Empirix
Be Confident.

QA/IT professionals: do you know about QAZone?

QAZone by Empirix

This new online community from Empirix is exclusively for you!

To start collaborating with industry peers, register for a free QAZone account now at: http://qazone.empirix.com.

About QAZone

This new online community from Empirix is dedicated to fostering communication and collaboration among QA and IT professionals. By joining QAZone, you can learn more about industry trends, QA best practices, Empirix's Web testing and monitoring solutions, as well as get tips on methodology in our knowledgebase, access test code samples or monitoring profiles in our Resource Center, or share your knowledge with others through our Forums.

Empirix, Inc. • 20 Crosby Drive, Bedford, MA 01730 • www.empirix.com • info@empirix.com • 866 EMPIRIX

April 2008 $9.95 www.StickyMinds.com

BETTER SOFTWARE

The Print Companion to StickyMinds.com

GET TO YES
How to win management buy-in

REV UP REUSE
Popular source code search engines

Incremental and Iterative Development

HOW they are different and WHY you should be doing both

IF YOU ARE INTERESTED
IN IMPROVING YOUR TEAM,
THERE ARE ANY NUMBER
OF COMPANIES YOU CAN CALL.

BUT
IF YOU WANT TO
IMPROVE YOUR ENTERPRISE,
THERE IS ONLY ONE.

CALL 1-888-LEAN-244
(1-888-532-6244)

www.netobjectives.com

LEAN-AGILE
CONSULTING
TRAINING
COACHING

NetObjectives
Achieving Enterprise Agility

Lean-Agile Software Development Solutions for Business, Management and Teams.

Take the handcuffs off quality assurance

Empirix gives you the freedom to test *your way.*

Tired of being held captive by proprietary scripting? Empirix offers a suite of testing solutions that allow you to take your QA initiatives wherever you like.

Download our white paper, "Lowering Switching Costs for Load Testing Software," and let Empirix set you free.

www.empirix.com/freedom

empirix
Be Confident.

TechExcel DevSuite

The ALM Platform for Scalable Agile Development

- Integrated requirements management, project planning, implementation tracking and QA testing
- Provides out-of-the-box implementations for Agile and other standard methodologies
- Facilitates close collaboration of globally distributed teams
- Scalable from small teams to thousands of users

A fully integrated suite of best-of-breed ALM products:

- **DevSpec**: Requirements management
- **DevPlan**: Project planning
- **DevTrack**: Implementation and issue tracking
- **DevTest**: QA test management
- **KnowledgeWise**: Knowledge and Digital Asset Management

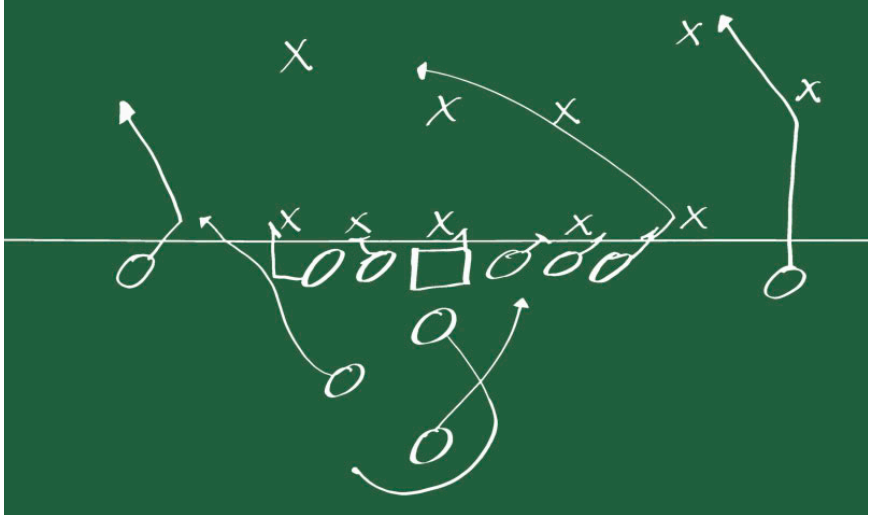
TechExcel

Try DevSuite Live at www.techexcel.com/bettersoftware

www.techexcel.com | 1-800-439-7782

BETTER SOFTWARE

April 2008
Volume 10, Issue 3



Cover Story

INCREMENTAL AND ITERATIVE DEVELOPMENT 24

It can be hard to wrap your head around the differences between incremental and iterative development. But they are different and must be managed differently. Find out why you shouldn't do one without the other. *by Alistair Cockburn*

Features

DEVELOPERS ... START YOUR ENGINES 30

Source code search engines can help you find chunks of reusable code. These reviews of some popular engines can help you rev up reuse in your work.
by Alan Berg

WHERE DO I GO FROM HERE? 36

Software testing is a business and you are the product. Learn ways to market yourself and add value to your career. *by Matthew Heusser*

Columns & Departments

TECHNICALLY SPEAKING 13

A Change Would Do You Good • *by Jonathan Kohl*
Boredom can affect your ability to solve complex technical problems. Discover how change can spice up your software processes.

CODE CRAFT 14

A "D" in Programming, Part 1 • *by Chuck Allison*
In certain company, the topic of favorite programming languages can elicit the same response as other taboo subjects, such as religion and politics. But Chuck's going out on a limb to discuss his new favorite language, D.

TEST CONNECTION 18

Learning the Hardware Lessons • *by Michael Bolton*
Software isn't just about correctness; it's also about solving problems for people. Michael's experience in a hardware store drives that lesson home.

MANAGEMENT CHRONICLES 20

The Art of Persuading Management • *by Naomi Karten*
You can't get your manager to give you what you want if he won't listen to you. These strategies can help you make your case to the powers that be.

THE LAST WORD 47

Software Quality and the Prisoner's Dilemma • *by Paco Hope*
This industry-specific spin on the classical dilemma illustrates the games we play when software quality is at stake.

