure man-

agement and operation is the statistics and reporting component. This component supports stakeholders in creating customized reports and statistics about the managed infrastructure, which can be used to deliver transparency into the gained performance data, and furthermore promote compliance.

Third, to provide the required data for the first two components, we introduce an analysis component. Based on the gathered and processed data that's provided by SCOS via the storage and access, the analysis component is able to conduct specific operations such as examining trends in the collected infrastructure data (for example, power conditions or temperature readings). Additionally, the analysis component facilitates standard and pre-built services provided by SCOS's processing and analysis. The analysis results can then be used to identify problems in the future or areas that require maintenance. Following this approach can increase the uptime of the infrastructure by

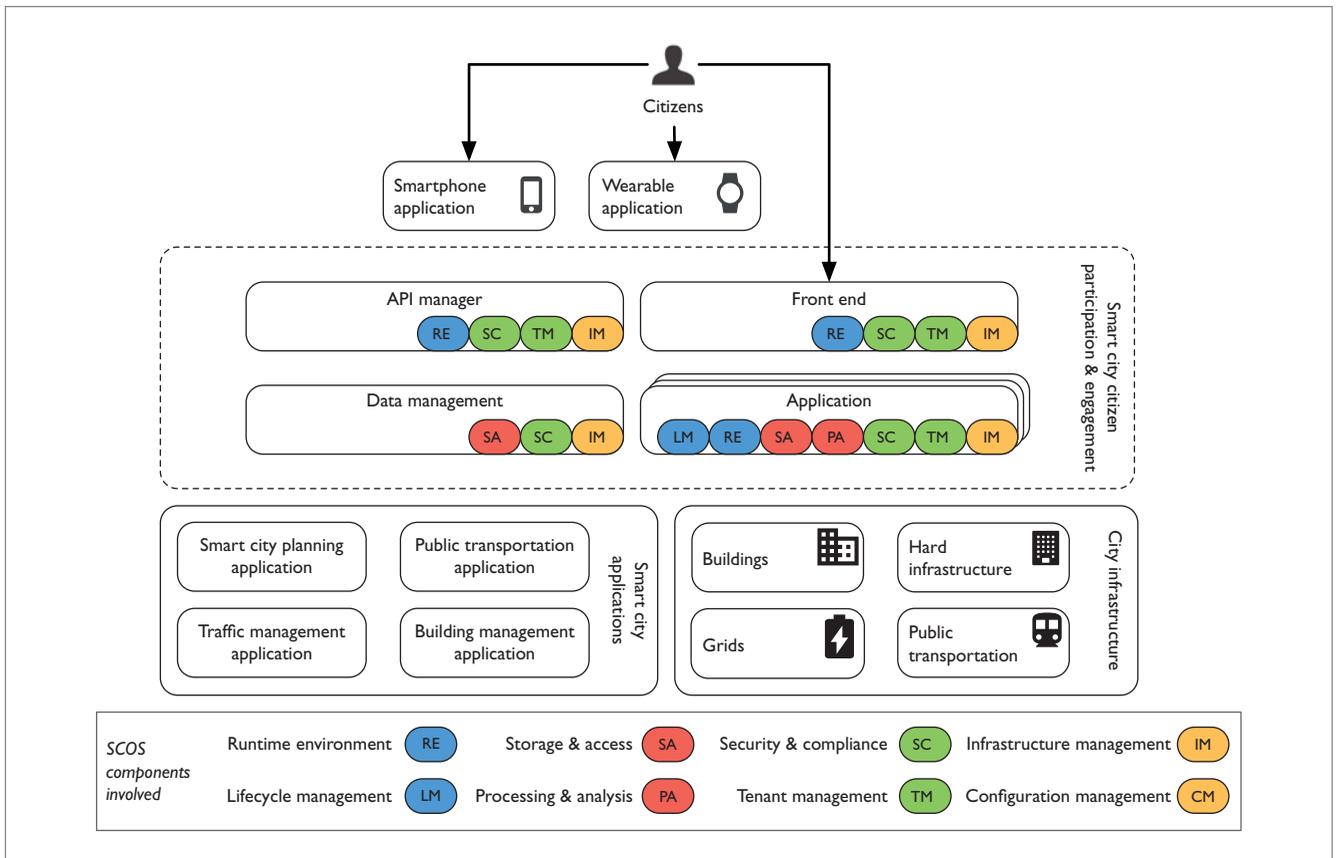


Figure 4. Smart city citizen participation and engagement blueprint. Citizens access the application using Web-based front ends or native applications deployed on smartphones or wearable devices that access the application through a managed API (provided by the API manager).

dealing with issues before they cause downtimes.

Finally, to collect data from the actual infrastructure and execute business logic, we facilitate a set of domain-specific IoT apps. These domain-specific apps can be executed in SCOS or directly in the actual smart city infrastructure. The apps gather data by facilitating the magnitude of IoT resources that are residing in the smart city infrastructure and are managed by SCOS. Furthermore, next to data collection, the domain-specific apps are able to control IoT resources to react to changes in the environment or optimize energy consumption of the managed infrastructure. To discover and communicate with IoT resources, the applications use the infrastructure management

and configuration management of SCOS. While the apps are executed by facilitating the runtime environment of SCOS, SCOS also manages their complete application lifecycle via the lifecycle management. Furthermore, because both the dashboard and statistics and reporting component require data and resources of multiple domains, SCOS handles and possibly restricts access to shared resources via the tenant management.

### Case 3: Smart City Citizen Participation and Engagement

Another core tenet of smart cities is the close integration of citizens with various aspects of the city to increase citizen well-being and quality of living. Examples of such citizen participation and engagement

applications are smartphone-based city transport information systems that let users quickly, efficiently, and ecologically travel through the city while anonymously providing data to transportation providers and city administrators. In turn, this data can be used to more accurately estimate upcoming demand, optimize city traffic by redirecting citizens away from congestions in both public and individual transports, and guide infrastructure planning.

Another case is the integration of social media feeds to detect potential trends in citizen movement or concentration points, to help first responders prevent or shorten the detection time of disaster scenarios. The utility of such applications is manifold, and not only benefits providers

and citizens alike, but also creates a better understanding of the city as an adaptive system that benefits engagement. To enable the creation of such services, it's essential to provide means for composing and integrating data from multiple smart city applications, ranging from transport and infrastructure management to grid control, up to applications in the social media domains. Based on these data, we need fast-feedback loops that let citizens and providers act on new information as soon as possible.

**Requirements.** Such citizen participation and engagement applications are characterized by a number of requirements. Because these applications heavily rely on data from many different stakeholders in the smart city – ranging from public transport providers and road traffic authority, up to energy providers – we must ensure that there's a unified way to securely and efficiently access data from multiple other smart city applications. Also, due to the integration and enrichment of information from multiple sources, there must be a way to effectively manage large amounts of diverse data. Integrating data about citizen movement and the use of city infrastructure further requires that compliance and privacy regulations are guaranteed, and that relevant security policies and tenant requests are enforced. Additionally, the runtime infrastructure for the application must be elastically provisioned and, if possible, deployed on cloud and edge infrastructure to minimize latency and communication costs while maximizing utility for citizens.

**Approach.** The blueprint shown in Figure 4 serves as a guideline to successfully implement citizen participation and engagement applications in smart cities based on four core building blocks. Citizens access the application using either Web-based front ends, or using native applications

deployed on smartphones or wearable devices (such as smart watches) that access the application through a managed API (provided by the API manager). These two components are deployed on the SCOS utilizing the runtime environment and they handle all citizen-facing interactions. Additionally, they rely on security and compliance to ensure that the critical compliance and privacy regulations are met, as well as on tenant management to respect the various different tenant requests. Last, they rely on infrastructure management to provide the necessary resources for elastically running them on cloud or edge infrastructures.

The data management component provides a unified and secure abstraction for access to data from multiple underlying smart city applications. To ensure this, it relies on storage and access for data integration into the smart city applications and city infrastructure. Additionally, it utilizes security and compliance to ensure no data regulations are infringed, as well as the infrastructure management to acquire the necessary resources. At the core of every citizen participation and engagement application is the custom application logic that actually provides added value and comfort to citizens. Applications can make use of a number of SCOS components to ease provisioning and management, as well as deployment and runtime.

The ideas we presented here are an essential stepping stone along the path of smart cities evolving towards the Internet of Cities, an open market place where applications can interact and be exchanged between cities. To enable these applications and to encourage a rapid adoption and deployment, it's essential to understand common problems and how to address them. Here, we outlined blueprints for three representative types of smart city applications, and identified

their key requirements, along with architectural guidelines to implement them based on our experience with SCALE. While smart city applications are diverse, there are multiple common aspects that should be present in a smart city application platform – primarily infrastructure and data management; security and compliance management; and a lifecycle and runtime environment for applications. Using an ecosystem such as SCALE, which incorporates these elements, lets practitioners efficiently and effectively implement complex smart city applications that are secure, scalable, and resilient. In our ongoing and future work, we plan to identify additional types of smart city applications and provide respective blueprints to address arising challenges. □

### References

1. J.M. Schleicher et al., "Enabling a Smart City Application Ecosystem: Requirements and Architectural Aspects," *IEEE Internet Computing*, vol. 20, no. 2, 2016, pp. 58–65.
2. J.M. Schleicher et al., "Towards the Internet of Cities: A Research Roadmap for Next-Generation Smart Cities," *Proc. 1st Int'l Workshop on Understanding the City with Urban Informatics*, 2015, pp. 3–6.

**Johannes M. Schleicher** is a PhD student at the Distributed System Group at TU Wien. His research interests are cloud computing, distributed systems, and smart cities. Contact him at [schleicher@dsg.tuwien.ac.at](mailto:schleicher@dsg.tuwien.ac.at); <http://dsg.tuwien.ac.at/staff/jjschleicher>.

**Michael Vögler** is a postdoctoral researcher at the Distributed System Group at TU Wien. His research interests are cloud computing, service-oriented architectures, distributed systems, and the Internet of Things (IoT). Contact him at [voegler@dsg.tuwien.ac.at](mailto:voegler@dsg.tuwien.ac.at); <http://dsg.tuwien.ac.at/staff/mvoegler>.

**Schahram Dustdar** is a full professor of computer science, and he heads the Distributed Systems Group at TU Wien. His work focuses on distributed systems. Dustdar is

an IEEE Fellow, a member of the Academia Europaea, an ACM Distinguished Scientist, and recipient of the IBM Faculty Award. He is on the editorial boards of *IEEE Internet Computing* and *Computer*. He's an associate editor of *IEEE Transactions on Services Computing*, *ACM Transactions on the Web*, and *ACM Transactions on Internet Technology*. He's the editor-in-chief of *Springer Computing*. Contact him at [\[dsg.tuwien.ac.at\]\(mailto:dsg.tuwien.ac.at\); <http://dsg.tuwien.ac.at/staff/sd/>.](mailto:dustdar@</a></p></div><div data-bbox=)

**Christian Inzinger** is a postdoctoral researcher at the software evolution and architecture lab (s.e.a.l.) at the University of Zurich. His main research focus is on helping developers write better cloud applications, and his research interests include architectures for cloud applications, software evolution, and

fault management in distributed elastic systems. Contact him at [inzinger@ifi.uzh.ch](mailto:inzinger@ifi.uzh.ch); [www.ifi.uzh.ch/seal/people/inzinger.html](http://www.ifi.uzh.ch/seal/people/inzinger.html).

*This article originally appeared in IEEE Internet Computing, vol. 20, no. 6, 2016.*

## Call for Articles

### IEEE Pervasive Computing

seeks accessible, useful papers on the latest peer-reviewed developments in pervasive, mobile, and ubiquitous computing. Topics include hardware technology, software infrastructure, real-world sensing and interaction, human-computer interaction, and systems considerations, including deployment, scalability, security, and privacy.

#### Author guidelines:

[www.computer.org/mc/pervasive/author.htm](http://www.computer.org/mc/pervasive/author.htm)

#### Further details:

[pervasive@computer.org](mailto:pervasive@computer.org)  
[www.computer.org/pervasive](http://www.computer.org/pervasive)

**IEEE** pervasive  
**COMPUTING**  
MOBILE AND UBIQUITOUS SYSTEMS



# The Best Software Development Teams Might Be Temporary

Rafael Prikladnicki, Marcelo Gattermann Perin, Sabrina Marczak, and Alessandra Costa Smolenaars Dutra

**COMPANIES SELECT** and recruit software development teams for specific projects. Once a project is over, the company can either keep using the same team for other efforts or break it up, assigning members to different tasks. Usually, companies keep the members together because of their shared expe-

riences, proven ability to deal with challenges and risks, and past effectiveness as a unit.

However, when studying how various configurations of teams working on Broadway shows affected performance, Brian Uzzi and Jarrett Spiro found that temporary teams might yield better results.<sup>1</sup> They said this

was related to the *small-world theory*, which claims that work teams generate better results via their networks of contacts and networking skills rather than their internal relationships with members with whom they've worked for a long time. In other words, being well connected is better than having a good previous experience with a team. This theory says that people generate an increasing number of connections by being part of clusters of networks, potentially creating more cohesion among individuals who might subsequently collaborate on projects.

We decided to study how software development teams are affected by whether members previously worked together. After analyzing projects at a large distributed software development company, we found that the best teams are often those whose members have never worked together. In other words, the best development teams might be temporary.

## Teams and Temporary Teams

A team is a small number of people with complementary skills who are committed to a purpose, performance goals, and an approach for



which they're mutually accountable. Different types of teams yield different levels of performance and effectiveness (see Figure 1).

Temporary teams engage in one or a few concrete, finite tasks to accomplish their goal. Long-term teams, on the other hand, often require multiple or repeated tasks to accomplish the many or recurring goals that were established at their inception or that evolve over time. The period defined as long-term varies with the team's task and goals.<sup>3</sup>

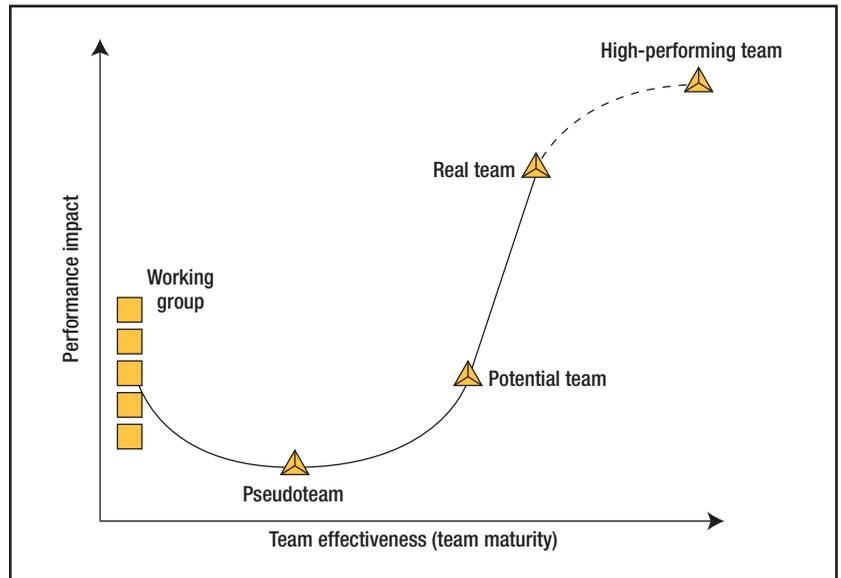
## Our Study

We investigated a US-based multinational, CMMI Level 3 IT company that develops software to support its business processes (such as online sales) and internal operations (such as employee career development). Its projects include new product development, software package customization, legacy system maintenance, and complex system integration. It has development centers in Brazil, India, and the US, as well as development operations in Canada, China, Ireland, Japan, Malaysia, and Russia.

The organization's main concern is providing software that satisfies its internal clients' expectations of quality, measured by how well the software meets specified business needs and how few defects it has. The company assigns each project to a globally distributed team that breaks up when the work is completed. The team members, who are either employees or contractors, use a large set of standardized tools.

## Research Methodology

We collected evidence on how members' prior work relationships affected the quality of 265 of our target company's software development projects



**FIGURE 1.** Team levels and team effectiveness.<sup>2</sup> Different types of organizational work teams provide varying levels of performance and effectiveness. For example, a working group will be less effective than a high-performing team.

in 2011. We adopted the company's own measures and variables to track project status and performance.

We considered four quality-related categories:

- the number of defects found in the development independent-testing, device-under-test, system-integration-testing, and user-acceptance-testing processes;
- the number of defects found in production;
- effort deviation, which is the percentage of effort spent executing a project beyond what was initially planned; and
- defects that caused either a critical issue with a crash or data loss, or a major issue with no workaround but no crash or data loss.

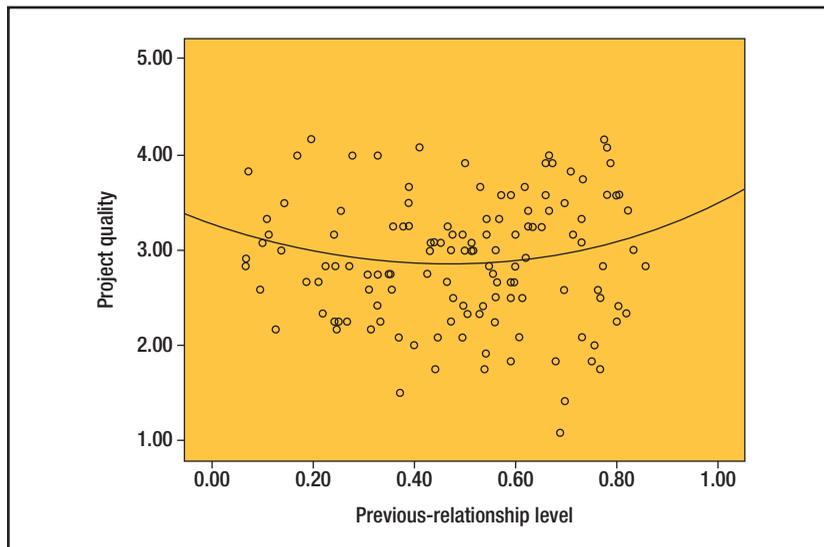
We used a scale of 1 to 5 to measure a project's performance in each category. A project's average score for

the four categories served as its quality indicator.

For each team, we identified how many members had worked together in any project from 2008 to 2010. We then calculated the team's overall previous-relationship level by dividing the number of pairs of members who participated in any prior project by the number of pairs that could be formed from all team members.

Our moderating variable was the software release type. *Major releases* introduced new features, functionality, and architectural improvements, as well as major cross-portfolio changes. They were identified by a change to the first digit of the firmware version number (X.y.z) and were released about every two years. They were usually more complex projects with longer development times.

*Minor releases* introduced new features and functionality and were identified by a change to the second



**FIGURE 2.** Project quality for major releases in relation to team members' previous working relationships. When fewer than half of the development team members had worked together previously, the quality dropped as the percentage of members who previously worked together rose. When more than 50 percent of members had worked together, quality increased as the percentage grew.

digit of the firmware version number (x.Y.z). They were released about every 6 to 12 months and were usually less complex projects with shorter development times.

For our regression analyses, we used four control variables. We extracted the number of hours spent on a project from the project repository. We counted the number of countries involved in the development process on the basis of the human-resources repository's employee-location data. We determined the number of days spent on a project by calculating the difference between the start and finish dates, including weekends and holidays, as recorded in the management repository. Finally, we identified the software development methodologies used during the project life cycle—agile, maintenance, traditional, data warehouse, and Oracle—on the basis of the target company's own definitions.

### Findings

For major releases, there was a U-shaped project-quality curve based on team members' previous relationship levels (see Figure 2). When fewer than half of team members had worked together previously, the project quality dropped as the percentage of members who worked together increased. When more than half the members had worked together, project quality rose as the percentage grew.

For minor releases, as the percentage of team members who previously worked together increased, the project quality decreased (see Figure 3). The better teams were those whose members had never worked together.

To verify the relationships in Figures 2 and 3, we performed a regression analysis. When we applied the quadratic term (squared) for the team members' previous-relationship

level, we verified the U-shaped project-quality curve. (To generate the quadratic term, we applied the mean-centered procedure.) As Table 1 shows, for major releases, the previous-relationship level had a significant positive effect on project quality ( $\beta = .161, p < .05$ ). But for minor releases ( $\beta = -.106$ ), a higher previous-relationship level had a negative linear effect on project quality ( $\beta = -.211, p < .05$ ).

**O**ur results suggest that for complex projects, it's best to have a mix of members who have and haven't worked together. For less complex projects, on the other hand, the best teams are those whose members haven't previously worked together. Although we can't generalize our data, our findings have important ramifications for development projects. The main ramification is that temporary teams should be the most effective.

Although old colleagues bring knowledge of the development process and prior norms from previous teams, new members bring fresh ideas that could promote project performance and creativity. Old colleagues might not do so and might not give new members a chance to implement their ideas. 

### Acknowledgments

Financial support for this research came from CNPq (the Brazilian National Council for Scientific and Technological Development) projects 312127/2015-4 and 406692/2013-0, FAPERGS (the Brazilian state of Rio Grande do Sul's Foundation for Research Support) project 2062-2551/13-7, and Brazilian Law 8.248/91.

