

On Tap: An Hour of Code • Michael Swaine • Teaching kids to code, getting Clojure, rethinking retrospectives, and more.

Choice Bits: Drones, Bots, and Bitcoins • Michael Swaine • A month recaptured in tweets and news bits.

Rothman and Lester: Landing the Dream Job • Johanna Rothman & Andy Lester • How to land that dream job. • career

Teaching Kids to Code: Reflections on An Hour of Code • David Bock • David reflects on a week teaching kids to code, and asks why we can't make every week Computer Science Education Week. • kids, teaching

Automating Android for Your Listening Pleasure: Books, Music, Messages • Mike Riley • Mike teaches his Android to walk and chew gum at the same time. • android, mobile, music

Clojure and EDN: Closure's Data Storage and Interchange • Michael Bevilacqua-Linn • The ins and outs of EDN • clojure, functional, language

Leveraging Retrospectives: You Have to Look Back in Order to See Ahead • Jeff Cohen • Retrospectives are not just for teams. • agile

The Roots of Revolution: Two Stories from Personal Computing's Early Days • Michael Swaine • The counterculture of the 1960s definitely influenced the personal computer revolution. Exactly how is complicated. • history

Antonio on Books • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: Some Common Sense Regarding Bitcoins • John Shade • John takes on the Bitcoin phenomenon and offers a modest, though extraterrestrial, proposal. • humor

Rear Window: A Parting Shot • Michael Swaine • Harry Garland and Roger Melen founded Cromemco, an early personal computer company. • history



On Tap: A Couple of Issues • Michael Swaine • In which we introduce not only the issue before you but a special issue as well.

Choice Bits: Bursts from the Twitterhose • Michael Swaine • A month recaptured in tweets.

Mac & Me: Thirty Years Before the Mouse • Michael Swaine • Michael reflects on 30 years of the Mac. • history

Rothman and Lester: How to Manage Your Boss • Johanna Rothman & Andy Lester • You say your boss is driving you crazy with questions and micromanagement? Try these insights for getting the boss off your back. • career

A Closer Look At Clojure Sequences: With A Dash Of Light Lisp History And A Side Of Laziness • Michael Bevilacqua-Linn • Clojure has come a long way from its origins in Lisp, but it manages to address a rich collection of data structures with a common Lisplike interface. • clojure, functional, language

Web-Based Intrusion Prevention with Repsheet: Dealing with the Bad Guys • Aaron Bedra • Your web app is under attack. What are you doing about it? • web, security

Core Data As Data Structure Service: Persistence Conquers All • Joshua Smith • Core Data makes all the hard parts go away. • cocoa

Lego League: Lessons Learned: Teaching Kids to Code • Seb Rose • Our series on teaching kids to code continues with lessons learned from coaching and judging a coding competition. • kids, teaching

Antonio on Books: Don't Make Me Think (and Other Excellent Advice) • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: The Parable of the Streetlight • John Shade • John shares a parable. • humor

Rear Window: A Parting Shot • Michael Swaine • Happy 30th birthday, Mac! • history



On Tap: Sound Advice on Sound • Michael Swaine • Designing sound, writing Clojure code, teaching kids to program, managing a job transition, and more.

Choice Bits: Bursts from the Twitterhose • Michael Swaine • A month recaptured in tweets.

Rothman and Lester: How to Transition to a New Role • Johanna Rothman & Andy Lester • Job transitions, whether within the same company or to a new company, are fraught with peril. • career

Constraints and Freedom: “29638 Bytes Free” — What a Gift! • Jimmy Thrasher • It was Igor Stravinsky who said, “The more constraints one imposes, the more one frees one’s self of the chains that shackle the spirit.” The same insight also applies when teaching kids to program. • kids, teaching

Sound Design Principles: Making Sound Matter • Tony Hillerson • Your app probably doesn’t need sound. Not really. And that’s exactly why you should approach sound from a design sensibility. • design, sound

Coding Unplugged: A Case Study in Teaching Kids to Code • Fahmida Y. Rashid • Most child-oriented programming initiatives target kids in the 9–16–year-old range. But their kids were just five and eight. They needed to get creative. • kids, teaching

Wrapping Things Up: Building Macros in Clojure • Michael Bevilacqua-Linn • Michael wraps up his series on the Clojure language, and to his credit, manages to avoid lame references to “getting closure.” Your editor, not so much. • clojure, functional, language

Poverty on Parade: An Old-Time Radio Show from the Days of the Dot-Com Crash • Michael Swaine • Your editor reprises an homage to Bob & Ray and waxes nostalgic over the dot-com crash. • humor

Antonio on Books: Cowboy Coding? • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: Wired’s Hip Replacement • John Shade • John reveals his true feelings about the Valentine issue of Wired. • humor

Rear Window: Jim Weirich, 1956-2014 • Michael Swaine • Jim Weirich, 1956-2014 • history, agile



On Tap: Outside Your Box • Michael Swaine • You're a software developer. But you're also, at times, a technical writer, a journalist, a member of a technical community, a designer, a teacher. And you don't want to suck at any of those things.