In Every Issue

Mark Your Calendar 4
Contributors 6
eLightenment 8
Product Announcements 43
10 Things You Might Not Know About ... 46
Ad Index 48

***Better Software* magazine**—The print companion to StickyMinds.com brings you the hands-on, knowledge-building information you need to run smarter projects and deliver better products that win in the marketplace and positively affect the bottom line. **Subscribe today to get ten issues.**

Visit www.BetterSoftware.com or call 800-450-7854.

StickyMinds.com We invite you to visit StickyMinds.com, the online companion to *Better Software* magazine. StickyMinds.com covers the same pertinent topics as the magazine, putting the power of information at the click of your mouse. Weekly columns, headline-making bugs, hundreds of technical papers, an online tools guide, discussion boards, and so much more make StickyMinds.com your site for 24/7 brainfood to help you build better software.

MARK YOUR CALENDAR

Training Weeks
www.sqetraining.com

Testing
June 2-6, 2008
Chicago, IL

September 8-12, 2008
New York/New Jersey Area

Agile Software Development
May 12-16, 2008
Boston, MA

June 2-6, 2008
Seattle, WA

Software Testing Certification
www.sqetraining.com/stf

May 4-6, 2008
Orlando, FL

May 13-15, 2008
Montreal, QC

May 13-15, 2008
St. Louis, MO

Conferences
STAR*EAST* 2008
Software Testing Analysis & Review
www.sqe.com/stareast

May 5–9, 2008
The Rosen Centre Hotel
Orlando, FL

Better Software Conference & EXPO 2008
www.sqe.com/bettersoftwareconf

June 9–12, 2008
The Venetian
Las Vegas, NV

STAR*WEST* 2008
Software Testing Analysis & Review
www.sqe.com/starwest

September 29–October 3, 2008
Disneyland Hotel
Anaheim, CA

BETTER SOFTWARE

Publisher
Wayne Middleton

Director of Publishing
Holly N. Bourquin

Editorial

Editor
Heather Shanholtzer

Managing Technical Editor
Lee Copeland

Technical Editors
Chuck Allison
Jonathan Kohl
Antony Marcano

Editor, StickyMinds.com
Francesca Matteu

Managing Editor, Multimedia
Joseph McAllister

Production Coordinator
Cheryl M. Burke

Design

Creative Director
Catherine J. Clinger

Advertising

Senior Advertising Sales Manager
Shae Young

Production Coordinator
April Evans

Circulation and Marketing

Marketing Coordinator
Megan Brown

Circulation Coordinator
Jamie Green-Gago

A PUBLICATION OF SOFTWARE QUALITY ENGINEERING

SOFTWARE QUALITY ENGINEERING
ASBPE American Society of Business Publication Editors
BPA WORLDWIDE BUSINESS

CONTACT US
Editors: editors@bettersoftware.com
Subscriber Services: info@bettersoftware.com
Phone: 904-278-0524, 888-268-8770
Fax: 904-278-4380
Address:
Better Software magazine
Software Quality Engineering, Inc.
330 Corporate Way, Suite 300
Orange Park, FL 32073

SOFTWARE TESTING ANALYSIS & REVIEW

The Greatest Software Testing Conference on Earth

STAR EAST

MAY 5–9, 2008

Keynotes by International Experts

James Whittaker
Microsoft

Bharat Mediratta and Antoine Picard
Google

Elisabeth Hendrickson
Quality Tree Software

Tom Wissink
Lockheed Martin

Hugh Thompson
People Security

John Fodeh
Hewlett-Packard

99.7% of 2007 Attendees Recommend STAREAST to Others in the Industry

www.sqe.com/stareast

REGISTER EARLY AND SAVE $200!

SOFTWARE QUALITY ENGINEERING
www.sqe.com

Conference Sponsors:

Capgemini — CONSULTING.TECHNOLOGY.OUTSOURCING
Cognizant
empirix — Be Confident.
hp — invent
SONATA — SONATA SOFTWARE
twist — collaborative test automation

Media Sponsors:

APLN
acm queue — tomorrow's computing today
BETTER SOFTWARE MAGAZINE
CoDe — www.code-magazine.com
Methods & Tools
StickyMinds.com

Contributors

Chuck Allison developed software for more than twenty years before becoming a professor of computer science at Utah Valley University. He is a technical editor for *Better Software* magazine and founding and current editor of the online journal *The C++ Source*. He spent most of the 1990s as an active member of the C++ Standards Committee and is author of two C++ books, including *Thinking In C++, Volume 2: Practical Programming*, with Bruce Eckel. His company, Fresh Sources, Inc., gives onsite training in C++, Python, and design patterns. His current top technical interest is the resurgence of functional programming. Whenever he finds a little down time he plays classical guitar or bikes the country roads of central Utah. Contact Chuck at chuck@freshsources.com.

Alan Berg, Bsc. MSc. PGCE, has been a lead developer at the Central Computer Services at the University of Amsterdam for the past eight years. In his spare time, he writes computer articles. He remains agile by playing computer games with his kids who (sadly) consistently beat him. You can email Alan at a.m.berg@chello.nl.

Michael Bolton lives in Toronto and teaches heuristics and exploratory testing in Canada, the United States, and other countries. He is co-author, with James Bach, of *Rapid Software Testing* and a regular contributor to *Better Software* magazine. Contact Michael at mb@developsense.com.

Alistair Cockburn is one of the world authorities on software development, having helped to craft the Manifesto for Agile Software Development and the project management Declaration of Interdependence. He created the Crystal family of ultralight methodologies to capture patterns of successful project teams. Many of his materials are available online at www.alistair.cockburn.us.

Matthew Heusser is on the technical staff and a QA group member at Socialtext. Matt has been developing, testing, and occasionally managing software since 1997. Along the way, he found time to organize the Grand Rapid's Perl Mongers, the Great Lakes Software Excellence Conference, earn his master's in CIS at night, and do a modest amount of writing and speaking. You can read Matt's blog at xndev.blogspot.com www.xndev.blogspot.com or email him at matt.heusser@gmail.com.

Paco Hope is a technical manager with Cigital, Inc. and has extensive experience securing Web applications, operating systems, and embedded devices (lottery systems, cell phones, casino gaming devices). He lectures on security testing and Web application security. Paco brings the techniques of security assessment into the mainstream activities of QA departments and testers.