**References**

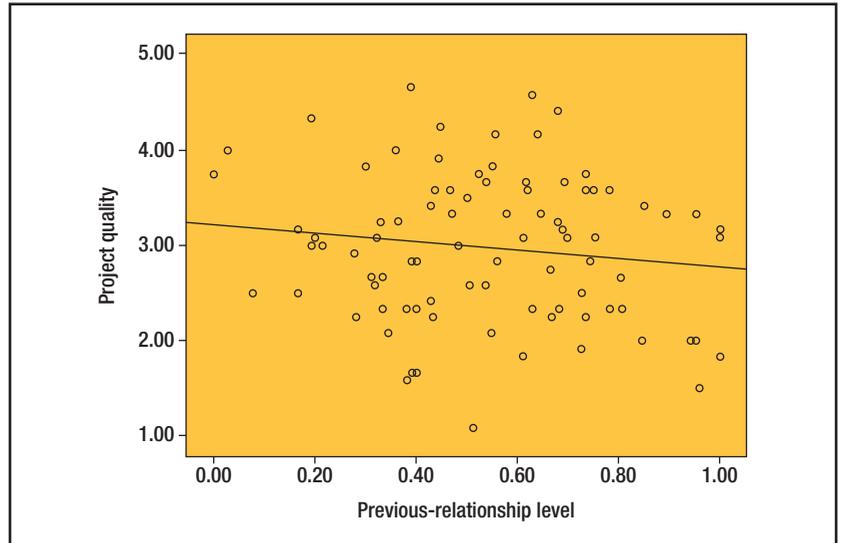
1. B. Uzzi and J. Spiro, "Collaboration and Creativity: The Small World Problem," *Am. J. Sociology*, vol. 111, no. 2, 2005, pp. 447–504.
2. J.R. Katzenbach and D.K. Smith, *The Wisdom of Teams: Creating the High-Performance Organization*, Harvard Business School Press, 1993.
3. P.C. Earley and E. Mosakowski, "Creating Hybrid Team Cultures: An Empirical Test of Transnational Team Functioning," *Academy of Management J.*, vol. 43, no. 1, 2000, pp. 26–49.

**RAFAEL PRIKLADNICKI** is an associate professor in the Computer Science School and director of the Science and Technology Park (Tecnopuc) at Pontifícia Universidade Católica do Rio Grande do Sul, where he also leads the MuNDDoS research group. He's on *IEEE Software's* editorial board and chairs the magazine's advisory board. Contact him at [rafaelp@pucrs.br](mailto:rafaelp@pucrs.br).

**MARCELO GATTERMANN PERIN** is a full professor in Pontifícia Universidade Católica do Rio Grande do Sul's Business School. Contact him at [mperin@pucrs.br](mailto:mperin@pucrs.br).

**SABRINA MARCZAK** is an associate professor in Pontifícia Universidade Católica do Rio Grande do Sul's Computer Science School. Contact her at [sabrina.marczak@pucrs.br](mailto:sabrina.marczak@pucrs.br).

**ALESSANDRA COSTA SMOLENAARS DUTRA** is an assistant professor in Pontifícia Universidade Católica do Rio Grande do Sul's Computer Science School. Contact her at [alessandra.dutra@pucrs.br](mailto:alessandra.dutra@pucrs.br).



**FIGURE 3.** Project quality for minor releases in relation to team members' previous working relationships. Project quality was worse for teams that had more members who had previously worked together.

**TABLE 1**

The results of regression analysis of statistics on the impact that development team members' prior experience working together had on project quality.\*

Control variables	Major releases	Minor releases
Actual effort (hours)	-.307 ( $p < .01$ )	-.364 ( $p < .01$ )
No. of countries in the development process	-.073	-.162
Total time (days)	.185	.160
Development method	.134	.115
Previous-relationship level	.057	-.211 ( $p < .05$ )
Previous-relationship level squared	.161 ( $p < .05$ )	-.106

\* The dependent variable was project quality.

This article originally appeared in *IEEE Software*, vol. 34, no. 2, 2017.





## The Roles of Code in Computational Science

Konrad Hinsén | Centre de Biophysique Moléculaire in Orléans

It's a safe bet that most readers of *CiSE* have written some code during their professional life. Many of us write code regularly as part of our scientific activity, perhaps even as a full-time job. But even though we write—and use—more and more code, we rarely think about the roles that this code will have in our research, in our publications, and ultimately in the scientific record.

In this article, I outline some frequent roles of code in computational science. These roles aren't exclusive; in fact, it's common for a piece of code to have several roles, at the same time or as an evolution over time. Thinking about these roles, ideally before starting to write the code, is a good habit to develop.

The most obvious role of code is its use as a tool for doing computations. In fact, it's this role that sets code apart from other types of information. In its tool role, code is evaluated using criteria such as suitability for a task, robustness, flexibility, performance, ease of use, and so on. Because this

role should be familiar to *CiSE* readers, I won't say much more about it and will concentrate on the other, less obvious roles.

### Code as a Scientific Notation

One important family of roles can be summarized as executable expressions of scientific information. In these roles, code can be considered a form of scientific notation. Criteria such as clarity of expression and exposition are therefore important, just like for any other form of scientific communication. However, executability matters as well, otherwise we would stick to more traditional notations such as plain language or mathematical formulas. Being able to execute the code is a proof of completeness. In an article, you can forget to define a quantity without anybody noticing for a long time. In a computer program, such an omission yields an error message. Executability is also a proof of absence of certain types of mistakes: the mistakes that a compiler can identify, and the mistakes that cause a program to fail during its execution.

The very first step in a research project is exploration: doing experiments and computations to better understand some phenomenon. In terms of scientific writing, this means lab notebooks and back-of-the-envelope calculations, meant for personal use or for sharing with close collaborators.

I call this a family of roles because very different kinds of scientific information can be expressed by code, and each kind requires its specific style, both for the code itself and for the documentation that accompanies it. In this respect, code isn't all that different from traditional scientific writing: a journal article and a textbook for students have different goals and are therefore written in very different styles. This analogy between code and scientific writing turns out to be useful for identifying different types of executable scientific knowledge.

The very first step in a research project is exploration: doing experiments and computations to better understand some phenomenon. In terms of scientific writing, this means lab notebooks and back-of-the-envelope calculations, meant for personal use or for sharing with close collaborators. At this stage, computations are performed interactively or by writing short scripts. But even interactively entered commands are code. Just like experimentalists keep lab notebooks, computational scientists should keep a detailed log of everything they type into their computers. Like all notes taken during exploration, this code must be cleaned up before publication.

Next comes publication. A paper in a scientific journal reports on a scientific study, explaining the context and motivation, the exact system being studied, and the methods that were applied. Results are then presented and discussed. The computational equivalent is a detailed record of all computations that were performed, including the software, the input data, and the results, presented in an understandable way.

Note that this detailed record isn't just code but code in its scientific context. This is true in general for the role of code as scientific notation. In fact, it's true as well for other scientific notations. A mathematical formula—or even a sentence in plain English—doesn't convey much meaning on its own. It acquires meaning only in the context of other formulas and sentences. Similarly, the code that was run in a computational research project makes sense only in the context of an overall description of that project.

This insight has led to the development of computational notebooks, pioneered by Mathematica ([www.wolfram.com/mathematica](http://www.wolfram.com/mathematica)) and more recently implemented for several programming languages by the Jupyter project (<http://jupyter.org>). Notebooks have become popular because they make it possible to combine the computational protocol used in a study with a scientific narrative into a single document. The reader of such a notebook not only can read the narrative but also can run the code, inspect intermediate results, and change parameters to see their impact on results. This ability to work with the code makes it easier to understand the scientific narrative, and vice versa.

Unfortunately, the term “notebook” has created a lot of confusion with lab notebooks, whose role is quite different, as I explained above. To make it worse, computational notebooks are also convenient tools for interactive work during the exploration phase. Finally, a published notebook is often the starting point for exploration of somebody else's work. From a tool perspective, computational notebooks have eliminated the borderline between exploration and publication. But in terms of scientific communication, the distinction still matters. Uploading notebooks containing interactive explorations to GitHub is definitely not sufficient for publishing code.

If journal articles were the only type of scientific document, we would have drowned in the ocean of millions of individual scientific studies long ago. What makes it possible to keep an overview of progress in a specific field is condensed summaries, which take the form of review articles. They provide a snapshot of the state of the art in a field that still undergoes rapid development. In the world of code, the equivalent of a review article is a library or program package that implements state-of-the-art algorithms with the goal of making them easily accessible to other researchers. Like the readers of a review article, the users of such a library are primarily experienced practitioners who can be expected to know the ideas behind the algorithms. However, precise and up-to-date technical documentation is essential: users must

be able to figure out easily what exactly the code does—in particular, what its limitations are.

Beyond review articles, there's a continuous consolidation process of scientific information whose final stage is textbook knowledge. Knowledge presented in textbooks is information that's well known and well understood by research practitioners from many specialty fields. The code equivalent of textbook knowledge is a library or program package that implements well-known and widely used algorithms in such a way that they can safely be used as black-box ingredients in research code. This requires particular attention to robustness. The code should behave reasonably even when used in an unforeseen way. Documentation must be written with nonexpert users in mind and should address both technical conditions and appropriate scientific contexts for the use of the code.

Unfortunately, we haven't yet found a good way to present libraries and program packages to human readers in their scientific context. Ideally, readers should see the code embedded in a discussion of the methods implemented. This discussion should be illustrated by example applications, and test cases should point out assumptions about the inputs, the handling of edge cases, and other subtleties. Literate programming<sup>1</sup> goes a long way toward this goal. Its basic premise is that software should be written as an essay for human readers, with markup that permits extracting pure code files for execution.

Note that literate programming is more complex than the computational notebooks I mentioned above. A notebook describes a linear sequence of computational steps with inputs and outputs alongside a scientific narrative. In contrast, algorithms and computational methods aren't linear. They don't have fixed inputs either—rather, they're designed to work with a wide range of inputs. This is perhaps one reason why literate programming hasn't been adopted yet by computational scientists. Another reason is the unsatisfactory state of support tools, which are incomplete and incompatible with each other. But the main reason is that the scientific community has only recently acknowledged the importance of the scientific notation role played by code.

The notation roles that I've discussed up to here deal with scientific models and methods, which have occupied the central stage of research for a few centuries. Today, data plays an ever more important role, and datasets are increasingly recognized as scientific publications in their own right. Most published datasets are raw or processed observational data, but it can also

be of interest to publish simulation output if regenerating it is particularly costly or difficult. Whatever the nature of the data, every electronic dataset has been created with the help of software, and that software's source code contains valuable information for interpreting the data. Ideally, the syntax and semantics of datasets should be well defined and documented (a topic I've written about before<sup>2</sup>), making it less important to have access to the code that generated the published files. However, just as descriptions of algorithms in plain language tend to be incomplete, descriptions of data models aren't always sufficient in practice. In the case of processed data, a detailed record of the processing steps in the form of the code that was actually run is also very helpful for interpretation. This role of code overlaps with the role of documenting a scientific study. The main difference is a focus on the process in one case compared to a focus on the results in the other. Another difference is that code that documents datasets should ideally be attached to the datasets as metadata and not require fetching another document from a different place.

### Code as the Shared Asset of a Scientific Community

A final role of code that deserves some discussion is its role in the structuring of scientific communities. Major program packages or libraries, but also more widely used infrastructure tools such as programming languages, have often been the nucleus for the formation of sizable user and developer communities, which, in the case of programming languages and infrastructure libraries, transcend the traditional scientific disciplines. The social impact of code has been much larger than for any other technical aspect of doing research.

I see the main reason for this in the enormous complexity of software. Keep in mind that when you run a 10-line Python script on your computer, you're really deploying a stack of software items that includes the Python interpreter, the libraries installed along with it, and an operating system such as Linux. Moreover, getting this software stack working on your computer has required other software that you may no longer see, such as a C compiler. The development, maintenance, and deployment of this enormous amount of code can't be handled by any individual or even a team of people working in close contact.

Consider a computational scientist working on the development of domain-specific code. This task requires a good understanding of the level below in the stack—a programming language, various libraries, and various tools such as compilers. Acquiring a useful level of competence with all of this is an important effort, so few people can

In practice, this means that the choice of a program package or a programming language implies the choice of a community one will interact with and maybe even become a part of.

become proficient in more than one or two such environments. The programming environment thus becomes an important aspect of one's work, and discussing this work is much easier with people who share the same background. Collaborating on code development imposes even stricter compatibility: all developers must be proficient using almost identical environments. In comparison, traditional scientific instruments are simple. A scientist familiar with the principles of, for example, electron microscopy can rapidly learn to use any particular electron microscope and has no difficulty in talking to or collaborating with colleagues who use different models. Likewise, a theoretician can easily read articles written by scientists from a different school that favors different mathematical notations or uses different definitions.

In practice, this means that the choice of a program package or a programming language implies the choice of a community one will interact with and maybe even become a part of. But every community also has shared values and attitudes that aren't directly related to its shared technologies. One community might be conservative, valuing long-term compatibility over progress, whereas another might like its code on the bleeding edge. Moving even further away from technical aspects, communities can be more or less welcoming to newcomers and more or less open to new ideas. At the extreme, communities can have sexist, racist, or elitist attitudes that can be serious obstacles to productive research work.

The social roles of code are likely to gain in importance as code itself is more and more recognized as important in research. At this time, publishing code has become common though not yet universal. But unlike journal articles, code isn't reviewed in the publication process. One obstacle to independent code reviewing is that a potential reviewer must have both the right scientific domain knowledge and solid technical competence in the development technologies employed by the code authors. If reviewing code becomes common, this will likely create social pressure toward standardization of such technologies inside each scientific domain.

### Tensions between the Roles

The different roles that a piece of code has aren't independent. On the contrary, they can easily be

in conflict, requiring scientific software developers to find a workable compromise. This is why I recommend considering the roles of your code before starting to write it. Compromises tend to be better when they're the result of conscious decisions rather than accidents of development.

An often-discussed tension is the one between performance in the tool role and clarity in the notation role. Performance optimization tends to make code less understandable. In the extreme, for example, when parallelizing code at a low level using MPI, it can become very hard to figure out what the optimized code does exactly. It's possible that better notations—that is, new programming languages designed from the start to act like scientific notations—can alleviate this tension, but for the moment, this is just wishful thinking.

Another important conflict exists between the generality that's desirable for the tool and social roles on one hand and the requirement for simplicity in the notation role on the other. Tools applicable to a wide range of problems are advantageous for their users, who can get away without learning too many tools. They're also advantageous for their developer communities because these communities can recruit members from a wider base and thus share the development work among more people. However, general tools lead to larger and more complex code bases that are usually more difficult to understand than simple code written specifically to solve one problem.

A related tension stems from the reliance on other code, in particular, libraries, something that we've discussed in this department before as well.<sup>3</sup> From the viewpoint of the tool role, using existing library code, assuming it's of good quality, is usually an advantage. In the social role, code with library dependencies creates asymmetric links between two communities, which has good and bad sides. From the notation viewpoint, dependencies on libraries implementing well-known techniques is an advantage, as readers are likely to be already familiar with them. But dependencies on new or little-known libraries make understanding the code more difficult for the average reader because there's more code to read—in fact, this is a special case of the generality dilemma.

### Interactions between Different Categories of Code

Another aspect to consider in writing code that fulfills its intended roles well is that no piece of scientific code exists in isolation. The computational protocol of a specific study relies on program packages and libraries of varying degrees of maturity, and each program package and library adds dependencies of its own.

From the tool perspective, the computational protocol has almost no value because its work is done. It might serve as an inspiration for the development of other tools, but it's unlikely to be reused in its entirety. Consequently, there's no point in doing any maintenance. In contrast, the domain-specific or general libraries it depends on will be used by larger communities for a longer time. They'll be maintained and extended, but that also means that they undergo permanent change.

This poses a problem from the notation perspective, which requires that a published computational protocol should remain reproducible for as long as the study itself remains of scientific interest, which can easily be several decades. This is the core issue of the reproducibility problem, which has come up repeatedly in *CiSE*, in particular in two theme issues (published in January/February 2009 and July/August 2012). In the past, the tool perspective has dominated scientific software, with the result that individual computational protocols are effectively built on quicksand foundations consisting of tools that change through continuous improvement. The increasing emphasis on reproducibility—and the ongoing tendency to publish scientific software—is thus likely to create pressure on stability at least in widely used libraries and program packages.

A final aspect to consider is the evolution of roles in time. From the notation perspective, it's desirable to write new code as the status of a computational method advances from innovative through established to well known. The requirements for these respective roles are too different to be assumed by a single code base. The analogy with scientific writing is useful again: it would be ludicrous to compose a review article from pasted-together sections of original research articles. But scientific code is still rarely written with its notation role in mind. From the tool viewpoint, reusing working code may well seem preferable, and that's also the way of least immediate effort.

Readers waiting for a final coherent message from this article will be disappointed: I can't offer any simple conclusion. The only advice I can

offer is to take all the issues I've outlined into account when starting a coding project. Consider why you're writing the code right now and for how long you expect to use it. Consider who else might use the code in the foreseeable future and who might collaborate with you on its future development. Do you plan to publish it? This might actually not be a choice for much longer, so better prepare to answer "yes."

For any code meant to be published, the notation roles should be explored carefully. As with any form of scientific communication, you have to write for your audience. For a code base, that audience consists of the users of the code but also consists of the audience that its users are addressing with their own work. If you want me to use your code, you have to make it easy for me to explain to my readers what your code does.

The social role of code is particularly important to consider when building a community around a software package is your explicit goal. But even if you write small and specialized software that's unlikely to ever become widely used, it's worth thinking about how your code will be judged by the user community of your development and infrastructure tools, if only because in the not-too-remote future this judgment might become part of the scientific reviewing process. Adopting the best practices of this community is the equivalent of making an effort to write your papers in good English—something that I haven't heard anyone argue against yet. ■

### References

1. D.E. Knuth, "Literate Programming," *Computer J.*, vol. 27, no. 2, 1984, pp. 97–111.
2. K. Hinsén, "Caring for Your Data," *Computing in Science & Eng.*, vol. 14, no. 6, 2012, pp. 70–74.
3. M. Turk, "Vertical Integration," *Computing in Science & Eng.*, vol. 17, no. 1, 2015, pp. 64–66.

**Konrad Hinsén** is a researcher at the Centre de Biophysique Moléculaire in Orléans and at the Synchrotron Soleil in Saint Aubin. His research interests include protein structure and dynamics and scientific computing. Hinsén has a PhD in theoretical physics from RWTH Aachen University. Contact him at [konrad.hinsen@cnrs.fr](mailto:konrad.hinsen@cnrs.fr).

*This article originally appeared in  
Computing in Science & Engineering,  
vol. 19, no. 1, 2017.*



# Move Your Career Forward

## IEEE Computer Society Membership

### Explore These Software Development Resources



#### *IEEE Software*

The authority on translating software theory into practice, this bimonthly magazine presents pioneering ideas, expert analyses, and thoughtful insights to help software professionals keep up with rapid technology change.



#### *IEEE Transactions on Software Engineering*

This bimonthly journal focuses on well-defined theoretical results and empirical studies with real potential to impact software construction, analysis, and management.

### Software Development Associate Engineer Certification

This certification is designed to recognize candidates who have acquired the basic knowledge and understanding required for developing software products. It requires a coherent and demonstrable understanding of the principles and processes involved in software requirements, design, construction, and testing.

### Professional Software Developer Certification

This certification is for experienced professional software developers with demonstrated skill and competency in the field. Part I tests competency in four different software knowledge areas: Requirements, Design, Construction, and Testing. Part II requires successful completion of two applied modules.

**FOR DIRECT LINKS  
TO THESE RESOURCES, VISIT**

[www.computer.org/edge-resources](http://www.computer.org/edge-resources)

IEEE  computer society

*The Community for Technology Leaders*



# The Social Responsibility of Software Development

Diomidis Spinellis

**LIFE IN A** Nazi concentration camp must have been horrendous. As an inmate, you'd almost certainly die in agony, perhaps after witnessing the tragic demise of your family and friends. As a guard, you might get to live, but with the death and suffering of millions in your conscience and your nation's collective conscience. You might think that these horrors are behind us. Think again. Humanity can advance from the unthinkable to the inevitable in small, often imperceptible, steps. You start by dismissing a Great War veteran delivering anti-Semitic polemics as an irrelevancy, you continue to ignore him when he wants to be called Führer, and you end up witnessing the industrial-scale slaughter of people.

Our world is increasingly running on software. As a software developer, for better or worse (it's up to you), you're building the fabric of our society, tomorrow's world. You're therefore directly responsible for averting some of those small imperceptible steps to future nightmares. Many of the things you do have ethical, social, and political implications.