Choice Bits: News and Views • Michael Swaine • How did this happen? Who is responsible? We plead the First.

Rothman and Lester: Developing Your Writing Skills • Johanna Rothman & Andy Lester • Every professional is also a writer, but few are professional in their writing. • career

The New Feudalism and the Common Good: Returning Beauty, Truth, and Goodness to Technology • Tom Mahon • We may be facing a critical decision point, when we settle who is to be master, the tools or the makers. • career

The Selfish Teacher: Teach a Kid to Code—for Your Own Good • Michael Swaine • How teaching a kid to code could be a wonderful thing to do—for yourself. • kids, teaching

Sound Design in Practice: Makin' Sonic Bacon with Pure Data • Tony Hillerson • Getting the hisses and pops just right. Your app never sounded so good. • design, sound

The Elegance of C: A Lesson in Style • David Lowe • It is rare that we see C code that so fully embodies the true nature of its language. And as a bonus, this code proves that even a very short program can embody great depths and subtlety. • c, language

XP is the Mac of Agile: Why XP Will Be More Important than Ever • Jonathan Rasmusson • Don't sell Extreme Programming short. It could be just coming into its own. • agile, xp

The Discovery of Programming Languages: The PragPub History of Programming Languages Series • Dan Wohlbruck • Launching a new series on the history of programming languages. • language, history

Antonio on Books: More is better. Lean is good. • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: I Am Satoshi Nakamoto • John Shade • John Shade reveals the shocking truth about the creation of Bitcoin. • humor



On Tap: Agile Today • Michael Swaine • The state of agile development, coaching and teaching programming.

Choice Bits: Yak Overflow • Michael Swaine • Ambient discoverability and other perspectives.

Rothman and Lester: The Mechanics of Job Hunting • Johanna Rothman & Andy Lester • In the process of searching for a job you will amass a ton of data. Your chances of finding a job you'll love increase with your skill at managing all that data. • career

Process Agnosticism: Getting the Agile Train Back on the Track • Allen Holub • If you think you know how to “do” Agile, you’re wrong. • agile

The Agile Mindset: Reflections on Being Agile • Jonathan Rasmusson • The Agile Samurai reflects on the agile way of thinking. • agile

The Next Seven Coaching Patterns: Change Begins with Yourself • Portia Tung • Seven (more) proven patterns for improving your coaching. • agile, coaching, patterns

Spinning Donuts • Chris Strom • Start with dessert and save the green beans for later. • kids, teaching

Programming Unit Record Machines: The PragPub History of Programming Languages Series • Dan Wohlbruck • A story of chicken salad and a massive data processing project that had gotten completely out of control. • language, history

Antonio on Books: Thirty New Books • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: Invention Intervention • John Shade • John has some advice for the makers of wearable tech. • humor



On Tap: Celebrating Five PragPub Years! • Michael Swaine • About our 5th anniversary issue, plus our special issue on teaching kids to code.

Choice Bits: Sifting the Twittersands • Michael Swaine • Guess how many countries or American States PragPub readers live in. Or consult our readership map.

Why Johnny Can't Be Agile: A Guru Meditation • Andy Hunt • Why adopting agile methods might be harder than you think. • agile

Punk Rock Languages: A Polemic • Chris Adamson • In an era of virtual machines and managed environments, C is the original Punk Rock Language. • c, language

Rothman and Lester: What You're Getting Hired For • Johanna Rothman & Andy Lester • The hiring manager may not know what the company needs, but you need to find out. • career

Building Your Battlestation: The New Old Joys of Making Your Own PC • Mike Riley • Your Battlestation is your PC work/play space, designed and built by you, from GPU selection to LED placement. • hardware

Functional Thinking For The Imperative Mind: Dipping Your Toe in the Functional Water • Michael Bevilacqua-Linn • How does functional thinking differ from the imperative kind? • functional, language, java

Programming Elixir: A Gentle Introduction • Dave Thomas • Elixir is a Ruby-like language on the Erlang VM. • elixir, functional, language

Chad Fowler on Ruby: An Interview with a Ruby Pioneer • Michael Swaine • A legend in the Ruby community reflects on the early days of Ruby and speculates on its future. • ruby, language

Estimation is Evil: A Warning • Ron Jeffries • What you don't know can hurt you, especially when you convince yourself that you do know it. • agile, estimation

Responsive Design: A Modest Proposal • Kent Beck • Kent reflects on his years designing software and concludes that we need a new approach. • agile, design

Black Friday: A Tale from the Ops Side • Michael Nygard • Michael gives a blow-by-blow account of dealing with "the only load test that matters." • devops

Programming and the ENIAC: The PragPub History of Programming Languages Series • Dan Wohlbruck • Dan brings his history of programming languages up to the ENIAC era. • history, language

Antonio on Books: Thirty New Books • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: Toward a Better Class of Manifestos • John Shade • John surveys the leading examples of software development manifestos and makes some suggestions. • humor



On Tap: The Swift Issue • Michael Swaine • Apple has shaken up OSX and iOS development with a new language. We're on it.