STAR EAST SPEAKER

BETTER SOFTWARE CONFERENCE & EXPO SPEAKER

Naomi Karten has delivered seminars and presentations to more than 100,000 people internationally to help them improve customer satisfaction, strengthen teamwork, and manage change. Her books and ebooks, including *Managing Expectations* and *Changing How You Communicate During Change*, provide guidelines and advice for carrying out projects, delivering superior service, strengthening teamwork, and managing change. Her Web-based. newsletter, "Perceptions & Realities," offers serious advice in a lively, chuckle-generating manner. She is a regular contributor to StickyMinds.com and *Better Software* magazine. Contact Naomi at naomi@nkarten.com or via her Web site, www.nkarten.com.

Jonathan Kohl is a software testing consultant with Kohl Concepts Inc., based in Calgary, Alberta, Canada. Jonathan writes about and speaks on software testing and is a technical editor for *Better Software* magazine. Contact Jonathan at jonathan@kohl.ca.

STAR EAST SPEAKER

Robert Sabourin, P.Eng., has more than twenty-five years' experience leading software development teams. He frequently speaks at conferences and writes on software engineering, testing, management, and internationalization. The author of *I am a Bug!*, the popular software testing children's book, Robert is adjunct professor of software engineering at McGill University and president of AmiBug.Com, Inc.

STAR EAST SPEAKER

EDITOR'S PICK

Joseph McAllister
Managing Editor,
SQE Multimedia
jmcallister@sqe.com

Test Your Pepper

One day, I made myself a salad for lunch. But it wasn't just any salad. It was a special salad.

I'm not anti-vegetables by any means, but I tend to make pretty boring salads at home. They typically comprise spinach leaves, sliced carrots, and a bit of dressing.

But then I was struck, whilst walking the aisles of my local supermarket, with the notion that I like salads and I like peppers, so I'd probably like peppers on a salad. That's called logic, folks. So I tested my logic by slicing one of the long, skinny green peppers and one of the short, wrinkly orange peppers (don't bother me with pepper names!) and putting the various pieces onto my salad. I took my salad back to my workspace and continued editing a *Better Software* article while chomping away at my greens.

Only then did it occur to me that two whole peppers might have been too much for my meager salad. By then, of course, I was quite a few bites in. Don't get me wrong—I love peppers, and I love spicy food. I just like to know it's coming.

My colleague Francesca, the editor of StickyMinds.com, has written about her own experiences with peppers in the StickyLetter Our Take "Pickled Pepper Predicament," and I'm sure I don't need to go into great detail about the wonderful powers of capsaicin. I will merely say that I learned a very important lesson that day about the production of salads with fresh peppers.

Test Your Pepper. Do not release your ~~application~~ salad for public consumption without first testing its ~~various elements~~ peppers. You may find that it has lots of extra ~~code~~ peppers or vicious ~~bugs~~ pepper seeds—and it may be too late. And no matter what, don't pretend afterward that a ~~bug~~ pepper is a feature of the ~~software~~ salad. No one will sympathize and your ~~customer's~~ mouth will still be ~~irate~~ on fire.

To read Francesca Matteu's "Pickled Pepper Predicament," visit
www.StickyMinds.com/editorspick10-3

Quotables

"One should insist on having committed and competent customers onboard from the start of the project. Sadly, this isn't as straightforward as making sure the fish get off the boat and into the factory as quickly as possible."
Clarke Ching, "Don't Let the Engine Run Out of Fuel"
www.stickyminds.com/quotables10-3a

"When my phone recently started spontaneously turning itself off and I repeatedly turned it on without thinking once about not pushing past the defects and taking it to get repaired, it hit me that I had been trained by the flaws of the tools to accept the defects."
Holly Bourquin,
"Desensitizing Ourselves for Disaster"
www.stickyminds.com/quotables10-3b

"It's true that we have tremendous processing power, but we seem to be using up all of it all the time. These days, we are able to do a pretty good analysis and test generation for a class in about one minute on average ... ten years ago it would've taken a full day."
Alberto Savoia, Gray Matters interview
www.stickyminds.com/quotables10-3c

"Systems and software aren't just about correctness; they're also about solving problems for people. One principle of the context-driven software testing movement says that if the problem isn't solved, the product doesn't work."

PAGE 19

Picks

Bumper Stickers for Testers

BY HARRY ROBINSON

Why is software testing perceived as dull? How many other jobs can list "crash," "hang," and "death march" in their daily vocabularies? In this column, Harry Robinson encourages testers to embrace a little pride and excitement in what they do, and Harry has just the mottos for bumper stickers that announce Tester Pride. Author's note: Feel free to add your own favorite slogan in the comment section at the end!

www.StickyMinds.com/eLetterpick10-3a

How Do You Think?

BY FIONA CHARLES

What are the attributes of a good tester—of a great tester? As every test manager knows, identifying the right people for a test team can be a struggle. In this column, Fiona Charles describes the qualities of mind she looks for in testers, and the interview questions she asks candidates so she can evaluate how they think.

www.StickyMinds.com/eLetterpick10-3b

When Requirements Collide

BY KARL WIEGERS

Could it be that not every set of business requirements has the customer's best interest in mind? Karl Wiegers had always believed that implemented software functionality should enable users to accomplish their goals and help the business achieve its objectives. But a recent experience with a less-than-helpful parking meter system suggested to him that conflicts sometimes might exist between business and user requirements.

www.StickyMinds.com/eLetterpick10-3c

POWERPASS POINTER

Your Job - Requirements = Less Value

BY DION JOHNSON

An effective Project Customer can turn a good product into a great one. But he's got to know how. Mike Cohn gives Project Customers seven simple rules for communicating product goals effectively to development.

www.StickyMinds.com/eLetterpick10-3d

eNEWSLETTER EXTRA

A sampling of content from our eNewsletter archives

Sticky ToolLook: *November 9, 2006*

A Word with the Wise: Security Tools with Gary McGraw

by Joseph McAllister

This month's installment of A Word with the Wise is a Q&A with software security expert Gary McGraw. He's the CTO of Cigital and the author of numerous books, including *Software Security: Building Security In* and the upcoming *Exploiting Online Games*.