Consider the following often-discussed examples. The filters you build in a social networking platform might create digital bubbles that reinforce a group's beliefs instead of allowing it to see others' opinions. This polarizes society and spreads hatred. Car software might protect the interests of the car manufacturer you work for, against the law, the environment, or the car's occupants. The dating site's algorithms you design might increase racial segregation and social-class divisions. (You get to play God with the human gene pool.) Your poorly designed user interface can cause people hardship, while your incorrect code can harm human life or damage property. The console game you create might be purposely designed to be highly addictive. How you route job ads can influence disadvantaged groups' work opportunities.

On a larger scale, new digital platforms, products, and services might disrupt the livelihood or working conditions of millions. Also, machine-learning algorithms might promote discrimination or produce invalid results in ways that

are difficult to detect and analyze. Governments are often no better than companies. Digital government systems often operate in a void of democratic accountability and might lack sufficient recourse opportunities. The requirements and design of a system you build might subvert public policy regarding transportation, education, or law enforcement. It gets worse. The US digital mass surveillance program and discussion regarding enhanced border controls or China's evolving social-credit system might be bellwethers of a totalitarian dystopia.

## From Learning to Participating

Is it realistic for you to worry about the problems I outlined? Technological progress is unstoppable and mostly welcome; thanks to it, millions around the world are healthier, wealthier, and better educated. Technology inevitably brings disruption. However, the balance between its benefits and perils is often obvious. For example, self-driving cars might displace drivers, but their widespread adoption could drastically decrease the 1.3 million deaths in road crashes each year and the 20 to 50

million people who end up injured or disabled. On the other hand, there can be no justification for developing cyberwarfare software to breach a dam's floodgates upstream of a large city. Given this complex, perilous state of affairs, what should you do? Here are some ideas.

### Learn

Being a brilliant Scrum master might let you build the software right but won't ensure that you develop the right software. Read up on history to appreciate the painful and often tragic path of our society's evolution. Did you know that in the 1940s, punch card technology was used to organize Nazi concentration camps and US internment camps for Japanese Americans? Keep abreast with current events and discussion—not through 140-character tweets, but with long readings that offer insightful analysis. Acquaint yourself with the Software Engineering Code of Ethics and Professional Practice. Study subjects besides technology: philosophy, political science, sociology, ethics, and the arts. Remember: technocratic judgment can get you only so far.

### Think

Contemplate the wider repercussions of what you're working on. What will happen when your shiny prototype gets widely deployed or stops being maintained? Will your new web-based service disadvantage a particular minority? What if all the data you're gathering falls into the hands of organized crime or a totalitarian government? Is the cyberweapon you're developing more likely to be used for defense or for offensive actions and terrorism that can hurt countless civilians? What if criminals can gain control of the law-enforcement backdoor access

you're providing? How will your new social-interaction feature affect children or families?

### Act

Walking the walk is probably the most difficult part of acting responsibly as a software developer. Strive to design and implement the software you work on so that it becomes a force for good rather than evil. Don't work for organizations whose mission is intrinsically detrimental to society. If your organization develops software that will harm society, speak up. Present your case to coworkers (sometimes engineers become blinded and seduced by the technology's power), and offer better alternatives. Don't turn whistleblower on impulse: seek alternatives, deliberate, and think through your actions. Also, each time your fancy software makes people redundant, consider the words of Hannah Arendt: "Radical evil has emerged in connection with a system in which all men have become equally superfluous."<sup>1</sup> Could your software instead make those people's work more meaningful?

I hear you saying that limiting your employment options or stirring up trouble in the workplace isn't realistic, when day in and day out you've got to bring bread to the table. However, the choice of where you work and what you do is rarely binary. There are thousands of choices between writing code to streamline operations at Médecins Sans Frontières (Doctors without Borders) and implementing process control software for the chemical factory that supplied sarin nerve gas to Bashar al-Assad. Depending on your personal circumstances and drive, you can certainly find ways to gravitate toward the positive end of the spectrum I outlined. By depriving



## Call for Articles

*IEEE Software* seeks practical, readable articles that will appeal to experts and nonexperts alike. The magazine aims to deliver reliable, useful, leading-edge information to software developers, engineers, and managers to help them stay on top of rapid technology change. Topics include requirements, design, construction, tools, project management, process improvement, maintenance, testing, education and training, quality, standards, and more.

Author guidelines:

[www.computer.org/software/author](http://www.computer.org/software/author)

Further details: [software@computer.org](mailto:software@computer.org)

[www.computer.org/software](http://www.computer.org/software)



**PURPOSE:** The IEEE Computer Society is the world's largest association of computing professionals and is the leading provider of technical information in the field. Visit our website at [www.computer.org](http://www.computer.org).

**OMBUDSMAN:** Email [ombudsman@computer.org](mailto:ombudsman@computer.org).

**Next Board Meeting:** 12–17 June 2017, Phoenix, AZ, USA

#### **EXECUTIVE COMMITTEE**

**President:** Jean-Luc Gaudiot

**President-Elect:** Hironori Kasahara; **Past**

**President:** Roger U. Fujii; **Secretary:** Forrest Shull; **First VP, Treasurer:** David Lomet; **Second VP, Publications:** Gregory T. Byrd; **VP, Member & Geographic Activities:** Cecilia Metra; **VP, Professional & Educational Activities:** Andy T. Chen; **VP, Standards Activities:** Jon Rosdahl; **VP, Technical & Conference Activities:** Hausi A. Müller; **2017–2018 IEEE Director & Delegate Division VIII:** Dejan S. Milojić; **2016–2017 IEEE Director & Delegate Division V:** Harold Javid; **2017 IEEE Director-Elect & Delegate Division V-Elect:** John W. Walz

#### **BOARD OF GOVERNORS**

**Term Expiring 2017:** Alfredo Benso, Sy-Yen Kuo, Ming C. Lin, Fabrizio Lombardi, Hausi A. Müller, Dimitrios Serpanos, Forrest J. Shull

**Term Expiring 2018:** Ann DeMarle, Fred Douglas, Vladimir Getov, Bruce M. McMillin, Cecilia Metra, Kunio Uchiyama, Stefano Zanero

**Term Expiring 2019:** Saurabh Bagchi, Leila De Floriani, David S. Ebert, Jill I. Gostin, William Gropp, Sumi Helal, Avi Mendelson

#### **EXECUTIVE STAFF**

**Executive Director:** Angela R. Burgess; **Director, Governance & Associate Executive Director:** Anne Marie Kelly; **Director, Finance & Accounting:** Sunny Hwang; **Director, Information Technology & Services:** Sumit Kacker; **Director, Membership Development:** Eric Berkowitz; **Director, Products & Services:** Evan M. Butterfield; **Director, Sales & Marketing:** Chris Jensen

#### **COMPUTER SOCIETY OFFICES**

Washington, D.C.: 2001 L St., Ste. 700,

Washington, D.C. 20036-4928

**Phone:** +1 202 371 0101 • **Fax:** +1 202 728 9614

**Email:** [hq.ofc@computer.org](mailto:hq.ofc@computer.org)

**Los Alamitos:** 10662 Los Vaqueros Circle, Los Alamitos, CA 90720 • **Phone:** +1 714 821 8380 •

**Email:** [help@computer.org](mailto:help@computer.org)

**Membership & Publication Orders**

**Phone:** +1 800 272 6657 • **Fax:** +1 714 821 4641 •

**Email:** [help@computer.org](mailto:help@computer.org)

**Asia/Pacific:** Watanabe Building, 1-4-2 Minami-

Aoyama, Minato-ku, Tokyo 107-0062, Japan •

**Phone:** +81 3 3408 3118 • **Fax:** +81 3 3408 3553 •

**Email:** [tokyo.ofc@computer.org](mailto:tokyo.ofc@computer.org)

#### **IEEE BOARD OF DIRECTORS**

**President & CEO:** Karen Bartleson; **President-**

**Elect:** James Jefferies; **Past President:** Barry L.

Shoop; **Secretary:** William Walsh; **Treasurer:**

John W. Walz; **Director & President, IEEE-USA:**

Karen Pedersen; **Director & President, Standards**

**Association:** Forrest Don Wright; **Director & VP,**

**Educational Activities:** S.K. Ramesh; **Director & VP,**

**Membership and Geographic Activities:** Mary

Ellen Randall; **Director & VP, Publication Services**

**and Products:** Samir El-Ghazaly; **Director & VP,**

**Technical Activities:** Marina Ruggieri; **Director**

**& Delegate Division V:** Harold Javid; **Director &**

**Delegate Division VIII:** Dejan S. Milojić

revised 26 Jan. 2017



evildoers of your talent, you contribute to everyone's well-being.

Also, if you see that your organization develops harmful software, try bringing constructive ideas to the table. You can argue that in the medium or long term, being socially responsible might benefit your organization. Or, you can come up with win-win options: software that's good for both your organization and society.

### **Educate**

Modern software is mind-numbingly complex, and it often interacts with our world in subtle, unanticipated ways. As a software developer, you might understand these things in both the abstract and concrete, but your colleagues from other backgrounds and the public at large might not. Help by uncovering, explaining, and publicizing any issues you recognize. Build bridges with people in other domains, such as medical professionals, who might comprehend the domain's ethics but might not realize the potential, choices, and ethical issues hidden underneath the technology. Discuss with your colleagues, post in social networks, write articles, and take a stance in daily interactions. As an educated professional, you have the duty to return to society part of the education it has endowed you with. And, if you're an educator, strive to include the topics in your lectures and the curriculum.

### **Connect**

Strengthen your voice by joining it with the voices of others. Engage with your professional societies to raise awareness of these issues and develop defense and support mechanisms so that your colleagues who speak up aren't left out in the cold. Mingle with people outside your (probably privileged) circle: those

*This article originally appeared in IEEE Software, vol. 34, no. 2, 2017.*

who struggle to make ends meet, those affected by your work, and scared underdogs. Write free software that could help our world become a better place. Work with think tanks, civil-society organizations, and political parties to draft and promote sensible policies. Volunteer for office and, it should go without saying, get out and vote!

Someone might argue that as software developers, we should focus just on doing our assigned work as best as we can because an organization can't function if all its members constantly question its operations. Instead, this argument goes, we should let the market decide or the government regulate what software is developed and how it's used. I disagree. This argument degrades our working existence to that of a robot. Part of being human is making moral choices in everything we do: from the detergent we buy, to the transportation we use to get to work, to the software we develop. If our children ask us in a few years what we were doing when the lights went dark, responses such as "I didn't know these horrors were happening" or "I was just doing my job" ("I was just following orders") won't cut it. 🙄

### **Acknowledgments**

I thank Felienne Hermans, Panos Louridas, Peter G. Neumann, Nancy Pouloudi, Eberhard Wolff, and Alexios Zavras for their constructive feedback on an earlier version of this piece. The opinions expressed here are mine; no endorsement by the IEEE, its officials, or its members is implied. As always, I welcome discussion and rebuttals.

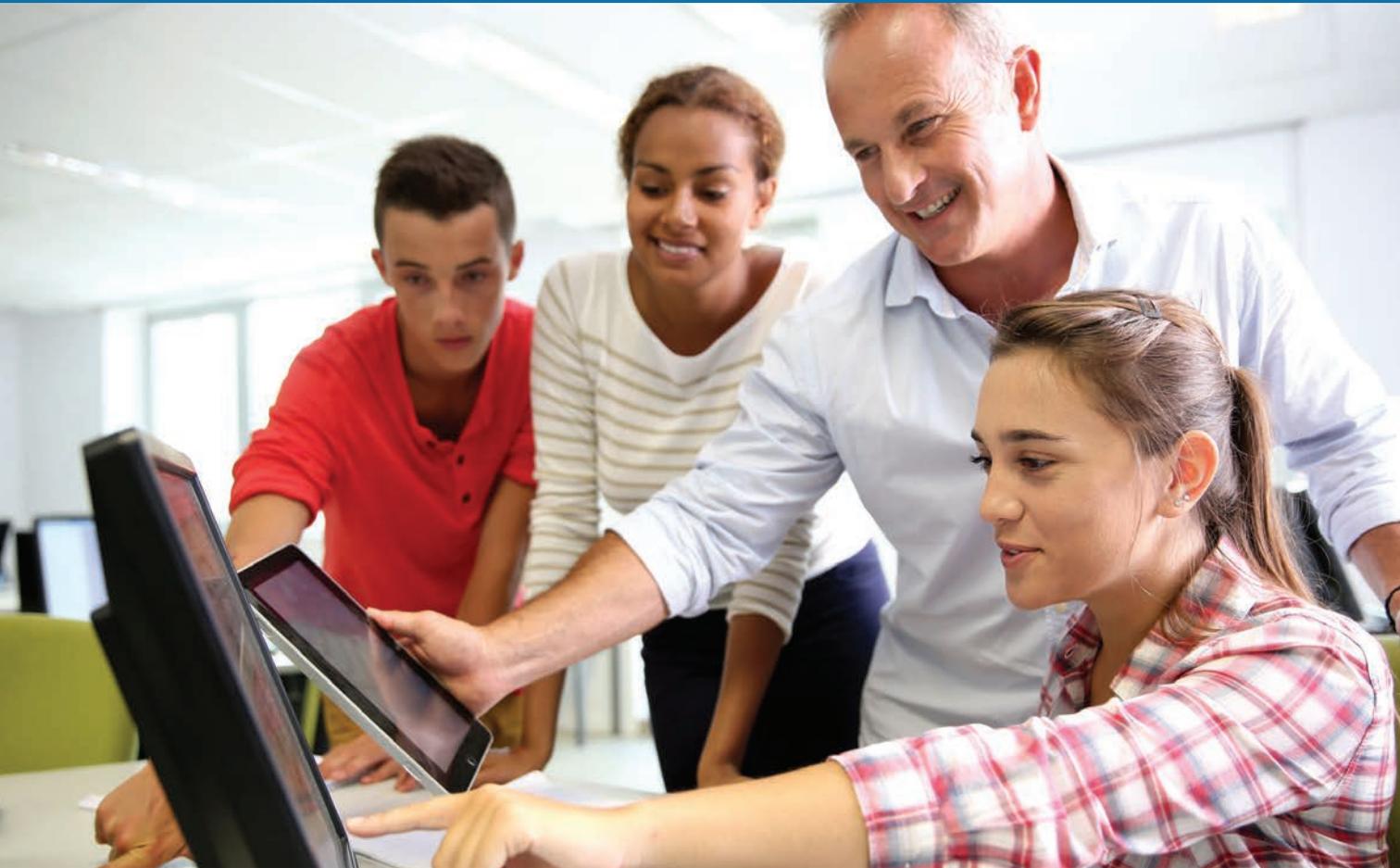
### **Reference**

1. H. Arendt, *The Origins of Totalitarianism*, Harcourt, 1973.



# CALL FOR NOMINEES

## Education Awards Nominations



### Taylor L. Booth Education Award

A bronze medal and US\$5,000 honorarium are awarded for an outstanding record in computer science and engineering education. The individual must meet two or more of the following criteria in the computer science and engineering field:

- Achieving recognition as a teacher of renown.
- Writing an influential text.
- Leading, inspiring or providing significant education content during the creation of a curriculum in the field.
- Inspiring others to a career in computer science and engineering education.

Two endorsements are required for an award nomination.

See the award information at:  
[www.computer.org/web/awards/booth](http://www.computer.org/web/awards/booth)

### Computer Science and Engineering Undergraduate Teaching Award

A plaque, certificate and a stipend of US\$2,000 is awarded to recognize outstanding contributions to undergraduate education through both teaching and service and for helping to maintain interest, increase the visibility of the society, and making a statement about the importance with which we view undergraduate education.

The award nomination requires a **minimum of three endorsements**.

See the award details at:  
[www.computer.org/web/awards/cse-undergrad-teaching](http://www.computer.org/web/awards/cse-undergrad-teaching)



Deadline: 15 October 2017  
Nomination Site: [awards.computer.org](http://awards.computer.org)





# The Means of Production

David Alan Grier, George Washington University

*Technical leadership remains one of the great problems of software production.*

In the study of software development, Ralph Flanders is a minor historical figure who's connected to agile methods only through a long chain with many rusted and incomplete links. Generally known as a business leader and politician, his moment of glory came in 1954 when he introduced the motion into the US Senate that would censure his colleague Senator Joe McCarthy. Yet, early in his career, after an apprenticeship as an industrial engineer, he identified several key problems that hindered industrial production, which are the same problems that hinder software development.

The connection between industrial engineering and software engineering is strong. Early software developers, including John von Neumann's colleague Herman Goldstine, shaped their early notions of software with concepts from industrial production. The word "program" came from the term that described a plan for factory operation. Flow charts came from the diagrams that traced the movement of material through a plant. Sequencing originally referred to the steps of factory operation. The terms "program quality" and the "software life cycle" were borrowed from the work to develop production plans that would create uniform goods.

Flanders did most of his engineering work in the 1920s, when Henry Ford was

the dominant industrial engineer. Ford argued for production methods that placed workers under strict production control: each worker had one place on the production line, did one job repeatedly, and learned how to do that job efficiently. Flanders argued that this approach was ill-suited to the production of many products. He claimed that the most challenging problems of manufacturing were not inefficient workers but instead manufacturing overhead. Flanders wrote that many plants saw only "a moderate increase in output" when they switched to Ford's methods, and yet these plants saw a great increase in overhead, particularly in "the number of foremen and in the amount of clerical labor they were called upon to do."

Flanders made recommendations that are familiar to anyone who has followed recent trends in lean and agile methodologies. He urged manufacturers to organize production by product, not by process, and to design their product to be as simple as possible. He suggested that factories use small teams and encourage team members to develop a wide and comprehensive set of skills. Finally, he recommended that groups work as independently as possible so that the group leader could supervise the work rather than attend to meetings, memos, and reports. "The foreman is bounded only by his production orders and his schedule," Flanders stated; within "these limits he is king of his territory."

Flanders appears to have had limited influence as an engineer, though

his ideas might have been thwarted first by the reduced production of the Great Depression and second by the overwhelming demands of World War II. However, his concern for technical leadership echoed throughout the engineering literature: it was picked up by management consultant Peter Drucker in the 1950s, Russian engineer S.P. Mitrafanov in the 1960s, and software engineer Frederick P. Brooks in the 1970s. A "major part of the cost [of software] is communication and correcting the ill effects of miscommunication," Brooks wrote in his seminal work on software production. Hence, he concluded, computer software needs to be built by "as few minds as possible."

Technical leadership remains one of the great problems of software production, for it requires a different set of skills for conventional mid-level management, skills that handle the tasks of planning, allocating, and coordinating assets. Technical leaders need to understand the details of their product, teach key skills to their workers, and assess the output of their team. Excessive communication, coordination, and reporting reduce the efficiency of these leaders—often fatally so.

We should care about Flanders because he cared about technical leadership long before it was fashionable to have such concerns. Like many who followed him, he recognized that such leadership was difficult and that we could make it easier only by finding better ways of organizing the means of production. ■

DAVID ALAN GRIER is an associate professor at George Washington University. Contact him at grier@gwu.edu.

*This article originally appeared in Computer, vol. 49, no. 9, 2016.*

Now there's  
**even more to  
love about your  
membership...**

Read all your IEEE Computer Society  
magazines and journals your**WAY** on

myCS

**NO  
ADDITIONAL  
FEE**



- ▶ ON YOUR COMPUTER
- ▶ ON YOUR SMARTPHONE

- ▶ ON YOUR eREADER
- ▶ ON YOUR TABLET

**Introducing myCS, the digital magazine portal from IEEE Computer Society.**

Finally...go beyond static, hard-to-read PDFs. Our go-to portal makes it easy to access and customize your favorite technical publications like *Computer*, *IEEE Software*, *IEEE Security & Privacy*, and more. Get started today for state-of-the-art industry news and a fully adaptive experience.



▶ LEARN MORE AT: **mycs.computer.org**

# Before Torchi and Schwilgué, There Was White

Denis Roegel  
University of Lorraine

Editor: David Walden

This brief note pushes further back the invention of key-driven calculating machines. Until recently, it was thought that the first such machine was Du Bois D. Parmelee's in 1850. Then, notice was made of Jean-Baptiste Schwilgué's machine (1844), then of Luigi Torchi's (1834), and now of James White's (1822).

## Schwilgué (1844) and Torchi (1834) Machines

In 2008, I published an article describing a machine that I thought was the first key-driven calculating machine (see Figure 1), patented by Schwilgué in 1844. The Schwilgué machine predated the earliest then known such machine, which was Parmelee's machine from 1850.<sup>1</sup> The first version of my article had the title "The First Key-Driven ...," and the editors were wise in suggesting a change to "An Early (1844) Key-Driven ..."