Exiting with Grace: How to Leave Your Old Job Well • Johanna Rothman & Andy Lester • How to leave a job and be well-remembered. Plus: Why you should never talk in an exit interview. • career

Swift: Didn't See That Coming: The Roots of Apple's New Language • Mark Chu-Carroll • Apple has shaken up iOS and OS X development with a new language. To really understand Swift, we need to see where it came from. • swift, language, functional

Swift: What You Need to Know: A Deep Dive into its Key Features • Mark Chu-Carroll • Mark lays out the key object-oriented and functional features of the language. • swift, language, functional

Swift Isn't Perfect: A Few Bumps in the Road • Mark Chu-Carroll • At present, Swift is buggy, is missing key features, and reflects some strange choices. Still... • swift, language, functional

Swift from a TDD Perspective: Do Playgrounds Play Well with TDD? • Ron Jeffries • Apple's new language features a nifty way to test out ideas — but does it undermine test-driven programming practices? • swift, language, testing, functional

Functional Thinking in Swift: Mastering the Functional Tools • Tony Hillerson • Tony uses Swift to explore the functional approach to programming. • swift, language, functional

Some Inconvenient Truths About Software: What Every Business and Software Design Team Should Know • Jonathan Rasmusson • You will always have more to do than time and money will allow. • agile

The First Stored Program: John von Neumann and the Language of EDVAC: The PragPub History of Programming Languages Series • Dan Wohlbruck • The EDVAC used four bits to represent an instruction's OP code, so its machine language could have no more than sixteen instructions. John von Neumann decided he only needed twelve. • language, history

Antonio on Books: Thirty New Books • Antonio Cangiano • All the new tech books of note. • books

Shady Illuminations: Who You Gonna Call? • John Shade • John decides that Elon Musk is just horsing around. • humor



On Tap: From FLOW-MATIC to Swift and Elixir • Michael Swaine • This month we compare Apple's new Swift language to other languages, tweak Vim, delve into history, offer career and job advice, and pick on Microsoft's new CEO.

Choice Bits: Plucked from the Twitterstream • Michael Swaine • We follow Twitter so you don't have to.

Rothman and Lester: The Fit • Johanna Rothman & Andy Lester • How do you determine if the job is right for you? And how do you convince the hiring manager? • career

Protocols in Swift, Ruby, and Elixir: Comparing Three Approaches • José Valim • The creator of the Elixir language compares it to Apple's new language. • elixir, swift, ruby, language, functional

Writing a Vim Syntax File for Swift: Making Vim Swift-Friendly • Ben Klein • Ben combines two interests—tweaking the Vim editor and exploring new languages—to show you how to make Vim Swift-friendly. • swift, vi

Two Guys in a Garage: Three Stories of Silicon Valley Startups • Michael Swaine • It's a recurring theme in technology: two guys start a company in a garage, and change the world. Here are three such stories. • history

The UNIVAC 1: From Machine Language to FLOW-MATIC: The PragPub History of Programming Languages Series • Dan Wohlbruck • Grace Hopper, FLOW-MATIC, and the beginning of real programming. • history

Antonio on Books: Thirty New Books • Antonio Cangiano • Concurrency. Parallelism. What are you waiting for? Antonio looks at all the new tech books of note. • books

Shady Illuminations: The Cloudy Future • John Shade • John thinks Satya Nadella is a master of obscurantism. • humor



On Tap: Delivering Value • Michael Swaine

Swaine's World: Retweets, Newsbits, and Frivolity • Michael Swaine • Curated curiosities from the collection of our editor.