Joseph McAllister: *How are security issues changing the way development organizations look at software quality and the development and testing tools that are needed?*

Gary McGraw: It's not clear that developers adopting software security tools are understanding that those are actually kind of QA tools. In fact, I'm a little concerned about the first generation of software testing tools for security that are coming down the pike. These tools supposedly capture a whole bunch of black box tests, and you can run them against any arbitrary program.

If you run all those tests and you don't find any problems, you don't know anything—you just ran some tests. On the other hand, if you run them and you do find some problems, you know that you're in deep trouble, because a stupid test knocked your program over.

I liken those to badness-ometers that range from "deep trouble" to "I'm not really sure." I think everybody should use them, because it's good to know if you're in deep trouble—maybe you'll do something about it. On the other hand, they're not security-meters, so if you run one of these test things against your program and it doesn't find any problems, you can't declare victory. You have to declare ignorance.

JM: *Do you think that the tools will become more effective?*

GM: There is no completely off-the-shelf, hacker-in-a-box solution that you just plug in, press return, and off you go. What you need to do is arm a real tester with the kinds of technology that you need to build seriously good security tests. But you'll never get the human out of the loop.

JM: *And are there any other particular security issues on which tools will have little impact?*

GM: There are these application firewall tools that are pretty silly, too, and they work by watching network traffic and trying to say, "That's bad input, that should never go to the program." If you're going to try to do some filtering, you should do it inside your program, not at the firewall or on the network wire.

JM: *Who knows better how your program should handle input than you, the guy who wrote your program? Not the security guy, who doesn't even know what a compiler is.*

GM: On the other hand, there are some new tools that have only been out for a year or two that take a very deep look at source code for software and help to identify vulnerabilities at the source-code level.

They are based on compiler technology. That kind of technology is very useful for those testers and QA people who practice white box testing and who do things like code review for a living. I think [those tools are] still in fairly low use out there, but I also see people beginning to adopt them quickly. If you're a QA person and you're responsible for code review, you should take a look at those static analysis tools for security right away.

JM: *And those will become more effective as more people hear about them?*

GM: They're already pretty effective. They'll have more impact as more people wield them. The funny thing about security is, there are about 500 or 600 things you could do wrong in C and C++ based on APIs, system calls, library calls, and just language idiocies. I don't know about you, but I can't remember all 500 of those while I'm staring at code trying to figure out what the heck it does. Those tools can help you remember those things while you're doing your review. If they identify areas in the code that you should worry about, you can probe those dynamically with reasonable testing.

Book Review

Design Patterns: Elements of Reusable Object-Oriented Software

by Gamma, Helm, Johnson, Vlissides
Reviewed by Dimitri Ilkaev
dilkaev@tier.com

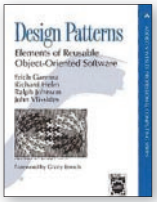
Design Patterns is a classic that describes simple and effective solutions to common problems in object-oriented software design. The book, written in a catalog format, provides examples of how design patterns can improve the software design, together with reusability and flexibility of code. The authors describe the intent of the pattern, motivation, applicability, structure, participants, collaborations, consequences, and implementation considerations. They provide detailed sample code, known users of that pattern, and a list of related patterns. Authors Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides—frequently referred to as "The Gang of Four"—separate these patterns into creational, structural, and behavioral categories.

The first two chapters of this book introduce the pattern-based design world, while the rest present a catalog of design patterns. *Design Patterns* is most helpful for the skilled software developer or architect, especially on relatively large projects. Multiple diagrams—most UML based—make the design concept for a specific, easy-to-understand pattern, and every pattern description is accompanied by code samples in C++ or Smalltalk.

The authors use a common vocabulary, which if implemented, encourages effective communication with peers in the design and development field. This book creates a pattern-based mindset, which, once adopted, stays with a developer forever. The authors set the groundwork for pattern-based design, making this book a "must have" book on a developer's desk.

I recommend this book to experienced developers or architects who need to look up specific cases and patterns related to their design challenges, yet the less-experienced developer should also find this book helpful because it sets a pattern-based direction and can be used as a guidebook.

SPOTLIGHT BOOK



Design Patterns
Elements of Reusable Object-Oriented Software

By: Gamma, Helm, Johnson, Vlissides

Pages: 395 **Published:** 1995

Publisher: Addison Wesley **ISBN:** 0201633612

Description:

Capturing a wealth of experience about the design of object-oriented software, four top-notch designers present a catalog of simple and succinct solutions to commonly occurring design problems. Previously undocumented, these twenty-three patterns allow designers to create more flexible, elegant, and ultimately reusable designs without having to rediscover the design solutions themselves.

The authors begin by describing what patterns are and how they can help you design object-oriented software. The authors then go on to systematically name, explain, evaluate, and catalog recurring designs in object-oriented systems. With *Design Patterns* as your guide, you will learn how these important patterns fit into the software development process, and how you can leverage them to solve your own design problems most efficiently.

Each pattern describes the circumstances in which it is applicable, when it can be applied in view of other design constraints, and the consequences and trade-offs of using the pattern within a larger design. All patterns are compiled from real systems and are based on real-world examples. Each pattern also includes code that demonstrates how it may be implemented in object-oriented programming languages like C++ or Smalltalk.

Read this book? Visit www.StickyMinds.com/bythebook10-3 to post your comments, or email editors@bettersoftware.com for information on how you can have a book review considered for publication on StickyMinds.com.

Top 5 Most Read on StickyMinds.com

Don't be the last one in the know. Read five of the most popular articles on StickyMinds.com from recent weeks.