Shortly after the publication of my article, I came into contact with Silvio Hénin, who had discovered that Luigi Torchi had also built a key-driven calculating machine, but 10 years earlier.<sup>2</sup> I was then quite grateful to the editors of the *Annals* for their suggestion! Hénin's article was published in 2010, and since then, little has changed. There is no known copy of Torchi's machine. And although Schwilgué was not the first to construct such a machine, he manufactured a number of them, and several of them have survived, making them the currently oldest key-driven calculating machines dated with certainty.<sup>3</sup>

Since then, another machine by Schwilgué, probably from the early 1840s, has also surfaced.<sup>4</sup>

## And Then White (1822)

Some time ago, I was going through some old photocopies of White's 1822 book *New Century of Inventions*,<sup>5</sup> and I was quite surprised to see that White had described an adding machine on pages 343–348 and included two plates. This machine happened to be key-driven. My photocopies had been lying silently around for more than 10 years and were not used for the article published in 2008. Not too much is known of James White (1762–1825), but he was a prolific inventor. He designed in particular several types of gears, of which some models are kept at the Conservatoire National des Arts et Métiers in Paris.<sup>6</sup>

White's adding machine is basically a column adder, and it follows an interesting construction, which is not found in the machines that were to become ubiquitous later. We cannot be absolutely sure that the machine was ever built, but given that White had been inventing and

patenting machines since the end of the 1780s, it is reasonable to assume that a prototype of this adding machine existed at one time. Even if it didn't, White's description proves that the idea of using keys had occurred to him no later than 1822.

## The Keys of White's Machine

White's adding machine has nine keys, arranged compactly in two rows. Figure 2 shows that the first four digits 1, 2, 3, and 4 are on the lower line (and from right to left), and the other five digits are on the upper line (and from left to right). Each key ends with a pulley that presses on an endless chain. The chain, of course, is not elastic, but it is stretched around the wheels A, B, C, and D, and the center of C lies on a spring F. Consequently, the chain can be loosened, and pressing keys will not break it. Moreover, the chain gears in the wheels B and D. These two wheels each have a small pawl in order to force them to turn in the indicated direction.

Each key can be pressed more or less, depending on the number it is to produce. When key 1 is pressed to the bottom, the chain will turn the wheel B by one tooth, but wheel D will not rotate. When the key is released, wheel B will not turn, but wheel D will. Eventually, the wheel B counts the sum of the pressed figures.

This construction makes it possible to type in several figures at the same time. However, there is no way to ensure that the keys are pressed to the bottom, which is something that some later machines will be able to enforce, for instance, by adding the figures only upon the release of a key and not upon its pressing.

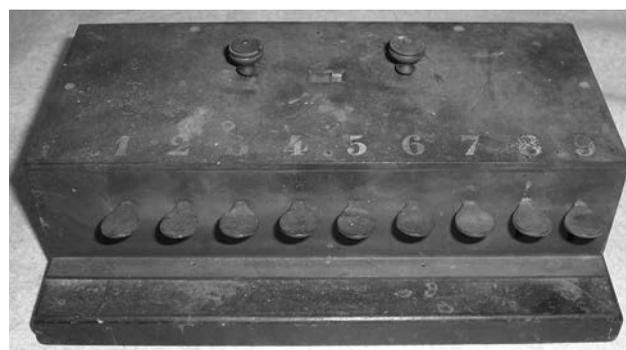


Figure 1. A copy of Schwilgué's small adding machine, patented in 1844. (Strasbourg Historical Museum, photograph by Denis Roegel)

### Carrying in White's Machine

Figure 3 of White's first plate (upper left in Figure 2) shows the principle of carrying. One wheel carries a pin that, when rotated, advances another wheel by one tooth. The position of the moved wheel is secured by a spring ED.

In addition to this basic construction, White introduced a "floating point" mechanism, in that the motion of wheel B (top of Figure 2) rotates a square shaft BG on which lies a sliding wheel I. This wheel can be moved to the appropriate position, should we add units, tens, hundreds, and so on, or even more arcane units such as those in use at the time of White. Another plate, not reproduced here, considers a special wheels for adding farthings, pence, shillings, and so forth.

### Conclusion

White's *New Century of Inventions* is a book full of treasures, but for some reason, his adding machine was overlooked for almost two centuries. White's machine now pushes back the invention of the first key-driven adding machine at least to 1822, although we do not know if this machine really worked. Hopefully, this late discovery will spur others in the little known works of other inventors.

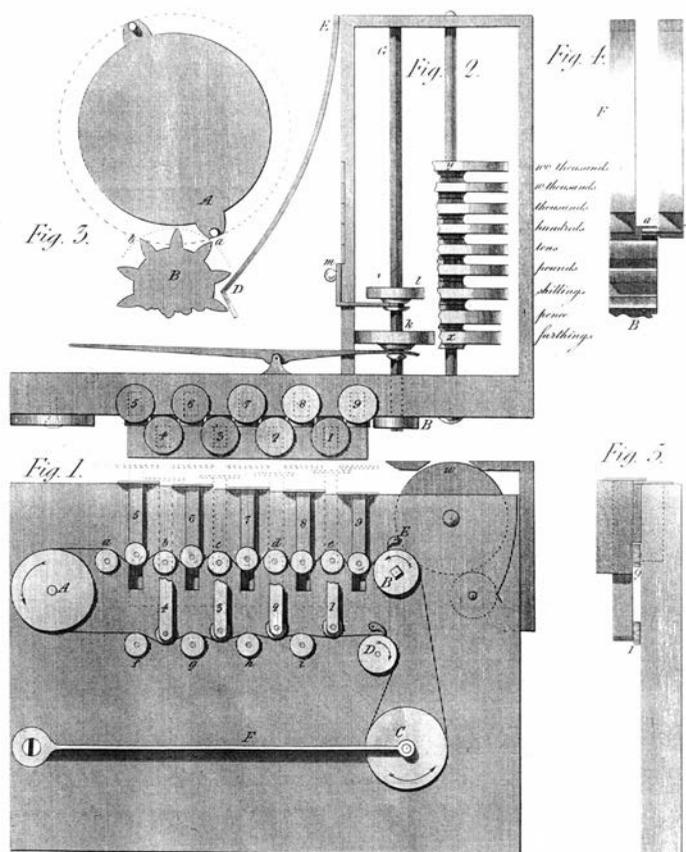


Figure 2. James White's main plate for a key-driven adding machine. This illustration appeared in his 1822 book *New Century of Inventions*.<sup>5</sup>

### References and Notes

1. D. Roegel, "An Early (1844) Key-Driven Adding Machine," *IEEE Annals of the History of Computing*, vol. 30, no. 1, 2008, pp. 59–65. A short addenda to this article appears in *IEEE Annals of the History of Computing*, vol. 30, no. 3, p. 3.
2. S. Hénin, "Two Early Italian Key-Driven Calculators," *IEEE Annals of the History of Computing*, vol. 32, no. 1, 2010, pp. 34–43.
3. D. Roegel, "An Overview of Schwilgué's Patented Adding Machines," *Bull. Scientific Instrument Soc.*, vol. 126, 2015, pp. 16–22.
4. D. Roegel, "A New Early Adding Machine by Schwilgué (c. 1840?)," *Bull. Scientific Instrument Soc.*, vol. 130, 2016, pp. 24–27.
5. J. White, *A New Century of Inventions: Being Designs and Descriptions of One Hundred Machines, Relating to Arts, Manufactures, and Domestic Life*, Leech and Cheetham, 1822. The first and second editions appeared in the same year.
6. H.W. Dickinson, "James White and His 'New Century of Inventions,'" *Trans. Newcomen Soc.*, vol. 27, no. 1, 1949, pp. 175–179.

**Denis Roegel** is an associate professor of computer science at the University of Lorraine, France, and a member of the LORIA Research Laboratory. He is a coauthor of the *LaTeX Graphics Companion* (Addison-Wesley Professional, 2007) and the creator and maintainer of the LOCOMAT site on mathematical and astronomical tables (<http://locomat.loria.fr>). From 2006 to 2012, he was part of the scientific committee supervising Schwilgué's astronomical clock. Contact him at [roegel@loria.fr](mailto:roegel@loria.fr).

This article originally appeared in *IEEE Annals of the History of Computing*, vol. 38, no. 4, 2016.

myCS

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

Aisling Kelliher  
Virginia Tech

## Technology and the Arts: Educational Encounters of the Third Kind

**M**y introduction to the world of programming as a child was through the line editor on an Atari 800XL, as I copied out Basic commands from computer magazines and waited for them to run... slowly. The fun quota for this not-exactly Minecraft activity was, perhaps unsurprisingly, reached rather quickly, and it took another 15 odd years or so before I attempted to wrestle with bending machines to my will, this time through the object-oriented joy of Java. Over time, I slowly began to enjoy the creative possibilities of communicating with computers at varying levels of programmatic abstraction, and became aware of the tremendous numbers of people doing quite extraordinary, beautiful, and mind-blowing work in concert with programmable machines.

Discovering the technology-arts community and the possibilities of creative coding provided me with a “career re-orienting” encounter for which I am very grateful. Having neither succeeded as a young hobbyist programmer nor pursued a formal computer science education, I am fortunate to have noodled along sufficiently as a design/art/technology hybrid to find a professional and creative outlet and home for my skills and interests.

In looking around today, I can certainly identify less circuitous routes toward achieving an integrated tech-arts education and career path preparation. However, some familiar arguments and scenarios regarding arts and science education continue to appear—albeit in different guises.

### Still with the Two Cultures?

In 1959, when C.P. Snow spoke of the “Two Cultures,” he was referring to the “traditional” arts/humanities and the “forward-thinking” sciences. He described the stark separation, misunderstandings, lost “creative chances,” and

social rigidity delineated between these increasingly separate and distinct forms of human inquiry.<sup>1</sup> Speaking within a Western context, he decried the “fanatical belief in educational specialization” making unbridgeable chasms between the sciences and the nonsciences. Although the notion of the Two Cultures has attracted support, derision, and extension in the last half century (for example, popular or streetwise science has been considered as a “third” culture<sup>2</sup>), the ghosts of Snow’s concerns still linger somewhat in the current and ongoing battle between the science and arts proxies of STEM versus STEAM.

The momentum in favor of STEM (Science, Technology, Engineering, and Math) learning is propelled in part by economic fears about faltering innovation and a reduction in technical workforce skills, particularly in Europe and the US in the face of growing foreign competition. Encouraging and enticing students into focused STEM education and career paths is widely seen as a solid and practical approach to addressing this issue and is favored and supported by many politicians, industry analysts, and funding agencies.

When pressed to include the “A-related” disciplines in this educational model, some proponents insist that the arts are already covered within STEM, while others believe that adding another leg to the chair only serves as a distraction within an arena already struggling with student engagement.<sup>3</sup> However, advocates for a more inclusive form of learning describe adding the “A” as being innovative in directing students through hands-on STEM-related projects that are imbued with opportunities for creative thinking, risk taking, and the cultivation of necessary design skills (see, for example, the STEAM initiative at the Rhode Island School of Design; [www.risd.edu/about/STEM\\_to\\_STEAM](http://www.risd.edu/about/STEM_to_STEAM)).

Although there are arts practitioners who caution about simply throwing the arts into the mix in a subservient or “service-based” role, careful arguments are also made as to how a STEAM formulation can serve as a hook or “on-ramp” for communities of students not typically oriented toward STEM careers or education. Positing a larger umbrella term as a means to increase and sustain diversity is supported in the literature, where art- and design-based technology experiences (from elementary school on up) prove helpful in expanding the pool of participants in technical fields.<sup>4,5</sup>

### **To Code or Not to Code**

A possible bridging argument that has found favor to varying degrees within both STEM and STEAM camps is the claim that everyone should learn how to code. Championed by NBA players (Chris Bosch), former New York city mayors (Michael Bloomberg), rock stars (will.i.am) and giant corporate geeks alike (Bill Gates and Mark Zuckerberg), organizations such as code.org are pushing the idea that a populace that knows how to program will be able to think systematically, break down problems, and become “wizards” of the future with their “superpowers” (see for yourself at [www.youtube.com/watch?v=STRPsW6IY8k](http://www.youtube.com/watch?v=STRPsW6IY8k)).

One could posit that marketing this flavor of computer science as a linear walk from hacking with JavaScript to Internet riches and fame is both self-serving to the affiliated corporate partners and overly narrow in conceptualizing the scope and degree to which programming as a skill and practice could manifest itself in people’s lives. Indeed, perhaps it’s all just a colossal waste of time, because surely the era of intentional programming<sup>6</sup> is just around the corner—or, more bluntly, the imminent time in which “we won’t program computers” but rather will “train them like dogs.”<sup>7</sup>

Speaking pragmatically, I still think we have quite a ways to go before the AI/machine learning/neural net combination shifts engineering to some form of metababysitting. In the meantime, we can continue to tease out the how and why of the variety of ways in which programming, and indeed, technology more broadly, is encountered, learned, and taught within and across the so-called “two cultures.”

### **Encountering Technology-Arts Education**

In the last 20 years, there has been strong growth in the number of integrative tech-arts

certificates, minors, majors, and graduate degree programs offered in universities and colleges across the globe. Foundational initiatives include studio research/atelier labs—such as the MIT Media Lab, New York University’s Interactive Telecommunications Program (NYU’s ITP), and Carnegie Mellon University’s Entertainment Technology Center. Other contemporary exemplars include Interaction Ivrea (now defunct), Umea’s Institute of Design, and the cross-institutional Stanford d.school. During this time, the evolution of new fields of inquiry, such as human-computer interaction, computational storytelling, information visualization, and interaction design, have helped drive the development of the creative technologies industry. Furthermore, the accessibility of cheap and powerful computers and the growth of open source creative tools have introduced new avenues for hybrid tech-arts practitioners.

### **Integrating Tools and Techniques**

Processing, OpenFrameworks, Cinder, and Pure Data are just some of the languages, environments, and libraries taught in innovative and creative tech-arts courses, many of which are also exploring meaningful ways to integrate contemporary algorithmic innovations into expressive and provocative projects. For example, there’s a rich history, tracing back to the early 1970s, of interactive artworks and projects incorporating computer vision techniques for detecting motion, presence, gaze, or facial expressions in participatory installations.<sup>8</sup>

Although computer vision has been a familiar part of tech-arts curricula and syllabi for many years, more recently we see the addition of machine learning as a fundamental component in media arts training. At NYU’s ITP, Heather Dewey-Hagborg, Patrick Hebron, and Gene Kogan have developed and taught a series of courses introducing machine learning to creative practitioners. In a recent article, Kogan outlined some of the challenges (such as the need for big and sometimes hard-to-find datasets and the need for fast machines) and opportunities (such as the wealth of online tutorials and guides and generalizability of machine-learning applications) encountered in teaching and learning machine-learning techniques for artists and designers.<sup>9</sup> Furthermore, Kogan added an as-yet cautionary note with regard to the use of deep-learning libraries in interdisciplinary machine-learning courses, highlighting the lack of “high-level abstraction found in creative



Figure 1. An exercise from the “Copy. Connect. Remap. Repeat” course at the Bergen Academy of Art and Design—remaking a glitched digital image into a quilt by Johanne: (a) the whole quilt and (b) part of the quilt in more detail. (Source: Ben Dalton and Amber Frid-Jimenez, Bergen Academy of Art & Design; used with permission.)

coding libraries,” and the deep and focused expertise required for effective and efficient debugging.

While the next few years will likely yield solutions to this deep dive issue, opportunities also exist to expand the toe-dipping experiences encouraging new types of learners.

### Hooks and On-Ramps

One such example is the “Copy. Connect. Remap. Repeat” course taught back in 2011 at the Bergen Academy of Art and Design by Amber Frid-Jimenez and Ben Dalton (see <http://globe.khib.no/remaprepeat>). Using the course title constructs to introduce students to computational principles, the instructors created a rich variety of novel pedagogical frameworks, including teaching using “glitch” and “reading code as tourists.” Over email, Dalton recently explained the motivation for using glitch as emerging “from the way that a glitch reveals the mechanics of a file format, data stream or process.”

Inspired by the works revealed in Iman Moradi’s book *Glitch: Designing Imperfection* (Mark Batty Publisher, 2009), the instructors created a series of glitching exercises (see Figure 1), enabling students to examine concepts of compression “by breaking various file formats and other people’s code.” Another framing mechanism

used in the course to introduce novices to more complex programs, and indeed, entire software ecosystems (such as GIS mapping software), involved treating programming languages as “if you were a tourist experimenting with a few key phrases, rather than learning a language completely from scratch.” This high-level, almost disposable approach to encountering technical languages provides a compelling hook enabling novices to at least get started using a commonplace metaphor.

### Alternative Venues for Learning Tech-Arts

There are also a variety of alternative online and bricks-and-mortar venues that offer novel, complementary, or continuing education opportunities to develop skills and expertise relevant to the field of technology-arts. As noted earlier, nonprofits such as code.org provide online programs and resources for elementary ages up, aimed at promoting coding as a valuable form of 21st century literacy.

The recently launched Kadenze website ([www.kadenze.com](http://www.kadenze.com)) partners with academic institutions around the world to produce and distribute tech-arts courses from recognized leaders and innovators. With a combination of free and membership-required offerings, Kadenze aims to be, in the emailed words of cofounder Perry Cook, “a truly arts/media friendly learning

management system,” where courses are prepared and taught by “the rock stars who created the language or system or technique... who are regarded as the very best teacher of a given subject, and rock-star artists who use these tools as guest faculty.” This combination of theory, technique, and practice, together with automated grading and feedback tools, aim to shape Kadenze into a lively and engaging environment for tech-arts learning.

The beautifully named School for Poetic Computation in New York provides a hybrid school/residency/research group-based experience (<http://sfpc.io>), where “students and faculty work closely to explore the intersections of code, design, hardware and theory—focusing especially on artistic intervention.” Launched in 2013, the *NY Times* profiled the school by asking readers to “imagine the Robin Williams character from the movie ‘Dead Poets Society’ teaching Objective C instead of ‘O Captain, My Captain.’”<sup>10</sup> The artist-run school offers a 10-week program of integrated tech-art courses three times a year, exploring concepts including physical computing, language design, and building the commons.

### Beyond the STEM/STEAM Menu

The notion of a 21st century liberal arts education without technology should really be as unthinkable as a 20th century version is without the arts. Many of the examples I’ve discussed posit the importance of technical proficiencies embedded within an arts context; however, the reverse also holds equally true. Preparing technologists of all stripes to deal with the complexity of today’s world requires a thorough understanding of the full scope of human experience.

To achieve this, our students need to do more than simply add on a discrete “A” here or an “E” or an “S” over there. Instead, they need to develop broad, encompassing problem-solving abilities through integrative humanistic technology training. This generalist approach provides a strong foundational base that can also directly connect in parallel, or down the line, to fields of specialization and focused expertise. Creating and supporting educational trajectories imbued with authentic encounters with diverse learning cultures can ultimately provide value to individuals, organizations, and institutions invested in cultivating future generations of creators and innovators.

**MM**

## References

1. C.P. Snow, *The Two Cultures*, Cambridge University Press, 1959.
2. J. Brockman, *The Third Culture: Beyond the Scientific Revolution*, Simon & Schuster, 1995.
3. G. May, “STEM not STEAM,” *Inside Higher Education*, 30 Mar. 2016, [www.insidehighered.com/views/2016/03/30/essay-criticizes-idea-adding-arts-push-stem-education](http://www.insidehighered.com/views/2016/03/30/essay-criticizes-idea-adding-arts-push-stem-education).
4. K. DesPortes, M. Spells, and B. DiSalvo, “The MoveLab: Developing Congruence Between Students’ Self-Concepts and Computing,” *Proc. 47th ACM Technical Symp. Computing Science Education (SIGCSE)*, 2016, pp. 267–272.
5. I. Greenberg, D. Kumar, and D. Xu, “Creative Coding and Visual Portfolios for CS1,” *Proc. 43rd ACM Technical Symp. Computer Science Education (SIGCSE)*, 2012, pp. 247–252.
6. C. Simonyi, *The Death of Computer Languages, The Birth of Intentional Programming*, tech. report MSR-TR-95-52, Microsoft, 1995.
7. J. Tanz, “Soon We Won’t Program Computers. We’ll Train them Like Dogs,” *Wired*, 17 May 2016; [www.wired.com/2016/05/the-end-of-code](http://www.wired.com/2016/05/the-end-of-code).
8. G. Levin, “Computer Vision for Artists and Designers: Pedagogic Tools and Techniques for Novice Programmers,” *J. Artificial Intelligence and Soc.*, vol. 20, no. 4, 2006, pp. 462–482.
9. G. Kogan, “Machine Learning for Artists,” *Medium*, 3 Jan. 2016; <https://medium.com/@genekogan/machine-learning-for-artists-e93d20fdb097#rk07a0a0z>.
10. A. O’Leary, “Code to Joy: The School for Poetic Computation Opens,” *The New York Times*, 12 Aug. 2013; <http://bits.blogs.nytimes.com/2013/08/12/code-to-joy-the-school-for-poetic-computation-opens>.

**Aisling Kelliher** is an associate professor in the Department of Computer Science and a Catalyst Fellow at the Institute for Creativity, Arts, and Technology at Virginia Tech. Contact her at [aislingk@vt.edu](mailto:aislingk@vt.edu).