Defining Value: Value Is What You Like • Ron Jeffries • Searching for the deep simplicities in software development, Ron asks and answers the question, "What is value?" • agile

Knowing Your Value: And Defining Your Value to Your Boss • Johanna Rothman & Andy Lester • To convince employers and managers and potential employers and managers of your worth to their company or team, you have to understand your value as an employee. • career

When TDD Doesn't Matter: What Do You Value? • Kent Beck • Every value that factors into your decision is a dimension. Making a decision that weighs scope vs clarity vs frequency of feedback means finding a point of maximum value in a 3-D space. • agile, testing

The Value of First-Class Functions in Swift: They Make It Easy to Write Powerful and Flexible Code • Daniel Steinberg • Daniel offers a gentle introduction to Apple's new iOS and OS X development language, Swift. In the process he dips into first-class functions and functional programming. • swift, functional, language

Giving Back: How You Can Pass on Your Expertise to Kids • David Bock • In this latest in our series on teaching kids to code, David shares his experience and invites you to join in. • kids, teaching

FORTRAN: The First High-Level Language: The PragPub History of Programming Languages Series • Dan Wohlbruck • Dan is exploring the history of programming languages in this series, and this month he looks at the origin of Fortran. • history, language, fortran

Antonio on Books: Thirty New Books • Antonio Cangiano • Antonio looks at all the new tech books of note. • books

Shady Illuminations: A Burlesque History • Michael Swaine • Mike thinks you might enjoy this burlesque history of Ada Loveless and Charles Babbage. • humor, history



On Tap: PureScript, Ruby, and COBOL • Michael Swaine

Swaine's World: Retweets, Newsbits, and Frivolity • Michael Swaine

Johanna Rothman on Your Career: Don't Fall into One of These Three Common Job Search Traps • Johanna Rothman • At one time or another you've probably fallen into one of these common job search traps. • career

Generative Testing With PureScript: A Hands-on Introduction to This New Language • Phil Freeman • PureScript is a simple functional language that compiles to JavaScript. Let its creator take you on a tour. • prescript, language, javascript, testing, functional

Ruby Performance Tuning: Iterators Hide a Multitude of Performance Sins • Alexander Dymo • Speed up your Ruby code drastically with these techniques. • ruby, language

The Right Thing in the Wrong Order: How I Messed Up a Medium-Scale Refactoring • Brian Marick • The best programmers are continually learning how to improve in their craft — and often generously share what they learn. • agile, clojure

When COBOL was King: The PragPub History of Programming Languages Series • Dan Wohlbruck • A look back at the life and times of the most successful programming language ever. • history, language, cobol

Antonio on Books: Thirty New Books • Antonio Cangiano • Antonio looks at all the new tech books of note. • books

Shady Illuminations: Back to the Future • John Shade • A 2001 vision of wearable technology in the future, i.e., 2007. • humor



On Tap: Career Advice • Michael Swaine

Swaine's World: Retweets, Newsbits, and a Puzzle • Michael Swaine • What happened in October besides Apple's announcements. Plus Apple's announcements.

Rothman and Lester: Time Is on Our Side... or Is It? • Johanna Rothman & Andy Lester • Some days are more productive than others. Why is that? Johanna and Andy share some techniques to make every work day more productive. • career

The Philosophy of Great Customer Service • Derek Sivers • career

The Many Faces of Swift Functions • Natasha Murashev • swift, objective-c, language, functional

Putting Down the Tools: Avoiding the No. 1 Mistake All New Software Managers Make in a Crisis • Marcus Blankenship • Don't turn crisis management into crisis generation. • career

Unix and C: The PragPub History of Programming Languages Series • Dan Wohlbruck • In the first of a two-parter, Dan looks at the C language, its roots, and its influence. • history, language, c

Antonio on Books: Thirty New Books • Antonio Cangiano • Antonio looks at all the new tech books of note. • books

Shady Illuminations: Design Rules • John Shade • John returns from vacation with some thoughts on why some things are the way they are. • humor



On Tap: It's All About Messaging • Michael Swaine

Swaine's World: Retweets and a Puzzle • Michael Swaine • What our editor is paying attention to this month. Looks like tweets.

Rothman and Lester: The New Guy • Johanna Rothman & Andy Lester • Have you been stuck mentoring the new guy? Or are you, on this job, the new guy? Her's how to play that. • career

New Manager's Playbook: Having Difficult Conversations with Employees • Marcus Blankenship • Launching a new column by Marcus on what to do when you become a manager. • career

Your Object Model Sucks: Resource Oriented Computing Is What Object Oriented Programming Wanted to Be • Ron

Hitchens • The big idea in Object Oriented Programming was always messaging, not objects or classes. Resource Oriented Programming gets that right. • resource-oriented computing

Microservices Is not the Answer: Resource Oriented Computing Is What Microservices Aspires to Be • Brian Sletten • Microservices has the right idea. It just doesn't take it far enough. • resource-oriented computing

Introducing Resource Oriented Computing: It's All About Messages • Tom Geudens • There's a program that's been running for 25 years, that scales like nobody's business, and that's based on open protocols. Can your code measure up to that? • resource-oriented computing

Turning APIs into Resources • Tom Geudens • resource-oriented computing

Antonio on Books: Thirty New Books • Antonio Cangiano • Antonio looks at all the new tech books of note. • books