1 Software Testing: A Profession of Paradoxes?
by Zeger Van Hese
www.StickyMinds.com/10-3Top1

2 Security Testing: What Fresh Hell Is This?
by Linda Hayes
www.StickyMinds.com/10-3Top2

3 What Project Managers Need to Know About Testing
by Johanna Rothman
www.StickyMinds.com/10-3Top3

4 How to Write Better Test Cases
by Dianne Runnels
www.StickyMinds.com/10-3Top4

5 State-Transition Diagrams
by Lee Copeland
www.StickyMinds.com/10-3Top5

HELP WANTED

Become an active part of the StickyMinds.com/*Better Software* magazine community. The following are real questions posted to the StickyMinds.com Discussion Boards where you can ask questions, give pointers, and exchange messages with your sticky-minded peers.

Perhaps you have your own burning question or a problem that could benefit from the input of your peers. Email us at editors@bettersoftware.com for consideration. Your question could be featured in a future issue of *Better Software* magazine.

I am being asked to propose and implement a testing framework using industry standards. Has anyone got any suggestions as to what might be the best standards in a very immature organization?
Posted by: markhen

If I have several phases of testing like system testing, system integration testing, user acceptance testing, and operational acceptance testing, how do I calculate the DDP (defect density percentage) for each phase and then for the overall testing phase?
Posted by: testkid

I am a little bit familiar with cross scripting. Does anyone know the site to learn how to perform the cross scripting? How does cross scripting affect security?
Posted by: sachinywaiker

You are a project manager for a testing team. You are assigned a project to manage for testing that is newly developed from the scratch (New Development). You are provided the scope of the project and you are asked to provide estimation of testing effort.

Question : How would you estimate the testing effort?
Posted by: rsubbu78

If you have insight or advice to share, log on to the StickyMinds.com Discussion Boards or email us at editors@bettersoftware.com. Select responses will be published in an upcoming issue of *Better Software* magazine.

eLightenment

CONTENT FROM StickyMinds.com and Beyond

PODCASTS

Have You Heard? We've Got ... StickyMinds.com

The StickyMinds.com Gray Matters podcast series has expert interviews to feed the stuff between your ears. Gray Matters features audio interviews with software experts and clips from Software Quality Engineering conference sessions. Our most recent podcasts feature conference clips and conversations with such notables as Lee Copeland, Esther Derby, and Alberto Savoia. Take a listen at www.stickyminds.com/Podcasts.asp

StickyMinds sound byte

The StickyMinds SoundByte is a podcast that keeps you updated on what's new at StickyMinds.com and in *Better Software* magazine. The April episodes of the StickyMinds SoundByte will feature updates from and about April StickyMinds.com columnists Naomi Karten, Johanna Rothman, Danny Faught, and Dion Johnson.

If you have a question or a comment for one of our April authors, call our Podcast number, 1.888.268.8770, and dial option 8 to leave a message. Your question or comment might be featured in a future podcast!

READ ALL ABOUT IT!

StickyMinds.com eNewsletters Have Something for Everyone

Have you checked out our latest eNewsletter, iterations? It's the eNewsletter that's about all things agile. If you've been missing out, visit the eNewsletter archive on StickyMinds.com to read every issue. In addition to the latest issue of the What's New Gram, you'll find every issue of these eNewsletters:

From the Editors of StickyMinds.com StickyLetter — The StickyLetter—Features original StickyMinds.com content hand-picked by the editors of StickyMinds.com

Between The Lines — Between the Lines—Offers industry views on recent news

FRESH INK — Fresh Ink—Provides an early glance at the next issue of *Better Software* magazine—features, columns, and more

iterations ALL THINGS AGILE — iterations—Delivers all things agile straight to your inbox in a monthly eNewsletter

Did you know ... As a StickyMinds.com member, you can use StickyMinds.com as a "distribution shelf" for your written work? StickyMinds.com provides broad visibility by adding your work to the searchable repository of information on which our users rely. Your article or technical paper will remain in your original words, format, etc. Check out www.StickyMinds.com/submitstuff.asp today.

Spring Is In The Air!

Just as April showers bring May flowers, Monday mornings bring a brand-new weekly column to the StickyMinds.com homepage. Knowledge is in season this month with columns from industry experts including Naomi Karten, Johanna Rothman, Danny Faught, and Dion Johnson.

A Change Would Do You Good

by Jonathan Kohl

I just visited a local bookstore where the shelves are lined with books like *Spice Up Your Diet, Exercise That Can Change Your Life*, and *Spice Up Your Love Life*. It stands to reason that if I get bored with these basic needs, my more complex needs, such as solving technical problems, are also going to be affected over time. If a change in my diet and exercise regimen can be beneficial, wouldn't a change to my favorite software processes do me good as well?

A senior manager and friend approached me with a set of communication problems his software development team was experiencing. He described situations where important information was missed, lost, or not made available to the right people, which led to duplication of effort, software errors, and employee frustration. To help deal with these issues, I suggested that he implement daily Scrum standup meetings. This is a meeting style where each team member talks about three things: what he has been working on since the last meeting, what he plans to work on, and any roadblocks he has to completing his work.

I checked back after a few months, and my friend was excited with the results. He said that the team was working together in ways he hadn't expected. There was less duplication of work and the shy members were much more assertive. We spoke again, months later. "You're not going to be very happy with me," he commented. He told me his team had stopped doing daily standups because after a while, they weren't productive. They had become a hollow routine, and while people were speaking to each other, they weren't really communicating anymore.

As a response, he had changed the standup schedule. During slower times in a project, they might have one or two standups a week. When they were busier, they would increase the frequency. Closer to a software release, when team members needed more communication, they went back to a daily schedule. He explained that this change had returned them to their former level of success. Instead of being upset with him, I was overjoyed. When he recognized the tool was no longer contributing to the success of the team, he changed it.

When I was in university, one lesson I learned in a business class was that new teams have trouble being productive at first because they lack *cohesion* (uniting for a common purpose). With time, organizational support, and a healthy environment, they become cohesive and are able to be productive. However, after a while, teams can become *too* cohesive and productivity suffers. In some cases, team members may become bored and spend time engaging in distracting behavior instead of working. As managers, our job is to monitor the level of group cohesion over time and help the team introduce just enough positive change to keep it engaged in a productivity sweet spot. The lesson is: If you don't adapt your processes and practices over time, you will see early success, reach a plateau, and then regress.