*This article originally appeared in IEEE MultiMedia, vol. 23, no. 3, 2016.*

**myCS**

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

# The State of the Cloud

**THE OBJECTIVE OF THIS LETTER IS TO CONVEY HOW I SEE THE STATE OF CLOUD COMPUTING.**

Before specifically looking at the cloud, it is worth mentioning that digitization is causing huge disruptions and changes in all public/government, education and research and businesses, including enterprises and small-medium businesses. I will lump many topics under digitization such as cloud, big data, Internet of Things (IoT), mobility, social, in-memory computing, artificial intelligence and hyper-connectivity. You can add to (or remove from) the list as you wish. Such digital disruption is forcing the Chief information officer (CIO) to put two hats on. The first requires them to be agile, fast and flexible; and the second requires them to be effective, resourceful and steady. For the former, the CIO needs to enable business growth, drive innovation, adopt smart services and ensure easy access to information. For the latter, the CIO focuses on reducing complexity, freeing up legacy IT budgets, ensuring high-quality operations and safeguarding data privacy & security.

We all know that at the dawn of integrating any new technology trend, there is an overabundance of

challenges, whether technological, cultural, business, organizational or simple acceptance. Technological challenges revolve around how to deal with data - in terms of security, compliance, and privacy as well as from its five "Vs" (volume, velocity, variety, veracity and value) - legacy infrastructure, and heterogeneity of IT deployments, security, and integration. Specific examples include, but are not limited to growing applications landscapes (which come in a wide-variety of versions); data compliance, privacy, export control; exponential growth of unstructured data each year; and the need to support worldwide business presence, which usually happens through expansions, mergers & joint ventures. Organizational and cultural challenges revolve around the accelerated pace of change, skills uplifting and information overload. Business challenges revolve around how to tailor the business and operations models to new realities and more.

What about cloud computing? Cloud computing is becoming the de-facto hosting platform for all types of applications and all social innovations. This is true for industry, government, organizations and society (especially social media). According to a survey published in December 2016 by RightScale ([www.rightscale.com](http://www.rightscale.com)), private clouds are showing stronger growth than public clouds for deployments with over 1000 Virtual Machines (VMs). The survey specifically mentions VMs, but there are usually other types of deployments besides VMs like containers. Readers should keep that in mind. Deployments increased from 22 percent in 2015 to 31 percent in 2016 for private clouds. However, for public clouds the corresponding numbers are 13 percent and 17 percent, respectively. Additional artifacts from the survey are a cloud user leverages six clouds on average; the lack of resources and expertise ranks as the number one challenge (surpassing security, which was demoted to number two), and cost management is becoming a major issue.

The fact that the RightScale survey found that an average cloud consumer uses around six clouds, clearly shows there is abundance of cloud providers in the market. Having many providers is of course good for users, but it can also cause headache and challenges for them. For example, with many cloud providers, users can select best-of-breed cloud service with the best price. We also see there is a race to the bottom when it comes to pricing cloud services, mainly be-

MAZIN YOUSIF

T-Systems International  
[mazin@computer.org](mailto:mazin@computer.org)



*This article originally appeared in  
IEEE Cloud Computing, vol. 4, no. 1, 2017.*

cause providers are jockeying for market share at the expense of more profits. However, interoperability and VM mobility remain open issues. Additionally, customers always fear of lock-in to a single cloud provider. A few other observations of what is happening in the cloud market:

- Cloud providers are forced to have global presence due to data compliance and legal requirements.
- IaaS and SaaS services seem to be dominant and have more traction with users than PaaS.
- Public cloud providers are establishing Private Virtual Cloud (PVC) to further attract enterprises.
- Cloud providers are playing tactical games such as price reduction.
- It is clear that the cloud market is getting bigger than any one single player.

Looking a little further into the challenges for adopting clouds, I would first like to categorize the challenges into three high-level buckets: migration, Total Cost of Ownership (TCO), and Data handling. You can add other categories, but for this article, let me stick to the three I listed. Concerning migration challenges, there is a basic issue that in many instances, an enterprise may not have full knowledge about all of their applications' landscapes. No one wants to admit it, but it is there. This is basic and essential to properly recognize which applications will migrate. In other instances, migration cost may reverse the savings from using the cloud, especially if there is a need for extensive manual handling. A third challenge is that migrated applications may never exploit the benefits of cloud such as scale up/down, because the applications are not cloud-native because they were not written with cloud features in mind. The only way around this is to refactor applications, which again can be costly. Fourthly, some migration approaches such as Lift&Shift seem to be more hype than reality.

Looking at the TCO angle, which is clearly more than unit pricing for a VM or container – the first price consumers research when wanting to use cloud services. A second element affecting TCO is that applications migration is usually more involved and costly than people think. Next, VM sprawl is happening and needs to be managed to control TCO. For the latter, we, of course, can develop policies to turn VMs off, but you can see the point conveyed here.

The third category I listed is handling and dealing with data, which requires more than technology to manage. We know that data is subject to local laws, in terms of privacy, compliance and legal of where data reside. This means that cloud providers need to have world-wide presence to support data requirements in different geographies. There is also the question of dealing with the 5 Vs of data which may well include partitioning and replicating to optimize performance and/or security and privacy.

There is clearly a huge market for public clouds, and a greater need for private clouds. Given the enormously diverse IT requirements for businesses, it is becoming evident that hybrid clouds are going to be more dominant going forward. This is because certain applications landscapes can easily be relayed to a public cloud, while other landscapes, which need to reside in-house for all types of reasons, will remain in private cloud or traditional IT. Yet other landscapes may span both private and public clouds. Ideally, (parts of) applications should migrate automatically based on non-functional requirements across private and public clouds. Also with multi-cloud services, enterprises are able to spread the risk associated with cloud and can take full advantage of financials and other benefits offered by different cloud providers.

**SO WHAT SHOULD A CIO OF AN ENTERPRISE DO?** In order to succeed, many enterprises are adopting a centralized approach to dealing with cloud computing. This approach makes it very easy for anyone in business to purchase cloud computing services, establishing shadow IT operations and at the same time translating into lack of control by the CIO. A centralized approach allows the CIO to broker and govern services and maintain consistent approaches to security, privacy, and costing in line with organizational policies. ●●

---

**MAZIN YOUSIF** is the editor in chief of *IEEE Cloud Computing*. He's the chief technology officer and vice president of architecture for the Royal Dutch Shell Global account at T-Systems International. He has a PhD in computer engineering from Pennsylvania State University. Contact him at [mazin@computer.org](mailto:mazin@computer.org)

# Careers in Software Engineering

**F**or this *ComputingEdge* issue, we asked George Hurlburt, chief scientist at STEM-Corp, a developer of autonomous networking systems; and Jeffrey Voas, a US National Institute of Standards and Technology computer scientist, about career opportunities in software engineering. Hurlburt and Voas co-authored the article “Software Is Driving Software Engineering?” from *IEEE Software*’s January/February 2016 issue.

**ComputingEdge:** What software-engineering careers will grow the most in the next few years?

**Hurlburt:** Networks now permeate the sciences and are used in mainstream areas such as big-data analytics, the Internet of Things (IoT), cyber-physical systems, autonomous systems, some forms of AI, and cybersecurity. There is a need for software engineers who have networking expertise in these areas. That said, there will still be a need for traditional software engineers, programmers, and app developers, who won’t occupy the top tiers of software development but will still fare well.

**Voas:** Careers in three areas—AI, algorithms, and understanding how data and data analytics feed business and economic decisions—will grow. Understanding the three areas together make for a very valuable software engineer for today’s world of IoT, blockchain, mobile apps, big data, and cloud computing.

**ComputingEdge:** What will give college students an advantage over the competition?

**Hurlburt:** You must understand dynamic systems, which are increasingly important, and their underlying mathematics. Additionally, familiarity with agile-development techniques used by teams operating under tight deadlines are still valuable, albeit more for the traditional development market.

**Voas:** Demonstrate in your interview that your good software-engineering background is only one of the valuable elements that you offer an employer. Employers want more than good grades in advanced courses. Tell recruiters about when

you first started writing software, what you wrote, and why. Tell them why you liked it. Let your responses to their questions show that you are a real person, not a robot. Look well-rounded, not geeky. Think about teaching yourself some new material. This shows that you're a serious player. Also, I only hired people who had a passion for solving hard problems and were willing to work until they were solved. I never hired people who asked more questions about the job's benefits than about the types of problems they'd be working on.

**ComputingEdge:** What should applicants keep in mind when applying for software engineering jobs?

**Hurlburt:** The well-rounded individual with demonstrable knowledge and a good work ethic, good social and coping skills, and flexibility will generally do well in a corporate interview. However, many companies hope to attract venture capital by featuring a software engineering superstar team. In these cases, demonstrating raw skill means everything in the interview.

**Voas:** It depends on where you want to work. If you want to work in a place where software is not the core product or service, such as a government agency, you should understand that the organization will probably consider software engineers as just part of the IT infrastructure. You might, therefore, have less job flexibility. If you want to work in a place where software is the core product or service, you should understand ideas such as agile and Scrum, as well as the relationship between testing and coding. With this type of organization, you'll have more responsibility for innovation and ingenuity, and, therefore, more job flexibility.

**ComputingEdge:** How can new hires make the strongest impression in a new position from the beginning?

**Hurlburt:** You should show professional proficiency coupled with a willingness to acquire new knowledge and to learn from mistakes and failures.

**Voas:** Plan to demonstrate how you think, solve problems, and work with teams. Show that you're as comfortable being the follower as being the leader.

**ComputingEdge:** Name one critical mistake for young graduates to avoid when starting their careers?

**Hurlburt:** Avoid assuming that learning is complete after college and not a lifelong commitment. Change continues to accelerate at unprecedented rates, which requires the acquisition of new skills.

**Voas:** Never think that taking more and more courses necessarily improves your chances of getting a job or employers' perception of you. Never think that being book smart is job-ready smart. Your managers may not have your educational credentials, but they have time-on-the-job credentials that you have yet to learn.

**C**omputingEdge's Lori Cameron interviewed Hurlburt and Voas for this article. Contact her at [l.cameron@computer.org](mailto:l.cameron@computer.org) if you would like to contribute to a future *ComputingEdge* article on computing careers. Contact Hurlburt at [gfhurlburt@gmail.com](mailto:gfhurlburt@gmail.com) and Voas at [jeffrey.m.voas@gmail.com](mailto:jeffrey.m.voas@gmail.com). 📧

myCS

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

## CAREER OPPORTUNITIES

**TEAM LEAD**, General Motors, Detroit, MI. Plan, gather reqmts, perform, validate, Team Lead execute & support full lifecycle Vehicle Services Management System (VSMS), Vehicle Health Management (VHM), Over the Air (OTA) reflash applications dvlpmt incldg vehicle modules & feature functionality on GM psgr vehicles, using Java/J2EE, Scala, Simple Build Tool, Play framework, Spring, Oracle 10g/11, Mysql, SQL, Unix, AngularJS, JavaScript, Ajax, & Windows NT. Analyze & improve Over The Air (OTA) re-flashes of psgr vehicle infotainment & telematics electronic control units (ECUs) incldg Central Gateway Module using Open Mobile Alliance Device Management (OMA-DM) protocol during different real-time cases in Vehicle Services Management System (VSMS). Mentor younger dvlpmt teams, & review software being developed to ensure the highest standards in code quality, performance, security, & overall delivery to reqmts. Collaborate with Software Architects to contribute to multilayered software solution architecture & drive design framework alignment. Bachelor, Information Technology, Computer Science, Computer Engrg or related. 5

yrs exp as Programmer Analyst, Technical Lead, Software Engineer, Engineer. Exp must include responsibility for applications & software dvlpmt using J2EE technologies, relational databases such as Oracle, & reqmts gathering, analysis, scheduling, designing, tracking & executing of project. Mail resume to Ref#4187, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI.

**SENIOR QUALITY ASSURANCE ANALYST**, General Motors, Detroit, MI. Test & perform soft updates Over the Air on vehicle internal module such as OnStar core module, & external modules such as center stack module & engine control unit through soft platform. Gather reqmts & define user stories (incldg dvlpmt of advanced Agile reqmts) within vehicle domains. Train Qlty Assurance Anlsts & Jr Anlsts on vehicle domains, & soft updates reqmts. Perform Internal & external reflash of Electronic Control Units & programmable master units. Automate scripts for apps testing using JavaScript (Eclipse) & Quick Test Professional. Perform regression testing & dvlp & enhance automated scripts & reusable library files

on QTPv11 (BPT & data-driven automation framework) using Visual Basic (VB) & Selenium scripts, & UNIX shell scripts. Perform build, sanity, smoke, functional, & regression testing on major, minor & break-fix VSMS releases. Identify test scenarios for automation using Selenium WebDriver. Create & maintain test cases using HP App Lifecycle Mgmt tool. Master, Electrical Engineering, Electronics Engineering, or related. 36 mo exp as Quality Analyst, Soft Team Lead, Systems Engineer, or related, performing regression testing, dvlpg & enhancing automated scripts (test scripts) & reusable library files on QTPv11 (BPT & data-driven automation framework) using VB or Selenium scripts, & UNIX shell scripts. Will accept a bachelor's degree or foreign equivalent degree, in Electrical or Electronics Engrg, or related, followed by at least 5 yrs progressive exp in the specialty, in lieu of the required education & exp. Will also accept any equally suitable combination of education, training, &/or exp which would qualify applicant to perform job offered. Mail resume to Ref#3010, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265.

### TECHNOLOGY

**Oracle America, Inc.**

has openings for

## SYSTEMS ANALYST

positions in **Redwood Shores, CA**

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to [subodh.somani@oracle.com](mailto:subodh.somani@oracle.com), referencing 385.16847.

Oracle supports workforce diversity.

### SOFTWARE

**Oracle America, Inc.**

has openings for

## SOFTWARE DEVELOPERS

positions in **Columbia, Maryland.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to [naga.akella@oracle.com](mailto:naga.akella@oracle.com), referencing 385.18189.

Oracle supports workforce diversity.

### TECHNICAL

**Oracle America, Inc.**

has openings for

## TECHNICAL ANALYST

positions in **Orlando, FL.**

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure. As a member of the IT organization, assist with the design, development, modifications, debugging, and evaluation of programs for use in internal systems within a specific function area.

Apply by e-mailing resume to [josh.geer@oracle.com](mailto:josh.geer@oracle.com), referencing 385.19713.

Oracle supports workforce diversity.

**MPHISIS CORP** has multi openings at various levels for the follow'g positions at its office in NY, NY & unanticipated client sites thr/o the US 1. Info. Sys. Anyst\* - Ana. & provide sys req & spec. 2. SW Dvlper\* - Design, dvlp & modify SW sys. 3. Sys. Architect Dvlper\* - Dvlp IT architecture 4. Graphic UI Desgr\* - Design UI & perform UAT 5. N/W Infra Eng\* - Maintain & TRBL n/w, design, dvlp, install n/w infra appl. 6. Business Operation Anyst\* - Ana bus process thru app of s/w sol. 7. IT Mgr\* - Plan & manage the delivery of IT proj. 8. Enterprise Svc Engagem't Mgr\* - E2E sale of IT svc/prod. 9. Eng Engagem't Mgr\* - Manage & direct business integration of proj activities. 10. Mkt Dvlpt Mgr\* - Promote IT svc/prod. & impl bus plans. Must have a Bachelor/equiv and prior rel. exp, Master/equiv, or Master/equiv and prior rel. exp. Edu/exp req vary depending on position level/type. \*Lead positions in this occupation must have Master/equiv+2yr or Bach/equiv+5yr progressive exp. Travel/relo req. Send resume & applied position to: recruitmentus@mphasis.com or 460 Park Ave. S., Ste# 1101, New York, NY 10016 Attn: Recruit.

**CYBERSECURITY URBAN ACTIVE PROGRAM MANAGER**, Warren, MI, General Motors. Manage, coordinate & track Connected Vehicle Cybersecurity Program among Vehicle Cybersecurity, IT Security, Legal, & Privacy Depts., & hardware/software suppliers, to ensure implementation of U.S./global cybersecurity designs & testing in hardware & embedded software in global vehicles & back office (end-to-end) for Car Sharing & Fleet Management Programs. Coordinate work with Security, IT & Engng leadership & technical staff to identify & plan measures to achieve mature security posture complying with GM Cybersecurity reqmts. Responsible for data protection in transit & at rest through Public Key Infrastructure (PKI), reviewing design network architectures & IT infrastructure to protect sensitive or high risk data of vehicles, companies & persons. Order IT staff to scan code & ensure that they perform due diligence. Responsible for encryption of sensitive data & e-mails, for server security & internet security. Analyze, define & update cybersecurity technical specifications & documentation about IT security technologies, embedded security

& platforms to facilitate communication of cybersecurity reqmts to internal & external customers. Manage safety & security risks of global programs for GM & its customers. Optimize security assessments & the cybersecurity of sourcing processes for vehicle parts. Ph.D. (Doctorate), Information Systems or Electrical Engng, or related. 12 mos exp as IT Project Manager, Cybersecurity Coordinator, or related, being responsible for data protection in transit & at rest through PKI, reviewing design network architectures & IT infrastructure to protect sensitive or high risk data of companies & persons, responsible for encryption of sensitive data & e-mails, for server security & internet security. Mail resume to Ref#2196, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265.

**VIMO** (DBA GetInsured.com) seeks QA Engineer to collab. w/ Prod./Eng. teams to create SW test plans/cases. Run tests. Work w/Auto. team to rev./add test cases. Triage defects. Resumes to Leslie Sullivan, GetInsured.com, 1305 Terra Bella Ave., Mountain View, CA 94043.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**PROGRAMMER  
ANALYST**

positions in **Redwood Shores, CA.**

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to  
sundaresh.shankar@oracle.com,  
referencing 385.19494.

Oracle supports workforce diversity.

**2018-2019  
IEEE-USA Government  
Fellowships**



**Congressional Fellowships**

Seeking U.S. IEEE members interested in spending a year working for a Member of Congress or congressional committee.



**Engineering & Diplomacy Fellowship**

Seeking U.S. IEEE members interested in spending a year serving as a technical adviser at the U.S. State Department.



**USAID Fellowship**

Seeking U.S. IEEE members who are interested in serving as advisors to the U.S. government as a USAID Engineering & International Development Fellow.

**The application deadline for 2018-2019  
Fellowships is 8 December 2017.**

For eligibility requirements and application information, go to  
**www.ieeeusa.org/policy/govfel**  
or contact Erica Wissolik by emailing  
e.wissolik@ieee.org or by calling +1 202 530 8347.

**IEEE★USA** 

## EAN.com LP

has openings for

### REGIONAL HEAD OF PARTNERS: CONNECT

(Job ID#: 728.947)

in **Bellevue, WA:**

Implement process in region and ensure that process has buy in from relevant departments it effects. 20-25% travel to various unanticipated sites throughout the U.S. and internationally required.

To apply, send resume to: EAN.com Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID# .

## SOFTWARE

### Egencia LLC

has openings for

### SOFTWARE ENGINEERS

(Job ID#: 728.1915)

in **Bellevue, WA:**

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: Egencia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID# .

## SOFTWARE

### Egencia LLC

has openings for

### SOFTWARE ENGINEERS

(Job ID#: 728.2457)

in **Chicago, IL:**

Design, implement, and debug software for computers including algorithms and data structures. Travel to various sites in the U.S. and internationally required (less than 50%), and may telecommute up to 20%.

To apply, send resume to: Egencia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

## TECHNOLOGY

### Oracle America, Inc.

has openings for

### SYSTEMS ANALYSTS-SUPPORT

positions in

**Redwood Shores, California.**

Job duties include: Delivery of technical and non-technical services involving Oracle software, hardware, Cloud, and support services to the Oracle customer base while serving as an advocate for customer needs. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to  
Thomas.Biggar@oracle.com,  
referencing 385.15899.  
Oracle supports workforce diversity.

## SOFTWARE

### Oracle America, Inc.

has openings for

### SOFTWARE DEVELOPER

positions in

**Broomfield, CO.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to  
dan.sivilli@oracle.com,  
referencing 385.19398.

Oracle supports workforce diversity.

## SOFTWARE

### Oracle America, Inc.

has openings for

### SOFTWARE DEVELOPER

positions in **Burlington, Massachusetts.**

Job duties include: design, develop, troubleshoot and/or test/QA software; as a member of the software engineering division, apply knowledge of software architecture to perform tasks associated with developing, debugging, or designing software applications or operating systems according to provided design specifications

Apply by e-mailing resume to  
guido.ficco@oracle.com,  
referencing 385.17621.  
Oracle supports workforce diversity.

**SOFTWARE DEVELOPER:** Design, develop, test and implement s/w using knowledge in Java, C++, JNI, Android SDK, NDK, ADB, DDMS, Logcat, Traceview, GIT, JIRA and Agile Development Lifecycle. Must be willing to travel & reloc to unanticipated client locations throughout the US. Reqs MS in comp sci, eng or rel. Mail resumes to Informativ Technologies Inc. 900 Oak Tree Ave, Ste C2, South Plainfield, NJ 07080.