> "As managers, our job is to monitor the level of group cohesion over time and help the team introduce just enough positive change to keep it engaged in a productivity sweet spot."

We see this in other disciplines as well. On sports teams, like hockey teams, coaches understand the need for continuous change. Hockey coaches "juggle lines," which means they change who is playing with whom as they try to win games. They don't change for change's sake, but they understand that mixing things up helps productivity. Athletes also change their training regimens, because they find that they plateau and stagnate if they do the same kind of training year after year. Moving back to software, in *Crystal Clear Applied: The Seven Properties of Running an Agile Project*, Alistair Cockburn points out: "The people, the technology, and the assignment change over the course of a project. The conventions the team uses need to change to match." Alistair has even recommended software teams try a different software development methodology on each project.

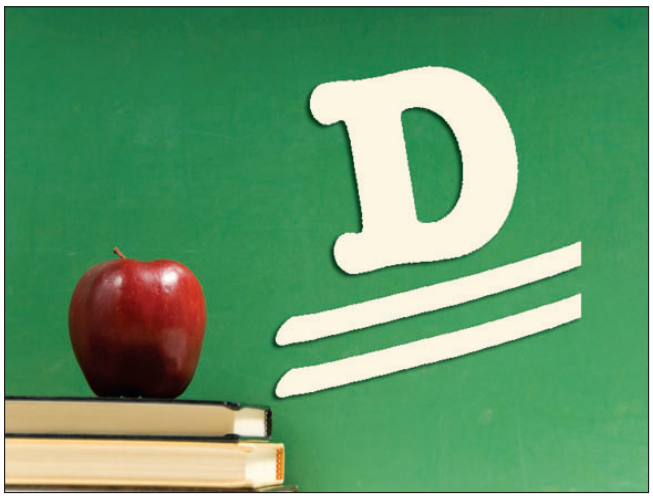
Methodology change is challenging. Sometimes, even "change-embracing" agilists can be surprisingly rigid when it comes to changing processes that aren't working anymore. We often put process on a pedestal and believe that merely following it will take care of everything else. We need to put process in its place. A software development process is merely a tool that we can use to help reach our goals.

One important goal is to provide value to our customers. Merely following a process and completing tasks is not always enough to satisfy and impress them. Adherence to a process shouldn't be the end goal; we should be more concerned with how our process helps us create value. The more processes we are fluent in, the more tools we have to help create value.

One interpretation of the Sheryl Crow song "A Change Would Do You Good" is that it describes a break up with a lover. Is it also time to "break up" with some of your old, ineffective software development practices and try something new? **{end}**

A "D" in Programming, Part 1

by Chuck Allison



ISTOCKPHOTO

They say there are three things you should never discuss if you want to get along with other people: politics, sports, and religion. Since I want to stay on your good side, I'll steer clear of these topics, but I would like to discuss something else that often elicits religious-like fervor from developers—favorite programming languages.

Language Wars

What's the best programming language? Be honest—how many times have you defended your favorite language du jour? And how many languages have been your "favorite" at one time or another?

To seed the discussion, let me relate my first and only exposure to COBOL, one of my "unfavorites." In 1975, I was standing in line, card deck in hand, waiting to submit my job to the IBM 360 expediters at Brigham Young University. I had about forty cards containing a Fortran program that numerically solved a system of ordinary differential equations. Complex stuff. Behind me stood another student with a deck of cards more than a foot thick—800 to 1,000 cards, I'd guess. Impressed and intimidated, I asked the student what his program did. "It writes a report." Hmmm. Since I had written reports in Fortran in considerably fewer statements, I decided right then and there I would never use COBOL. No offense to the many COBOL aficionados out there—I just knew it wouldn't work for me. My mathematical background has ingrained within me a constant yearning for economy of expression.

One language is not necessarily better than another, of course. For example, I prefer Python to Perl, but both are fine scripting languages as well as general-purpose, multi-paradigm programming languages. I see Python as cleaner and more orthogonal in its design—your mileage may vary. Nonetheless, you may appreciate the following "Jedi wisdom":

> **YODA:** Code! Yes. A programmer's strength flows from code maintainability. But beware of Perl. Terse syntax … more than one way to do it … default variables. The dark side of code maintainability are they. Easily they flow, quick to join you when code you write. If once you start down the dark path, forever will it dominate your destiny, consume you it will.
>
> **LUKE:** Is Perl better than Python?
>
> **YODA:** No … no … no. Quicker, easier, more seductive.
>
> **LUKE:** But how will I know why Python is better than Perl?
>
> **YODA:** You will know. When your code you try to read six months from now. [1]

Not too many years have passed since the Java "revolution," fueled not only by Sun's leveraging the sudden popularity of the Internet but also in part as a "reaction" to C++. I well remember public discussions at conferences and online debunking C++ as unsafe ("pointers are evil and Java doesn't have pointers"—a deception) and too difficult to learn (Java is no cakewalk either). These comments were relentlessly proffered as context for evangelizing the so-called superiority of this new programming language. I respect Java and its inventors, have used the language professionally and taught it in academia, but I still wince when I recall the bad form that characterized Java's rise to popularity. There is a certain irony in the fact that the HotSpot JVM is written in C++.

That said, there is always room for improvement. Java does have a simpler object model and a more flexible execution model than C++. It has brought portable concurrency and GUI tools to Joe Programmer. The relatively new Scala language is in turn an improvement on Java, combining high-level, functional programming constructs with static type safety. Scala translates to Java byte code providing access to the massive Java library. The Groovy language is a similar alternative but with dynamic typing.

Alas, it's been twelve years since Java made its debut, and its popularity seems to have peaked as it is finding its niche mainly in enterprise applications. Just today I found the following in an *InfoWorld* article entitled, "Java is Becoming the New COBOL":

> "Java, the oldest new programming language around, is falling out of favor with developers. When it comes to developing the increasingly common rich Internet applications, Java is losing ground to Ruby on Rails, PHP, AJAX and other cool new languages … Now that Java is no longer the unchallenged champ for Internet-delivered apps, it makes sense for companies to find programmers who are skilled in the new languages. If you're a Java developer, now's the time to invest in new skills." [2]

New Favorite

In recent Code Craft articles, I praised the relatively new D programming language. D combines the speed and power of C++ with desirable features of many other programming languages, such as Java, C#, Eiffel, and functional programming languages. D is high-level, readable yet concise, multi-paradigm, and efficient. In this article and the next I'll conclude (for now) my D evangelizing by exploring a few of its more noteworthy features.