**IT PROFESSIONALS:** Established IT firm has multiple openings for IT Project Managers, Engagement Managers, Technical Project Leads, Technical Team Leads, Account Directors, Business Analysts (Systems), Sr. Consultant/Software Engineers & Software Developers. \*PROJ. MANAGERS require Master's or equiv. in Engg (any), CS, IS or related & 12 mos' relevant indus exp. \*ENGAGEMENT MANAGERS require Master's or equiv. in Business Administration or Mgmt, CS, IT, Engg (any) or related & 12 mos' relevant indus exp. \*PROJ. & TEAM LEADS require Master's or equiv. in

Engg (any), CS, Computer Applications or related & 12 mos' relevant indus exp. For Managers & Leads, we also will accept a Bachelor's or equiv. in the fields stated & 5 yrs' progressively responsible & relevant indus exp. \*ACCOUNT DIRECTORS require a Bachelor's or equiv. in IT, Info. Systems, Mgmt, Finance or related & 5 yrs' relevant indus exp. \*BUSINESS ANALYSTS (SYSTEMS) require a Bachelor's or equiv. in Business Administration or Mgmt, CS, IT, Engg (any) or related & 24 mos' relevant indus exp. \*SR. CONSULTANT/SOFTWARE ENGINEERS & SOFTWARE DEVELOPERS require a Bachelor's or equiv. in Engg (any), CS, IS or related & 24 mos' relevant indus exp. All positions based out of Edison, NJ HQ & subject to relocation to various unanticipated locations throughout the U.S. Qualified applicants mail resumes to: HR Manager, SYSTIME Computer Corporation (dba KPIT), 379 Thornall Street, Edison, NJ 08837.

**VIRTUOSA GLOBAL SOLUTIONS** requires Programmer Analysts to Analyze,

design, develop, test & implement Web based applications and Windows based applications using knowledge of C#.NET, VS 2010, XML, HTML, ASP.NET, VB.NET, WCF, Oracle 11g,SQL Server 2008, Javascript & Windows 98/00/03/NT. Must be willing to travel & reloc to unanticipated client locations throughout the US. Req MS comp sci, eng or rel. Mail resumes to Virtuosa Global Sols LLC 674 US Highway 202/206,Suite 4, Bridgewater, NJ 08807.

**DIALOG SEMICONDUCTOR INC.** Opening for Sr. RF Design Engineer in Chandler, AZ. Design, verify integrated circuits and building blocks to specifications. Exp incl: (RFICs), WLAN or Wi-Fi applications; RF system-in-package, VerilogAMS, Smith Chart, Matlab, and Simulink; Test/validate and characterize LNAs, Pas, and RFIOs, using (SA), (VNA), including CMOS or CML (current-mode-logic), EM tools, and silicon chips. Apply: resumes@diasemi.com, referencing job #10564.63. EOE.

**SOFTWARE**  
**Oracle America, Inc.**  
has openings for  
**SOFTWARE DEVELOPERS**  
positions in **Cambridge, MA.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to john.m.morrison@oracle.com, referencing 385.14549.

Oracle supports workforce diversity.

**Ocean Prediction Postdoctoral Positions**  
**Naval Research Laboratory,**  
**Stennis Space Center, MS**



The Naval Research Laboratory is seeking postdoctoral researchers to push forward the frontiers of ocean forecasting. The work covers a wide scope of physics including surface waves, thermohaline circulation, nearshore circulation, and ocean/atmosphere coupling from global to nearshore scales. This challenging work includes processing and analysis of satellite and in water observations, construction of numerical model systems on high performance computing systems and assimilation for predicting the ocean environment. For a quick overview of some of the research work within the NRL oceanography division at Stennis Space Center, visit the web site:

**<https://www7320.nrlssc.navy.mil/pubs.php>**

Applicants must be a US citizen or permanent resident at time of application. Applications will be accepted until positions are filled. Please e-mail a resume and description of research interests:

**Gregg Jacobs: [gregg.jacobs@nrlssc.navy.mil](mailto:gregg.jacobs@nrlssc.navy.mil)**

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL  
ANALYST-  
SUPPORT**

positions in **Lehi, Utah.**

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs. Offer strategic technical support to assure the highest level of customer satisfaction. Create/utilize automated technology

Apply by e-mailing resume to  
dale.h.morris@oracle.com,  
referencing 385.20178.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE  
DEVELOPER**

positions in **Nashua, NH.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to  
steven.fried@oracle.com,  
referencing 385.19306.

Oracle supports workforce diversity.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**DATABASE  
ADMINISTRATOR**

positions in **Jacksonville, FL.**

Job duties include: Provide enterprise-wide, database administration support for production systems and provide DBA services to application development teams, including database design, database generation, coding, and/or database production support.

Apply by e-mailing resume to  
marc.verreault@oracle.com,  
referencing 385.17731.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL  
ANALYST-  
SUPPORT**

positions in

**Colorado Springs, CO.**

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to  
damon.edmond@oracle.com,  
referencing 385.20626.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE  
DEVELOPER**

positions in **Orlando, FL.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to  
rahul.saraf@oracle.com,  
referencing 385.17082.

Oracle supports workforce diversity.

**INFO TECH**

**Oracle America, Inc.**

has openings for

**IT SENIOR  
MANAGER**

positions in **Redwood Shores, CA**

Job duties include: Manage development, design and optimization of automated testing initiatives and processes.

Apply by e-mailing resume to  
gaurav.narasimhan@oracle.com,  
referencing 385.14346.

Oracle supports workforce diversity.

**DIGITAL PLATFORM ENGINEERS** Multiple positions in Boston, MA. Contribute to architecture, design and development decisions around the creation of comprehensive services and application program interface (API) platform. Create and execute digital information delivery and services platform. Provide comprehensive middleware lifecycle development from design to delivery to scalability. Develop REST/SOAP web services using JAVA and C++ technologies. Build and scale web applications in a production environment and deploy applications into the cloud environment. Apply: C. Andre, MIP F115 Massachusetts Mutual Life Insurance Company, 1295 State Street, Springfield, MA 01111; Please Reference Job ID: 708204000.

**RESEARCHER**, Warren, MI, General Motors. Perform advanced theoretical research on industry standards & benchmark academic investigations, develop & improve next generation passenger vehicle Electrical Control System (ECS) architectures, including prototyping & simulation using C++ & MATLAB. Evaluate & support bench implementation of inter-system communication using Linux, SomelP, & Ethernet technology. Perform

safety verification of software for autonomous passenger vehicles. Evaluate Temporal Logic (timed-automata) based method in C++ for passenger vehicle. Participate in global vehicle manufacturer AUTOSAR, consortium for industrial software architecture standardization. Develop strategy & definition of system level Fault-Tolerant & Fail-Operational ECS architectures involving redundancies & reconfigurations. PhD, Electrical Engrg or Computer Science. 6 mos exp as Graduate Research Assistant or Engineer, performing safety verification of nonlinear dynamic model or of software for autonomous passenger vehicle, & programming or evaluation of Temporal Logic method in C++ for vehicle or robot. Mail resume to Ref#1963, GM Global Mobility, 300 Renaissance Center, MC:482-C32-D44, Detroit, MI 48265.

**MACHINE LEARNING SCIENTIST** - Physiosigns, Inc. (Sunnyvale, CA) Research & explore fundamental issues in computing, dvlp theories models to address issues. Req: bldng prdctve models from elctrcn health rcrds; dvlping & applying orig machine learning algorithms, cntrbtng to pblctns &/or patents in machine learning; indstrl data sci. exp, incl database

querying, data cleanup, & data visualization; prgrmg in Java & Python & database languages, such as SQL & noSQL; dstrbtd processing frameworks (Spark, Hadoop or similar); prfrmng data analysis & visualization in Unix/Linux envrn.; familiarity w/med terms such as ICDs, CPT, & LOINC. Mail resumes: J. Huang, 800 W. California Ave., Ste 200, Sunnyvale, CA 94086, ref job# 116671. EOE.

**DIALOG SEMICONDUCTOR, INC.** Opening for Digital Design Engineer in Santa Clara, CA. Digital Designing, validating/testing; ASIC design; synthesizing RTL code; working with Cadence schematic editor; performing static timing analysis; understanding advanced verification method. Apply: resumes@diasemi.com job# 10564.76. EOE.

**FORWARD NETWORKS, INC.** is recruiting for our Palo Alto, CA office: Member of Technical Staff: Design, develop, troubleshoot and debug software to translate the packet processing behavior of networking devices into our internal software model Mail resume w/ job code #38110 to: Forward Networks, Inc., Attn.: HR, 550 S. California Ave., Ste. 200, Palo Alto, CA 94306.

**TECHNOLOGY**  
**Oracle America, Inc.**  
 has openings for  
**PROGRAMMER ANALYST**  
 positions in **Rocklin, CA.**

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure.

Apply by e-mailing resume to [tom.winston@oracle.com](mailto:tom.winston@oracle.com), referencing 385.19497.

Oracle supports workforce diversity.

**cloudera**  
**CLOUDERA, INC. IS RECRUITING FOR OUR PALO ALTO, CA OFFICE:**

**SOFTWARE ENGINEER #37377:** Design & implement lg distributed systems that scale well—to petabytes of data & 10s of 1000s of nodes.

**SOFTWARE ENGINEER #36761:** Design, implement & test features for distributed system via documentation, code or code review.

**SOFTWARE ENGINEER #37413:** Architect & deploy technical infrastructure to scale ever-growing business needs.

**DATA SCIENTIST #37804:** Present to customers & at conferences on topics related to data science & Cloudera, to support marketing & evangelism of data science topics.

**SOFTWARE ENGINEER-DEVELOPMENT #37559:** Design & implement large distributed systems that scale well—to petabytes of data & 10s of 1000s of nodes.

**SOFTWARE ENGINEER #38061:** Perform backend engineering for cloud infrastructure, inc engineering & modification for build, continuous delivery & deployment systems.

**SOFTWARE ENGINEER-DEVELOPMENT #37891:** Utilize Java & Python to build distributed systems that work from a functionality, performance & scalability standpoint.

**MAIL RESUME WITH JOB CODE # TO CLOUDERA ATTN: HR 1001 PAGE MILL RD., BLDG. 3 PALO ALTO, CA 94304**

**TECHNOLOGY**

**Expedia, Inc.**

has openings for the following positions in **Seattle, WA**:

**Software Engineers (Job ID#: 728.1631):** Design, implement, and debug software for computers including algorithms and data structures.

**Web Designers (Job ID#: 728.1649):** Develop and execute design and layout solutions for online content.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

**SOFTWARE**

**Expedia, Inc.**

has openings for

**SOFTWARE ENGINEERS**

**(Job ID#: 728.2111)**

in **Irving, TX**:

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

**TECHNOLOGY**

**Expedia, Inc.**

has openings for the following positions in **San Francisco, CA**:

**Database Developers (Job ID#: 728.1498):** Coordinate changes to computer databases, test and implement the database applying knowledge of database management systems.

**Oracle Apps Developers (Job ID#: 728.1018):** Responsible for design, development, and offshore vendor management.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL ANALYST**

positions in **Lehi, Utah**.

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs. Offer strategic technical support to assure the highest level of customer satisfaction.

Apply by e-mailing resume to [Khaled.itani@oracle.com](mailto:Khaled.itani@oracle.com), referencing 385.19952.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE DEVELOPER**

positions in **Seattle, WA**.

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to [zk.kong@oracle.com](mailto:zk.kong@oracle.com), referencing 385.20874.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL ANALYSTS**

positions in **Burlington, MA and Marlboro, MA**.

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure. As a member of the IT organization, assist with the design, development, modifications, debugging, and evaluation of programs for use in internal systems within a specific function area. Travel to various unanticipated sites throughout the United States required.

Apply by e-mailing resume to [larry.campbell@oracle.com](mailto:larry.campbell@oracle.com), referencing 385.19482.

Oracle supports workforce diversity.

**JAVA DEVELOPER.** Central NJ IT Consulting Company requires candidates for following position at their primary Princeton, NJ location, Java Developer: to Design, develop & implement business software apps using Java & J2EE, MQ, struts, Jira and CVS,GIT. Candidates must have 1-2 yrs of mandatory exp in related -field. All positions require: MS in CS/Engineering/Comp Applications/Business or related. BS degree + 5yrs exp can be substituted for the MS requirement any combination of foreign edu + related exp equivalent to a US Masters, or any combination of foreign edu +related exp equivalent to a BS Degree will be accepted. Travel to several unanticipated locations all over US & might involve relocation consistent w/client needs & State & Local needs. Mail your resume to: Kellton Tech Inc., Attn: HR, 3 Independence Way, STE 209, Princeton NJ 08540.

**SAP PROJECT MANAGER.** Princeton, NJ based IT Consulting company requires SAP Project Manager: Oversee the successful planning and implementation of multiple projects/releases of

SAP, using standard project management tools and procedures. Will be responsible to initiate, plan, execute, monitor and control and close, and the supporting processes. Position requires at least 12 months of applying Accounting principles to SAP Enterprise Resource Planning (ERP)+MS in Computer Science/Accounting Any combination of foreign edu equivalent to a US Masters will be accepted. Position involves travel to several unanticipated locations all over US & might involve relocation. Mail your resume to: Global HR, Kellton Tech Inc., 3 Independence way, STE 209, Princeton NJ 08540.

**LEAD SYSTEMS ANALYSTS.** Princeton, NJ based company requires, Lead Systems Analysts: To analyze and implement business critical ERP Systems, maintain IT Infrastructure, Security, Data Governance and ensure the availability of complete DC (Data Centre) Systems - Servers, DB's (Databases), Storage, Applications, Network& Security components. Candidates should have one year mandatory exp in Platespin Data Center Migration, Linode Stack, Micro

Focus, Matlab, Teradata, Cisco UCS, Golden Gate, Qlikview/QlikSense All positions require MS Comp Sc/Engrg/Info sys/Science or any related field. BS degree + 5yrs exp can be substituted for the MS requirement and any combination of foreign education equivalent to a US Masters or Bachelors degree will be accepted. Position require travel to client sites all over the US. Mail resume to: Kellton Tech Inc., Attn: HR, 3, Independence way, Suite 209 Princeton, NJ 08540.

**FILEMAKER, INC.** has openings for the following positions in Santa Clara, CA: **Software Engineer Applications Manager (Req# A3L2PW)** Oversee the team working on server platform to support both on-premise and cloud. **Software Engineer Applications (Req# 9GMUAY)** Des & dev SW for desktop apps & databases. Refer to Job# & mail resume & transcript: FileMaker, Inc, Attn: CP, 5201 Patrick Henry Drive, Santa Clara, CA 95054. FileMaker, Inc. is an EOE/AA m/f/disability/vets.

**TECHNOLOGY**  
**Oracle America, Inc.**  
 has openings for  
**APPLICATIONS DEVELOPERS**  
 positions in  
**Burlington, Massachusetts.**

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications.

Apply by e-mailing resume to  
 rakeshk.arora@oracle.com,  
 referencing 385.19522.

Oracle supports workforce diversity.

**SOFTWARE**  
**Oracle America, Inc.**  
 has openings for  
**SOFTWARE DEVELOPER**  
 positions in  
**Cambridge, Massachusetts.**

Job duties include: design, develop, troubleshoot and/or test/QA software; as a member of the software engineering division, apply knowledge of software architecture to perform tasks associated with developing, debugging, or designing software applications or operating systems according to provided design specifications. May telecommute from home.

Apply by e-mailing resume to  
 ashish.surana@oracle.com,  
 referencing 385.20836.

Oracle supports workforce diversity.

**TECHNOLOGY**  
**Oracle America, Inc.**  
 has openings for  
**CONSULTING PROJECT PRINCIPAL CONSULTANT**  
 positions in **Redwood Shores, CA.**

Job duties include: Responsible for ensuring that a quality, integrated software solution is delivered in a timely manner, at budget, and to client satisfaction. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to  
 vernal.hamilton@oracle.com,  
 referencing 385.17094.

Oracle supports workforce diversity.

SOFTWARE

## Expedia, Inc.

has openings for

# SOFTWARE ENGINEERS

(Job ID#: 728.1760)

in **Ann Arbor, MI:**

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

SOFTWARE

## HomeAway.com, Inc.

has openings for

# SOFTWARE ENGINEERS

(Job ID#: 728.4169)

in **Austin, TX:**

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: HomeAway Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

SOFTWARE

## HomeAway.com, Inc.

has openings for

# SOFTWARE ENGINEERS

(Job ID#: 728.1992)

in **Bellevue, WA:**

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: HomeAway Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

TECHNOLOGY

## Oracle America, Inc.

has openings for

# CLIENT SUCCESS SPECIALIST

positions in **Austin, Texas.**

Job duties include: Drive maximum adoption of Oracle solution and identify/drive product expansion opportunities via high value relationship with the client. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to [craig.lockwood@oracle.com](mailto:craig.lockwood@oracle.com), referencing 385.10750.

Oracle supports workforce diversity.

TECHNICAL

## Oracle America, Inc.

has openings for

# TECHNICAL/ PROGRAMMER ANALYST

positions in **Rocklin, CA.**

Job duties include: Analyze user requirements to develop, implement, and/or support Oracle's global infrastructure.

Apply by e-mailing resume to [tom.winston@oracle.com](mailto:tom.winston@oracle.com), referencing 385.17158.

Oracle supports workforce diversity.

TECHNOLOGY

## Oracle America, Inc.

has openings for

# APPLICATION DEVELOPER

positions in **Burlington, MA.**

Job duties include: Analyze, design, develop, troubleshoot and debug software programs for commercial or end-user applications.

Apply by e-mailing resume to [parveen.minocha@oracle.com](mailto:parveen.minocha@oracle.com), referencing 385.18354.

Oracle supports workforce diversity.

**PROGRAMMER ANALYSTS.** Central NJ IT Consulting Company requires Programmer Analysts at their primary Princeton, NJ location, to Design, develop & implement business software apps using SharePoint 2010, .NET, AngularJS, Mongo DB, NHibernate, Node js, Rest Micro Service, Grunt,Karma,Protractor, Log4Net. Candidates must have 1 year of mandatory exp in related -field. All positions require: MS in CS/Engineering/Comp Applications/Business or related. BS degree + 5yrs exp can be substituted for the MS requirement any combination of foreign edu + related exp equivalent to a US Masters, or any combination of foreign edu +related exp equivalent to a BS Degree will be accepted. Travel to several unanticipated locations all over US & might involve relocation consistent w/client needs & State & Local needs. Mail your resume to: Kellton Tech Inc., Attn: HR, 3 Independence Way, STE 209, Princeton NJ 08540.

**SR. MANAGER.** Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Resp. for end-to-end develop. & maint. of Hyperion

Financial Mgmt. (HFM) appls. Develop & manage Hyperion solutions. Provide critical appl. design recommends. that align with long & short term IT & business strategies. Analyze requests for new functionality & enhancements. Required to build solution as per external SEC reporting & mgmt.. reporting using Hyperion tools. Requires: M.S. degree in Comp. Sci, Finance, Eng. or related field & 3 yrs. exp. in job offered or 3 yrs. exp. as a Consultant or Analyst. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in the comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. using Hyperion & 1 yr. exp. with SEC reporting. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

**CONSULTING MANAGER.** Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Prepare, facilitate & support appl. design sessions for data integration & data qual. using Oracle Hyperion Financial Data Quality Mgmt. (FDM/FDQM), Oracle Hyperion Financial Data Quality Mgmt. Enterprise Mgmt. (FMEE) & Oracle Data

Integrator (ODI). Define project scope & day-to-day execution of client engagements for Finance/Enterprise Performance Mgmt. deliverables incl. strategy, blueprints, design, implement. & testing. Requires: M.S. degree in Comp. Sci., Mgmt, Eng. or related field & 3 yrs. exp. in job offered or 3 yrs. exp. as a Consultant or Mgr. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in the comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 3 yrs. exp. preparing, facilitating & supporting appl. design sessions for data integration & data qual. using Oracle Hyperion FDM/FDQM. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

**APPLICATIONS ENGINEER.** Valassis Communications, Inc. has an opening for the position of Applications Engineer in Livonia, MI to analyze, model/design, create, modify, & support complex systems, processes or operations. To apply mail resume to Valassis, Attn: Patty [MI-11307.25], 19975 Victor Parkway, Livonia, MI 48152.