Foremost for me is the fact that D is strongly typed, compiles to native code, and yet is garbage collected. Currently this is a singular achievement in the programming world. But there is much more about D to recommend.

Code Consistency with Scope Guards

Unless carefully planned for, runtime errors can easily place a program in an inconsistent state. Resource management is a typical example. Suppose you have a situation similar to the function in listing 1.

A Java-like solution uses a `finally` block similar to the D program in listing 2.

A C++-style approach encapsulates resource management into an object's constructor and destructor. This is the well-known Resource Acquisition Is Initialization (RAII) idiom, shown in listing 3.

This approach makes `f`'s code concise, but it also requires an auxiliary class.

D supports RAII, but it also lets you explicitly supply specific cleanup code for when execution exits a scope, as shown in listing 4.

The `scope` statement, called a *scope guard*, activates a code block that may or may not run when a scope is exited. Figure 1 lists the three scope-guard options.

`scope(exit)`	the code always runs (like `finally`)
`scope(failure)`	the code runs only if an exception occurs
`scope(success)`	the code runs only if no exception occurs

Figure 1

The utility of the `scope` statement becomes more obvious with multi-step resource acquisition requiring rollback semantics in the case of failure. Consider the three-part transaction in listing 5.

In this case, we want all three operations to succeed or fail together. RAII alone will not solve this, and the `try-finally` approach is not pretty—witness listing 6.

Things only get worse with more pieces in the transaction. Not so with D. Listing 7 shows how easily you can back out of complicated transactions with D's scope statement.

When execution leaves a scope, all scope-guard blocks that have executed are visited in last-in-first-out order, so transactions roll back gracefully, and thirteen lines of code become six. If `risky_op2()` is the first to fail, only `undo_risky_op1()` executes as the execution propagates out of `g`. Likewise, if only `risky_op3()` fails, then `undo_risky_op2()` executes, followed by `undo_risky_op1()`.

```
void f() {
    acquire();
    risky_op(); // Might fail
    release();
    writeln("f succeeded");
}
```

Listing 1

```
void f() {
    acquire();
    try {
        risky_op();
        writeln("f succeeded");
    }
    finally {
       release();
    }
}
```

Listing 2

```
class Resource {
public:
    Resource() {
       cout << "resource acquired\n";
    }

    ~Resource() {
       cout << "resource released\n";
    }
};

void f() {
    Resource r;
    risky_op();
    cout << "f succeeded\n";
}
```

Listing 3

```
void f() {
    acquire();
    scope(exit) release();
    risky_op();
    writeln("f succeeded");
}
```

Listing 4

```
void g() {
   risky_op1();
   risky_op2();
   risky_op3();
   writeln("g succeeded");
}
```

Listing 5

In Part 2, I'll show how programming with nested functions and delegates can be more concise and powerful than classes and objects. **{end}**

REFERENCES:

1] www.netfunny.com/rhf/jokes/99/Nov/perl.html

2] www.infoworld.com/article/07/12/28/52FE-underreported-java_1.html


```
void g() {
    risky_op1();
    try {
        risky_op2();
    }
    catch (Exception x) {
        undo_risky_op1();      // Back-out op1
        throw x;               // Rethrow exception
    }
    try {
        risky_op3();
        writeln("g succeeded");
    }
    catch (Exception x) {
        // Back-out op1 and op2 in reverse order
        undo_risky_op2();
        undo_risky_op1();
        throw x;
    }
}
```

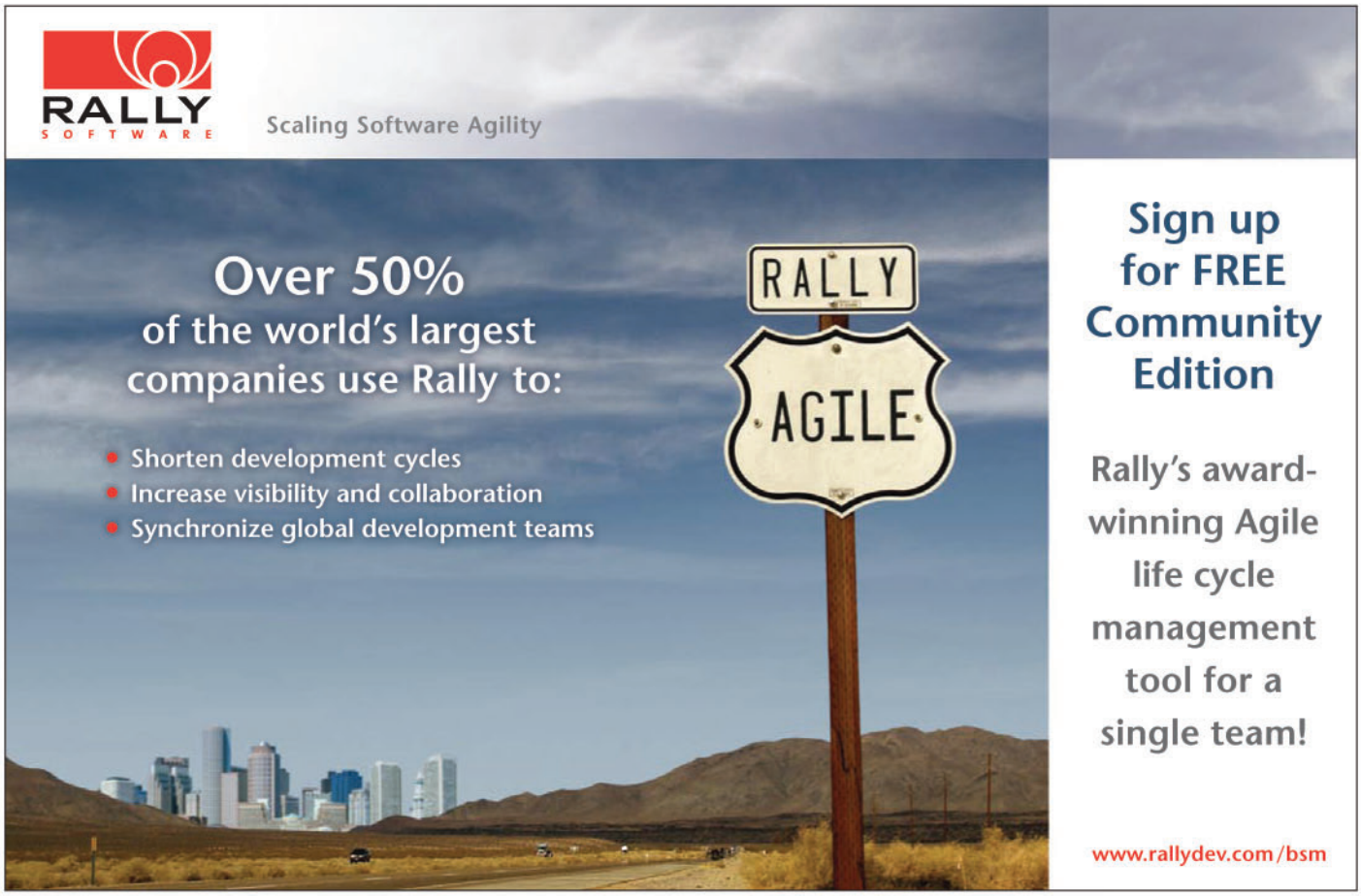
Listing 6

```
void g() {
    risky_op1();
    scope(failure) undo_risky_op1();
    risky_op2();
    scope(failure) undo_risky_op2();
    risky_op3();
    writeln("g succeeded");
}
```