TECHNOLOGY

Expedia, Inc.

has openings for the following positions in **Bellevue, Washington** (various/levels/types):

**Software Engineers (Job ID#: SW417):** Design, implement, and debug software for computers including algorithms and data structures. **Network Engineers (Job ID#: 728.1873):** Build, configure and troubleshoot voice infrastructure hardware. Travel to various unanticipated sites throughout the U.S. required. **Program Managers (Job ID#: 728.2410):** Responsible for microcomputer software product design features and coordinating development of software. **Network Engineers (Job ID#: 728.2463):** Provide 2nd/ 3rd line technical data infrastructure support. Travel to various unanticipated sites throughout the U.S. required (less than 50%). **Managers, Software Development (Job ID#: 728.2353):** Responsible for building world class software to help detect and prevent fraud. **Release Engineers (Job ID#: 728.1641):** Build and perform deployment system defect tracking, identify all of the sources, tools, environments, and other components. 20-25% travel required within the U.S. May travel internationally. May telecommute from home. **User Experience Designers (Job ID#: 728.1648):** Responsible for the delivery, quality, and success of user experience for a product or business throughout the entire lifecycle. **Analytics Managers (Job ID#: 728.1892):** Team up with Landing Pages product team/other SLP associated teams to figure out how to enhance page traffic, conversion/profit, and improve user experience through actionable analytics. **Analytics Managers (Job ID#: 728.1661):** Convert raw data into actionable business insights. **Oracle Functional Analysts (Job ID#: 728.2215):** Play role in supporting and ensuring smooth operations of the Oracle E-Business Suite environment. **Oracle Functional Analysts (Job ID#: 728.1687):** Act as a group wide and customer advocate for quality architecture design, coding and best practices. **Managers, Software Development (Job ID#: 728.2500):** Manage team, providing software solutions for state of the art e-commerce processing systems. **Data Scientists (Job ID#: 728.1897):** Design and develop automated framework to measure and monitor quality metrics of critical attributes.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

## CAREER OPPORTUNITIES

**IT SOFTWARE DEVELOPER**, Roswell, GA, General Motors. Design, dvlp & deliver software apps used by users in Global Purchasing & Supply Chain (GPSC) org in Korea. Establish roadmaps for software apps to support delivery of new software & web apps & provide operational excellence in GPSC. Assure standard infrastructure architecture & validate the environment is set up properly incldg Storage Area Network (SAN)/Network Attached Storage (NAS) storage mount, & load balancer. Dvlp J2EE apps with JSP, Servlets, Struts, Spring & MVC framework. Deploy source code on servers such as Apache Tomcat & Oracle WebLogic using Jenkins & SVN or WAR file. Analyze source code through HP Fortify in ANT build environment. Design database & create SQL queries & stored procedures using Oracle PL/SQL language & other DB objects in Oracle RDBMS. Write Linux shell scripts that gives app ability to be called by scheduler system such as AutoSys or Crontab. Bachelor, Computer Science, Information Systems Engrg, Computer Engrg, or related. 6 mos exp

as IT Software Developer, Engineer or related, assuring standard infrastructure architecture, validating environment set up properly incldg SAN/NAS storage mount, & load balancer, dvlpg J2EE applications with JSP/Servlets, or related. Mail resume to Ref#1125, GM Global Mobility, 300 Renaissance Center, Mail Code 482-C32-D44, Detroit, MI 48265.

**SOFTWARE ARCHITECT-ATG**. **Harland Clarke Corp** has an open position of Software Architect-ATG in San Antonio, TX to partner with the Enterprise Architecture team & business units to develop business technology strategy & drive the technical design & implementations. To apply mail resume to Harland Clarke Corp, Attn: Monica [17-TX-6174.86], 15955 La Cantera Pkwy, San Antonio, TX 78256.

**SYSTEMS ANALYST III-EPIC SAN ADMINISTRATOR** Provide advanced level support for software upgrades, implementations and maintenance as

needed. Responsibilities will include: UNIX system administration, HPUX, RedHat, Other Enterprise Linux Distribution; administration of Disaster Recovery (DR) environment for UNIX servers; storage administration; management of AHS backup environments (Avamar); and hardware implementation for server and storage platforms including standalone servers, blade chassis, and various storage arrays. Willingness to work nights as needed, required. Apply to: Gerald O'Mara, #82115, AHS Hospital Corp, 100 Madison Avenue, Morristown, NJ 07960.

**LIGHTBEND** seeks: Software Consultant - design & develop software using modern technologies like Java, Akka, and Play Framework; Develop Company's Reactive Platform using functional programming concepts. Up to 40% Domestic Travel. Worksite: May telecommute from any US location or headquarters (San Francisco, CA 94105). To apply send resume/cover(20170405YG) to Lightbend, 625 Market St., Ste. 1000, San Francisco, CA, 94105.

### TECHNOLOGY

## Expedia, Inc.

has openings for the following positions in **Chicago, IL** (various/levels/types):

**Software Engineers (Job ID#: SW417-CH)**: Design, implement, and debug software for computers including algorithms and data structures. **IT Service Management Leads (Job ID#: 728.2125)**: Create, develop and help maintain Change Management processes. Analyze change records to determine the priority, urgency, impact, and risk. **Technology Leads (Job ID#: 728.2221)**: Develop powerful, scalable, and reliable data products using SQL and non-SQL technologies. **Test Design Engineers (Sr. QA Engineers) (Job ID#: 728.2141)**: Participate in daily standups sprint planning, and retrospective sessions. **Web Designers (Job ID#: 728.2188)**: Define visual hierarchy and treatment for range of optimizations and new features of products throughout entire lifecycle. **Managers, Engineering (Job ID#: 728.2178)**: Manage a team of developers who collaborate with stakeholders, program and product management, and architects to ensure that the solution meet requirements. **Managers, Engineering (Job ID#: 728.2183)**: Responsible for architecture, design, construction, testing, and implementation of software. **Sr. Database Engineers (Technical Lead Database Administrators) (Job ID#: 728.2126)**: Technical Lead DBA for Oracle Exadata Database Machine. **Engineering Managers (Job ID#: 728.2165)**: Design, implement, and debug software for computers including algorithms and data structures. **Technical Product Managers (Job ID#: 728.2162)**: Gather detailed business requirements from stakeholders and work closely with technology staff to translate requirements into functional designs and specifications. **Directors of Technology (Job ID#: 728.2083)**: Manage multiple concurrent, complex, and unique projects and initiatives core for internet travel service company. 15% travel required to various unanticipated sites throughout the US and the world. **Application Engineers (Job ID#: 728.2562)**: Design and implement software to improve availability, scalability, latency, and efficiency.

To apply, send resume to: Expedia Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

**REAL INTENT, INC.** Opening for Sr. Technical Marketing Mgr (Sales Eng) in Sunnyvale, CA. Collaborate w/R&D, Field Apps, & Sales to understand cust. reqmnt. Exp incl: Verilog &/or VHDL hw lang & SystemVerilog; tech specs of EDA software; SoC design; electronic design automation; SDC constraints & CDC syst reqmnts; tech presentations; & dev sales plans. Apply: [cheryl@realintent.com](mailto:cheryl@realintent.com), job# 664.24. EOE.

**CH2M HILL, INC.** in Kansas City, MO seeks a Project Manager 1 to lead and manage projects/programs that are aligned to the water business group markets and underlying business objectives. To apply, mail resume to: Shelly Saitta, CH2M HILL, 9191 S. Jamaica St., Englewood, CO 80112. Must reference job code: 13176.

**ERICSSON INC.** has openings for the following positions: **ENGINEER- SOFTWARE** \_ Ericsson Inc. Engineer-Software. **SANTA CLARA, CA.** Research, design, develop & test os-level sw for comm. comp. apps. Up to 15% domestic/international travel. Mail resume to Ericsson Inc. 6300 Legacy Dr, R1-C12 Plano, TX 75024 Job ID# 16-CA-3605. **ENGINEER QUALITY-ASSURANCE** \_ Ericsson Inc. Engineer Quality-Assurance, **NEWTON, MA.** Develop & execute software test plans to identify software problems & causes. Up to 10% international travel. Mail resume to Ericsson Inc. 6300 Legacy Dr, R1-C12 Plano, TX 75024 Job ID# 16-MA-3648. **SOLUTIONS ARCHITECT** \_ Ericsson Inc., Solutions Architect, **OVERLAND PARK, KS** provide guidance in cross-team Agile delivery mgmt. Travel required. Mail resume to Ericsson Inc. 6300 Legacy Dr, R1-C12 Plano, TX 75024 Job ID# 16-KS-3793.

**SPLUNK INC.** has the following job opportunities in **San Francisco, CA:** **Senior Software Engineer in Test** (Req#**ACGV86**) Design & dev internal sw tools & infrastructure. Work on perf improvements & backend algorithms. **Senior Software Engineer** (Req#**9N-4RXK**) Build front-end & back-end apps on top of co. platform. Create, expand & maintain scalable & fault tolerant apps. **Pricing Strategist** (Req#**9YDUL8**) Drive prod pricing strategy & define go-to-market plans. Define pricing & licensing models for new prod & services. **Solutions Engineer** (Req#**9ZWSAA**) Drive and manage tech eval & prototyping of cloud data interface & collection add-ons. **Splunk Inc.** has the following job opportunity in **San Jose, CA:** **Software Engineer** (Req#**A6YVR4**) Create

fault-tolerant distributed sys in multi-threaded, multi-process environment for co. platform. Refer to Req# & mail resume to Splunk Inc., ATTN: J. Aldax, 270 Brannan Street, San Francisco CA 94107. Individuals seeking employment at Splunk are considered without regards to race, religion, color, national origin, ancestry, sex, gender, gender identity, gender expression, sexual orientation, marital status, age, physical or mental disability or medical condition (except where physical fitness is a valid occupational qualification), genetic information, veteran status, or any other consideration made unlawful by federal, state or local laws. To review US DOL's EEO is The Law notice please visit: [https://careers.jobvite.com/Splunk/EEO\\_poster.pdf](https://careers.jobvite.com/Splunk/EEO_poster.pdf). To review Splunk's EEO Policy Statement please visit: <http://careers.jobvite.com/Careers/Splunk/EEO-Policy-Statement.pdf>. Pursuant to the San Francisco Fair Chance Ordinance, we will consider for employment qualified applicants with arrest and conviction records.

**COMPUTER SYSTEMS ANALYST.** Position available in Frederick, MD. Assess the usefulness of pre-developed application packages and adapt them to client user environments. Analyze information processing or computer needs and plan and design computer systems. Define the goals of the client system and describe logical operational steps of programs to reach goals. Train client users to work with implemented computer systems and programs. Prepare cost benefit and return on investment analyses to aid in decisions on system implementation. Use computer in analysis and solution of business problems. Evaluate software and hardware to determine business compatibility. Provide technical assistance as needed for hardware, software, and services including VMWare, SonicWALL, Dell servers, and Microsoft Exchange. Suggest systems solutions. Apply to Systemadix Technologies LLC: [jobs@systemadix.com](mailto:jobs@systemadix.com), reference job ID CSA17.

**TECHNOLOGY**

**WhatsApp, Inc.**

currently has the following openings in **Menlo Park, CA** (multiple openings/various levels):

**Software Engineer (3236J)** Help build the next generation of systems behind WhatsApp's products, create web and/or mobile applications that reach over one billion people, and build high-volume servers to support our content.

**Product Designer (4213J)** Design, prototype, and build new features for WhatsApp's website or mobile applications.

Mail resume to: WhatsApp, Inc. Attn: SB-GIM, 1 Hacker Way, Menlo Park, CA 94025. Must reference job title & job# shown above, when applying.

**SOFTWARE**

**Hotwire, Inc.**

has openings for

**SOFTWARE ENGINEERS**

(Job ID#: **SW417-HSF**) in **San Francisco, CA** (various/levels/types):

Design, implement, and debug software for computers including algorithms and data structures.

To apply, send resume to: Hotwire Recruiting, 333 108th Avenue NE, Bellevue, WA 98004. Must reference Job ID#.

## TECHNOLOGY

# Intuit Inc.

has openings for the following positions in **Mountain View, California**:

**Software Engineers (Job code: SW417)**: Apply software development practices to design, implement, and support individual software projects. **Senior Software Engineers (Job code: SSW417)**: Exercise senior level knowledge in selecting methods and techniques to design, implement, modify and support a variety of software products and services to meet user or system specifications. **Staff Software Engineers (Job code: STSW417)**: Apply master level software engineering and industry best practices to design, implement, and support software products and services. **Software Engineers in Quality (Job code: SWQ417)**: Apply best software engineering practices to ensure quality of products and services by designing and implementing test strategies, test automation, and quality tools and processes. **Senior Software Engineers in Quality (Job code: SSWQ417)**: Apply senior level software engineering practices and procedures to design, influence, and drive quality and testability of products and services. **Product Managers (Job code: I-2653)**: Design and bring to market revenue generating, customer-driven software products and services that deliver great customer experiences.

Positions in **San Diego, California**:

**Software Engineers (Job code: SW417-SD)**: Apply software development practices to design, implement, and support individual software projects. **Senior Software Engineers (Job code: SSW417-SD)**: Exercise senior level knowledge in selecting methods and techniques to design, implement, modify and support a variety of software products and services to meet user or system specifications. **Senior Application Operations Engineers (Job code: I-482)**: Exercise judgment in selecting methods and techniques to deliver operational solutions. Engages the broader team to understand and apply best practices in the selection process.

Positions in **Woodland Hills, California**:

**Staff Business Data Analysts (Job code: I- 2767)**: Write SQL queries to pull sales data using Site Catalyst.

To apply, submit resume to

Intuit Inc., Attn: Olivia Sawyer, J203-6, 2800 E. Commerce Center Place, Tucson, AZ 85706.

You must include the job code on your resume/cover letter. Intuit supports workforce diversity.

**COMPUTER PROGRAMMER:** Write, update, & maintain computer programs, review programs. Utilize Data Mapping, Data Conversion, Report Generation, Version Control Tools, XML, Unix, SQL Server, Oracle, Selenium, UFT, Load Runner, Python. Will work in unanticipated locations. Req 2 years exp. Send resume to NobleSoft Solutions, Inc. 405 Executive Dr. Langhorne PA 19047.

**RALLY HEALTH, INC.** seeks Principal Software Engineer in San Francisco, CA: Analyze reqs, architect, design, dev, integrate & maintain software systems. Tune sw performance to scale the sys. Refer to Req# ADXNWF & mail resume to Rally Health Inc, Attn: D. Griffin, 665 3rd St, Ste 320, San Francisco, CA 94107.

**MIST SYSTEMS, INC.** is recruiting for our Cupertino, CA office: Member of Technical Staff: Design, write code, and test software for running in the network device (access point). Develop in and for Linux operating system. Mail resume w/ job code #38250 to: Mist Systems, Attn.: HR, 1601 S. De Anza Blvd., Ste. 248, Cupertino, CA 95014.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**USER  
EXPERIENCE  
DEVELOPER**

positions in  
**Cambridge, Massachusetts.**

Job duties include: develop and apply software design/usability processes in the investigation of technical problems; may develop user profiles, with emphasis on human error control, display issues, visual interaction, and task and objective analyses.

Apply by e-mailing resume to [matthew.lennon@oracle.com](mailto:matthew.lennon@oracle.com), referencing 385.19226. Oracle supports workforce diversity.

**Cisco Systems, Inc. is accepting resumes for the following positions:**

**ATLANTA, GA: Solutions Integration Architect (Ref.# ATL6):** Responsible for IT advisory and technical consulting services development and delivery. Telecommuting Permitted.

**BOXBOROUGH, MA: Network Consulting Engineer (Ref.# BOX20):** Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted.

**CENTENNIAL, CO: Systems Engineer (Ref.# CENT51):** Provide business-level guidance to the account team or operation on technology trends and competitive threats, both at a technical and business level.

**ISELIN/EDISON, NJ: Network Consulting Engineer (Ref.# ED2):** Responsible for the support and delivery of Advanced Services to company's major accounts.

**LAWRENCEVILLE, GA: Software Engineer (Ref.# LV5):** Responsible for the definition, design, development, test, debugging, release, enhancement or maintenance of networking software. **Customer Support Engineer (Ref.# LV6):** Responsible for providing technical support regarding the company's proprietary systems and software.

**NEW YORK, NY: Systems Engineer (Ref.# NY13):** Provide business-level guidance to the account team or operation on technology trends and competitive threats, both at a technical and business level. Telecommuting permitted and travel may be required to various unanticipated locations throughout the United States. **Systems Engineer (Ref.# NY5):** Provide business-level guidance to the account team or operation on technology trends and competitive threats, both at a technical and business level.

**RESEARCH TRIANGLE PARK, NC: Network Consulting Engineer (Ref.# RTP245):** Responsible for the support and delivery of Advanced Services to company's major accounts. Travel may be required to various unanticipated locations throughout the United States. **IT Architect (Ref.# RTP125):** Design, develop, and implement data models for enterprise-level applications and systems. **Network Consulting Engineer (Ref.# RTP145):** Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted and travel may be required to various unanticipated locations throughout the United States. **IT Project Manager (Ref.# RTP105):** Responsible for defining IT project requirements, plans, schedules, and resources for small to large/complex projects. **Network Consulting Engineer (Ref.# RTP954):** Responsible for the support and delivery of Advanced Services to company's major accounts. Telecommuting permitted. **Network Consulting Engineer (Ref.# RTP2):** Responsible for the support and delivery of Advanced Services to company's major accounts. **Custom Application Engineer (Ref.# RTP136):** Responsible for troubleshooting, resolving technical issues and implementing SAP HANA/Basis, Hadoop Admin, Unix and Cloud Solution. **Project Manager (Ref.# RTP15):** Responsible for the day to day execution, support, developing continual improvement opportunities and analysis of an IT Service.

**SAN JOSE/MILPITAS/SANTA CLARA, CA: Hardware Engineer (Ref.# SJ5):** Responsible for the specification, design, development, test, enhancement, and sustaining of networking hardware. **Database Administrator (Ref.# SJ43):** Provide database design and management function for business and/or engineering computer databases. **Systems Administrator (Ref.# SJ12):** Provide systems design and management function for business and/or engineering computer systems.

**PLEASE MAIL RESUMES WITH REFERENCE NUMBER TO CISCO SYSTEMS, INC., ATTN: G51G, 170 W. Tasman Drive, Mail Stop: SJC 5/1/4, San Jose, CA 95134.** No phone calls please. Must be legally authorized to work in the U.S. without sponsorship. EOE.

[www.cisco.com](http://www.cisco.com)

## Apple Inc. has the following job opportunities in Cupertino, CA:

**Hardware Development Engineer (Req# A2WVL7)** Dvlp & excute image qulty anlysis tstng fr iPhone, iPad, & Mac cmra sys.

**PC Board Designer (Req# A9FVBB)** Stdy, evlute, & implmnt PCB dsgn trnds & pcceses to Apple's core tech.

**Software Engineer Applications (Req# ABETZC)** Des and dev sw for network security.

**Software Development Engineer (Req# ACJTUC)** Strat, architect, and build infras, utilizing state-of-the-art tech and cutting-edge sys auto to help Apple's sys perform and scale.

**Web Developer (Req# 9XESVJ)** Des & dev SW & srvc for Intranet web portals.

**Hardware Development Engineer (Req# 9GQ4MC)** Research, des, dev & launch next-gen Sensing Tech. Travel req 20%.

**Software Development Engineer (Multiple Positions) (Req# 9QLUV5)** Resp for plan & execution of cert test on iOS cell products, iPhone & iPad.

**Machine Learning Engineer (Req# 9V7VV9)** Dev innovative mach lrn'g & data mining models & algrthms to solve probs in business apps.

**Hardware Development Engineer (Req# 9H5W57)** Des, dev, & intrgrte haptic actuators, sys, & tech's for Apple prods. Travel req 20%.

**Software Engineer Applications (Req# ADBSV6)** Provide DB archtctr & dsgn solutions for admnstratng lrg DB infrstrctr.

**Software Engineer Applications (Req# 9XT2VC)** Des & dev SW for high-perform index & search sys.

**Software Development Engineer**

**(Req# 9DP2YG)** Cmbne deep analytics & user study to discover new opportunities for Siri worldwide.

**Software Development Engineer (Req# 9UG3N5)** Dfne a framewrk for imge pccssng with in-complt info derivd from mltple sourcs such as still, video & near-infrared.

**Software Engineer Applications (Req# 9UHQXY)** Dsgn & dvlp wb-bsd, high-prfrmnce apps fr Apple's (WWDR) team usng Java & J2EE rlld techs.

**Network Engineer (Req# ACQUP9)** Excte on ntwrk engineering prjcts & collabrtre w other enginrng & bsns teams to delvr & maintain netwrkng infrstrctre.

**Software Engineer Applications (Req# AAR2VR)** Dsgn high-scale bcknd web srvc for iCloud & rlld apps within Apple ecosystem.

**Software Engineer Applications (Req# ADLNLW)** Dsgn & dvlp SW for Apple servcs.

**Software Development Engineer (Req# A4234P)** Build connctvty & data commncions srvc used on all of Apple's iOS & OS X dvics.

**ASIC Design Engineer (Req# 9Z53LX)** Create SW to verify architecture & functionality of pre-silicon HW designs.

**Systems Design Engineer (Req# 9F7SKZ)** Design & develop hw & sw for camera test in iPhone/iPad/Mac products. Travel req'd 20%.

**Systems Design Engineer (Req# AA2VV8)** Dsgn & dvlp HW for consumer electronics systems.

**System Design Engineer (Req# 9E5PK2)** Des & eval radio prfmnce for phones & tablets.

**Software Engineer Systems (Req# 9SS4B6)** Des & dev SW used by

3rd party dvlprs to validate & inspect SW implmnt'n.

**Software Development Engineer (Req# 9A7UFD)**: Dsgn & dev SW for millions of watch users.

**Hardware Development Engineer (Req# 9CVQXS)** Dsgn & vlde optcal snsr systems for cnsmr elctrncs prdcts. Travel req'd 15%.

**Document Control Specialist (Req# A2E5C3)** Anlyz, proc & track Eng Change Req, Eng Change Order, Manufac. Change Order submissions to min proc time btwn submssn & impl.

**Hardware Development Engineer (Req# 9H8Q2K)** Develop audio tech product features.

**Software Engineer Applications (Req# A5N4QX)** Dsgn, dev & execute load & prfrmnce tests for various iCloud srvc.

**Software Engineer Applications (Req# 9X5SPZ)** Archtct, author & deliver SW to improve availability, scalability & security of Apl's intrnt srvc.

**Software Development Engineer (Req# 9Y2Q4G)** Dsgn, dvlp & mntn frmwrk for tstng Apple's HomeKit Automation sltn.

**Software Engineer Applications (Req# 9ZDVX9)** Wrk with cross functnl teams to gather business reqs & deliver iOS & MacOS sltns.

**Software Development Engineer (Req# 9SAUD4)** Dsgn & dvlp algrthms for img pccssng for photography sys.

**Software QA Engineer (Req# A5549S)** Dev tst plans & implmnt tst autmtion for Core Srvc APIs & bck-end.

**Software Development Engineer (Req# 9UD4A6)** Des, dev & implmnt SW for wireless telecomm sys.

**Product Design Engineering Manager (Req# 9UPTL4)** Mng dvmnt & prctce of analytical methods based on Multiphysics modeling & simulation.

**Software Engineer Applications (Req# 9U52R4)** Instrmnt prodctn deploymnts & monitor prodctn deploymnts of cntnt gnration.

**Hardware Development Engineer (Req# 9D6NNC)** Dsgn, dev, & assess dsply HW for inclusion n Apple prdcts. Travel Req'd 15%

**Software Quality Assurance Engineer (Req# 9VSSQ2)** Tst mltple srvcis in dstrbtud envirmnt usng web & svr-side tst methodologies for Apple Online Store.

**Software Development Engineer (Req# A3Z35K)** Res, des, dev, impl and debug geospatial database sw.

**Product Design Engineer (Req# 9UGT3S)** Rsrch, dsgn & dev sltns for audio prods. Travel req'd 20%.

**Software Development Engineer (Req# ADANLX)** Dev machine learnng algorithms & infrstrctre to imprve Apple map route quality.

**Software Development Engineer (Req# 9WWUPJ)** Research, design, & debug reactive web servcs.

**Software Development Engineer (Req# 9MGW5R)** Des, dev & debug SW to control NAND flash devices by implementing firmware for ASIC controllers.

**Software Engineer Applications (Req# AARS4S)** Review HW, SW infrastructure, & app function for optimization & identify prfrmnc bottlenecks.

**Systems Design Engineer (Req# 9DH2DV)** Des & dev RF calibration & test algorithms for telecomm sys. Travel req 30%.

**Software Development Engineer (Req# A3S24P)** Des & dev SW for Continuous Integration & Delivery cloud services.

**Software Development Engineer (Req# 9F4SJM)** Des, dev, & mntn complex SW sys & processes that supp the op of Apple Internet Services.

**ASIC Design Engineer (Req# A5HUFQ)** Interface w/all disciplines to get fcntrl prods to millions of customers quickly.

**Systems Design Engineer (Req# 9XR3MB)** Test cellular telephony fnc of iOS devices. Travel req: 30%

**Software Development Engineer (Req# 9ARUYT)** Analyze the performance of GPU workloads.

**Software Quality Assurance Engineer (Req #9JFPF2)** Conduct SW testing of product feature & contents.

**Human Factors Design Engineer (Req# A4K3PL)** Identify data needs to drive eng's des inquiries.

**Systems Design Engineer (Req# 9T93KV)** Des & dev iOS, Android, & other internal tils for fld eng & wireless sys activities.

**Software Development Engineer (Req# ABCSJ7)** Prfrm Maps dta analysis of mltns of records & report rslts to users & stakeholders.

**Engineering Project Coordinator (Req# 9TV5JQ)** Implmnt dsgn & validation of core module tech.

**Software Engineer Applications (Req# 9VCS5A)** Dsgn, implmnt, & mntn robust cross-platform SW sltns in order to meet Apple's cntnt protctn needs (server, desktop, & embedded).

**Software Development Engineer (Req# 9TF4S2)** Dsgn & dvlp SW algrthms for natural lang process-

ing, with a focus on automtc speech recognntn.

**Software Development Engineer (Req# AF4VYQ)** Automate the dev tools infrastructure & infra provisioning process.

**Software Development Engineer (Req# A333GA)** Dsgn & dvlp SW for interactive backend tools.

## Apple Inc. has the following job opportunities in Newark, CA:

**Systems Programmer (Req# 9FNPSS)** Support, maintain, & operate workload & process automation tools & systems.

## Apple Inc. has the following job opportunities in Austin, TX:

**Business Systems Analyst (Req# 9V62NM)** Analyze SAP & ABAP data processing platforms to implement & improve comp sys.

Refer to Req# & mail resume to Apple Inc., ATTN: D.W., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/ disability/vets.

## CAREER OPPORTUNITIES

**SR. CONSULTANT.** Job location: Miami, FL & any other unanticipated locations in U.S. Travel Required. Duties: Resp. for performing current state assessment of client's Procure to Pay ops. & Supply Chain orgs. using defined ops. taxonomy, Capability Maturity Model workshops, demand & supply planning techniques, & e-Procurement tools such as Coupa & Ariba. Perform end-to-end Spend Analysis for clients using BIQ software. Identify improvement opps. in supply chain, sourcing & procurement by developing business cases, using financial modelling

techniques, performing data analytics, & bldg.. roadmaps using Microsoft Office Suite & Thinkcell. Resp. for project scheduling, tracking, managing resources, executing client meetings, & risk mgmt. Resp. for develop. of business proposals by conducting co. research using CapitalIQ tool. Requires: M.S. degree in Comp. Sci., Bus. Admin., Eng. or related field & 3 yrs. exp. in job offered or 3 yrs. exp. as a Consultant, Eng. Assoc. or Tech. Staff. Will accept B.S. (or foreign equiv.) & 5 yrs. exp. in the comp. ind. in lieu of M.S. & 3 yrs. exp. Concurrent exp. must incl.: 1 yr.

exp. assessing supply chain; & 1 yr. exp. using CapitalQ and BIQ. Send resume (no calls) to: Michelle Ramirez, The Hackett Group, Inc., 1001 Brickell Bay Dr., Suite 3000, Miami, FL 33131.

**FIREEYE, INC.** has openings for **Senior Software Engineers** in Dallas, TX: Develop the next generation network security and threat management suite. To apply, mail resume to FireEye Inc., Attn: HV, 14151 Newbrook Drive, Ste 450, Chantilly, VA 20151. EOE. Please reference Job#10760.367.

### Apple Inc. has the following job opportunities in Cupertino, CA:

**Systems Design Engineer (Multiple Positions Open) (Req# MSPTJG)** Perform field testing on pre-release HW. Travel required 25%.

**Firmware Engineer (Req# A4Z4BZ)** Des & dev algrtms for Human Interface Devcs.

**Hardware Development Engineer (Req #9TF4YS)** Rsch, des, dev, implmt, test & validate touch sensor for consumer elec dvc's.

**Software Engineer Applications (Req# 9TAU2C)** Des & implmt highly scalable datacenter mngmt infra to sup Apple's global Internet srvcs.

**Automation Engineer (Req# 9F7U6P)** Des & implmnt automation, robotics, prod'tn & manufctr'g sys for prod'tn of iPhones, iPads & other Apple prods. Travel required 30%.

**Software Development Engineer (Req# 9XDW5P).** Des & dev dig personal assist for mobile dvcs.

**Global Supply Coordinator (Req# 9Y9R59)** Idntfy, dev,& handle extng & potential Apple mchncl suppliers. Travel Req'd 30%

**Software Development Engineer (Req# 9Y2QH3)** Respsnble for the des & dev of algorithms & SW for comptr vision systs.

**Hardware Development Engineer (Req# A4SRGM)** Des therml & powr cntrl systems for Apple devcs.

**Hardware Development Engineer (Req# 9TWR3Q) (Multiple Positions)** Dsgn & dev camera modules for cnsurm elctrcn dvcs. Travel Req'd 20%

**Software Development Engineer (Req# 9T9NVH)** Build highly-optimized vector content for Maps app

**Software Development Engineer (Req# A6CPTE)** Write server-side code & bld data pipelines to create robust, high vol prod apps.

**Systems Design Engineer (Req# 9TU2G9)** Dsgn & evaluate Radio Frequency sys for iPhones & iPads.

**Systems Design Engineer (Multiple Positions) (Req# 9TATYW)** Dev & optimize RF auto sys used on Apple's new products including iPhones, iPods, iPads & others. Travel req: 25%

**Software Development Engineer (Req# 9FD4G6)** Anlyz comp HW & SW probs, determine cause, assess probs risk & priority lvls, & determine app sols for internal eng teams.

**Hardware Development Engineer (Req# 9TF4ZP)** Def test req and coverage, calib strategy, & alg of sensing devices in camera tech. Travel req: 30%.

**Hardware Development Engineer (Req# 9T7VMG)** Dev metrology HW & SW for camera lens testing.

**Software Engineer Systems (Req# A5FQPB)** Des & dev SW for video encoder in embedded real-time sys platform.

### Apple Inc. has the following job opportunities in Austin, TX:

**ASIC Design Engineer (Req# 9PV2XQ)** Model tasks req for the dvlpmnt & verifictn of GPU dsgn.

Refer to Req# & mail resume & transcript to Apple Inc., ATTN: D.W., 1 Infinite Loop 104-1GM, Cupertino, CA 95014. Apple is an EOE/AA m/f/disability/vets.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE DEVELOPERS**

positions in **Orlando, FL.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to [ulhas.pinjarkar@oracle.com](mailto:ulhas.pinjarkar@oracle.com), referencing 385.18436.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL ANALYST-SUPPORT**

positions in **Orlando, FL.**

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to [patrick.flynn@oracle.com](mailto:patrick.flynn@oracle.com), referencing 385.20172.

Oracle supports workforce diversity.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**CLIENT SUCCESS SPECIALIST**

positions in **Redwood Shores, CA.**

Job duties include: Accelerate the realization of customers desired business outcomes by aligning sponsor objectives with the capabilities of the Oracle Cloud Platform. Travel to various unanticipated sites throughout the United States required. May telecommute from home.

Apply by e-mailing resume to [craig.lockwood@oracle.com](mailto:craig.lockwood@oracle.com), referencing 385.16068.

Oracle supports workforce diversity.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**DATABASE ADMINISTRATOR**

positions in **Redwood Shores, California.**

Job duties include: provide enterprise-wide, database administration support for production systems and provide DBA services to application development teams, including database design, database generation, coding, and/or database production support. May telecommute from home.

Apply by e-mailing resume to [jeff.bates@oracle.com](mailto:jeff.bates@oracle.com), referencing 385.17054.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL ANALYST**

positions in **Lehi, Utah.**

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs. Offer strategic technical support to assure the highest level of customer satisfaction.

Apply by e-mailing resume to [ronald.hahn@oracle.com](mailto:ronald.hahn@oracle.com), referencing 385.19787.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE DEVELOPER**

positions in **Broomfield, CO.**

Job duties include: Design, develop, troubleshoot and/or test/QA software.

Apply by e-mailing resume to [matthew.aitkenhead@oracle.com](mailto:matthew.aitkenhead@oracle.com), referencing 385.18711. Oracle supports workforce diversity.

**TECHNOLOGY**

**Oracle America, Inc.**

has openings for

**DATA  
SCIENTIST**

positions in **Broomfield, CO.**

Job duties include: Interact with product and service teams to identify questions and issues for data analysis and experiments.

Apply by e-mailing resume to [audrey.thompson@oracle.com](mailto:audrey.thompson@oracle.com), referencing 385.20210.

Oracle supports workforce diversity.

**Oracle America, Inc.**

has openings for

**TECHNICAL  
ANALYST-  
SUPPORT**

positions in **Colorado Springs, CO.**

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to [brad.ericksen@oracle.com](mailto:brad.ericksen@oracle.com), referencing 385.19910.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE  
DEVELOPMENT  
MANAGER**

positions in **Irving, TX.**

Job duties include: Manage a team that designs, develops, troubleshoots and debugs software programs for databases, applications, tools, networks etc. May telecommute from home.

Apply by e-mailing resume to [sanjay.thakur@oracle.com](mailto:sanjay.thakur@oracle.com), referencing 385.18959.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL  
ANALYST**

positions in **Orlando, FL.**

Job duties include: Deliver solutions to the Oracle customer base while serving as an advocate for customer needs. Offer strategic technical support to assure the highest level of customer satisfaction.

Apply by e-mailing resume to [jeff.gregory@oracle.com](mailto:jeff.gregory@oracle.com), referencing 385.19735.

Oracle supports workforce diversity.

**TECHNICAL**

**Oracle America, Inc.**

has openings for

**TECHNICAL  
ANALYST**

positions in **Lehi, UT.**

Job duties include: Deliver post-sales support and solutions to the Oracle customer base while serving as an advocate for customer needs.

Apply by e-mailing resume to [patrick.flynn@oracle.com](mailto:patrick.flynn@oracle.com), referencing 385.19662.

Oracle supports workforce diversity.

**SOFTWARE**

**Oracle America, Inc.**

has openings for

**SOFTWARE  
DEVELOPER**

positions in **Naples, FL.**

Job duties include: Design, develop, troubleshoot and/or test/QA software. As a member of the software engineering division, apply knowledge of software architecture to perform tasks associated with developing, debugging, or designing software applications or operating systems according to provided design specifications.

Apply by e-mailing resume to [chetan.dihenia@oracle.com](mailto:chetan.dihenia@oracle.com), referencing 385.18314.

Oracle supports workforce diversity.

## TECHNOLOGY

Help build the next generation of systems behind Facebook's products.

# Facebook, Inc.

currently has the following openings:

Openings in **Menlo Park, CA (multiple openings/various levels):**

**Application Product Manager (8384)** Create, monitor, and maintain Hyperion software and database applications including Planning, Essbase, DRM, FDM, Command Center and other related automated processes. **Application Engineer (8794)** Develop and maintain integrated, scalable, solutions and applications focused on data analytics and reporting. **Software Engineer (4819)** Create web and/or mobile applications that reach over one billion people, & build high volume servers to support our content. **Product Manager (5374)** Engage in product design and technical development of new products. Lead the ideation, technical development, and launch of innovative products. **Front End Engineer (9697)** Work with Product Designers to implement the next generation of Company's products. Build efficient and reusable front-end abstractions and systems. **Data Specialist (5986)** Apply your expertise in quantitative analysis, data mining, and the presentation of data to see beyond the numbers and understand how our users interact with our core products. Occasional travel required to various unanticipated locations throughout the U.S, India, and Ireland. **Data Engineer (9438)** Design and build data reporting and visualization needs for a product or a group of products. **Production Engineer (7849)** Participate in the design, implementation and ongoing management of major site applications and subsystems. **SMB Analyst (9105)** Use data analysis to understand customer profiles, produce reports to track our business, and build models to provide insight into the Small & Medium Business customer base. **Partner Engineer (8501)** Define and develop software solutions to drive adoption of the Facebook mobile platform. Position requires national and international business travel to unanticipated locations. **Production Engineer (8480)** Participate in the design, implementation, and ongoing management of our large MySQL database cluster. **Research Scientist (6249)** Research, design, and develop new optimization algorithms and techniques to improve the efficiency and performance of Facebook's platforms.

Openings in **Seattle, WA (multiple openings/various levels):**

**Software Engineer (SWE417-B)** Create web and/or mobile applications that reach over one billion people & build high volume servers to support our content. Bachelor's degree required. **Software Engineer (SWE417-M)** Create web and/or mobile applications that reach over one billion people & build high-volume servers to support our content, utilizing graduate level knowledge. Master's degree required. **Data Engineer (8562)** Design and build data reporting and visualization needs for a product or a group of products.

Openings in **New York, NY (multiple openings/various levels):**

**Software Engineer (NYSWEB417)** Create web and/or mobile applications that reach over one billion people & build high volume servers to support our content. Bachelor's degree required. **Software Engineer (NYSWEM417)** Create web and/or mobile applications that reach over one billion people & build high-volume servers to support our content, utilizing graduate level knowledge. Master's degree required. **Technical Program Manager (2496)** Manage technical projects to develop products. Responsible for managing cross-functional projects for Front End Engineering Design (FEED). Position requires national travel to unanticipated sites 5-10% of the time. **Data Engineer (9345)** Design and build data reporting and visualization needs for a product or a group of products.

Openings in **Dallas, TX (multiple openings/various levels):**

**Software Engineer (8873)** Help build the next generation of tracking technology behind Facebook's Virtual Reality products, create software that will enable over one billion people to experience high quality immersive virtual reality.

Mail resume to: Facebook, Inc. Attn: SB-GIM, 1 Hacker Way, Menlo Park, CA 94025.

Must reference job title & job# shown above, when applying.



PREFERRED PLUS



TRAINING  
& DEVELOPMENT



RESEARCH



BASIC



STUDENT

# New Membership Options for a Better Fit

And a better match for your career goals. Now IEEE Computer Society lets you choose your membership — and the benefits it provides — to fit your specific career needs. With four professional membership categories and one student package, you can select the precise industry resources, offered exclusively through the Computer Society, that will help you achieve your goals.



IEEE  computer society

Learn more at [www.computer.org/membership](http://www.computer.org/membership).

# Achieve your career goals with the fit that's right for you.

Explore your options below.



Select your membership	Preferred Plus		Training & Development		Research		Basic		Student
	\$60 IEEE Member	\$126 Affiliate Member	\$55 IEEE Member	\$115 Affiliate Member	\$55 IEEE Member	\$115 Affiliate Member	\$40 IEEE Member	\$99 Affiliate Member	\$8 Does not include IEEE membership
Computer magazine (12 digital issues)*	✓		✓		✓		✓		✓
ComputingEdge magazine (12 issues)	✓		✓		✓		✓		✓
Members-only discounts on conferences and events	✓		✓		✓		✓		✓
Members-only webinars	✓		✓		✓		✓		✓
Unlimited access to <i>Computing Now</i> , computer.org, and the new mobile-ready myCS	✓		✓		✓		✓		✓
Local chapter membership	✓		✓		✓		✓		✓
Skillsoft's Skillchoice™ Complete with 67,000+ books, videos, courses, practice exams and mentorship resources	✓		✓						✓
Books24x7 on-demand access to 15,000 technical and business resources	✓		✓						✓
Two complimentary Computer Society magazine subscriptions	✓				✓				
myComputer mobile app	30 tokens				30 tokens				30 tokens
Computer Society Digital Library	12 FREE downloads		Member pricing		12 FREE downloads		Member pricing		Included
Training webinars	3 FREE webinars		3 FREE webinars		Member pricing		Member pricing		Member pricing
Priority registration to Computer Society events	✓								
Right to vote and hold office	✓		✓		✓		✓		
One-time 20% Computer Society online store discount	✓								

\* Print publications are available for an additional fee. See catalog for details.

[www.computer.org/membership](http://www.computer.org/membership)



# Endless Possibilities...

*and countless risks*

Let the IEEE Member Professional Liability Insurance Program Help Minimize Your Exposures.

**Y**our career in the technology field offers you endless possibilities. From the broad spectrum of technology specialization to the higher-earning income potential, your future ahead is very bright. *But what if something happens to get in your way?*

**You'll face countless professional risks.** No matter how well you design it or how accurate your advice is, you can be sued. Whether a claim is frivolous or justified—your reputation, assets and career could be at risk. And it could be costly to defend yourself.

**IEEE understands your profession like no other!** That's why through the IEEE Member Professional Liability Insurance Program, IEEE members can access enhanced coverage options specifically designed to help minimize their exposures and protect them against the many risks they face. And the Choice Platform provides you with options from industry-leading insurers.\*

- Features risk management and legal services such as contract review, pre-claim prevention and access to a risk management & claims hot-line.
- Includes a broad definition of computer systems and information technology-related professional services.
- Wide range of benefit limits and deductible options available.

**Protect your career and your future with this important professional liability coverage today!**

**1-800-375-0775 | [IEEEinsurance.com/IEEEPL](http://IEEEinsurance.com/IEEEPL)**

Canadian IEEE members can visit [www.ieeeinsurance.com/canadapl](http://www.ieeeinsurance.com/canadapl) for more information about the insurance program brokered by Marsh Canada Limited and underwritten by Certain Underwriters at Lloyd's of London.

\*The IEEE Member Professional Liability Insurance Program with the Choice Platform is available to active IEEE members who reside in the U.S. IEEE members in Canada (excluding Quebec) have access to the IEEE Member Professional Liability Insurance Plan through Marsh Canada Limited. Coverage options may vary or may not be available in all states. Not all plan features will be available under all carriers or plan options.

Program Administered by Mercer Health & Benefits Administration LLC

In CA d/b/a Mercer Health & Benefits Insurance Services LLC  
AR Insurance License #100102691  
CA Insurance License #0G39709

78523 (5/17) Copyright 2017 Mercer LLC. All rights reserved.