Listing 7

What have been your favorite languages over the years? What is your opinion of D's scope statement?

Follow the link on the StickyMinds.com homepage to join the conversation.



BETTER SOFTWARE CONFERENCE & EXPO

JUNE 9–12, 2008
LAS VEGAS, NEVADA
THE VENETIAN

SOFTWARE QUALITY ENGINEERING CONFERENCES
www.sqe.com

KEYNOTES BY INTERNATIONAL EXPERTS

Ken Schwaber
Advanced Development Methods, Inc.

Michael Mah
QSM Associates

Jean Tabaka
Rally Software Development

Johanna Rothman
Rothman Consulting Group, Inc.

TUTORIALS WORKSHOPS CLASSES

CELEBRATING 5 FIVE YEARS

Alan Shalloway
Andy Glover
Andy Hunt
Andy Kaufman
Ari Takanen
Beth Layman
Bob Hartman
Chris Ronak
Chuck Allison
Dan North
David Garmus
David Herron
David Spann
Ed Weller
Elle Ringham
Guruprasad Gopalakrishnan
Hubert Smits
James McCaffrey
James Newkirk
Jared Richardson
Jean Tabaka
Jeff Patton
Jeff Payne
Jerry Smith
Jimmy Xu
Johanna Rothman
John Janakiraman
Julie Gardiner
Ken Pugh
Ken Schwaber
Kent McDonald
Kevin Bodie
Lee Copeland
Lee Devin
Linda Rising
Linda Westfall
Lisa Crispin
Michael Mah
Michael Sutton
Michele Sliger
Mike Seavers
Mike Tholfsen
Mitch Lacey
Nelson Perez
Paco Hope
Payson Hall
Pollyanna Pixton
Rebecca Wirfs-Brock
Richard Bender
Rob Myers
Robert Galen
Stacia Broderick
Tim Korson
Todd Little
Vladimir Pavlov
Will McKnight

www.sqe.com/bsce

Industry Sponsors:

empirix Be Confident. · NetObjectives · McCabe SOFTWARE · Telelogic

Media Sponsors:

acm queue · APLN · IEEE COMPUTER SOCIETY · BETTER SOFTWARE MAGAZINE · CoDe · Methods & Tools · StickyMinds.com

Learning the Hardware Lessons

by Michael Bolton

ISTOCKPHOTO

The other night, I went to a local hardware store to replace a broken piece of a shelving unit in my daughter's closet. The store was closed, so I went to the competitor across the street—a chain store that I usually avoid because of patterns of familiar problems. I found something that looked like the part that I needed and took it to the cashier. I spent the usual several minutes in line, watching the cashiers address one problem after another related to the point-of-sale system that the store uses. Some items were missing the barcode stickers that allow items to be scanned and recognized by the software; the prices of some sale items were inconsistent with the prices displayed on the shelves; and some items apparently weren't in the system at all, even though they had seemingly legitimate stickers. Each customer transaction took several unnecessary minutes to resolve.

The fellow ahead of me was sipping a cup of coffee. I should have gotten one for myself, I thought—we're likely to be here for a while. As usual. The coffee drinker looked around. "Bloody computers," he said to me. "They're always broken at this store."

"That's my experience, too," I replied. "Although, technically, the *computers* are probably working just fine; it's the *programs* that are broken." We had time for a chat, and eventually he learned that I was a software tester.

"Ha! I guess the program for *this* system wasn't tested very well," my new friend said.

"Well, without specific information about the project, I can't be sure of that," I replied. "When we testers find a problem, it's up to programmers and managers to decide what to do about it. They might decide to fix it, or they might decide that the problem isn't serious enough to bother with. They often decide not to fix problems because they perceive that it might be expensive or risky, which can be a reasonable decision in a lot of cases. On the other hand, you only really get to find out about the quality of your risk assessment by paying attention to what happens in the field."

He finished his cup of coffee. "It doesn't look like they're paying much attention here," he said.

"It doesn't, does it? It's too bad, because confusion, delays, or annoyances—for the sales clerks or the customers—are real problems. They affect employee morale, the length of time that it takes to pump a sale through the system, and things that the customers value—like their time. Ultimately, *that* affects the bottom line. The store needs more cashiers to handle the same number of customers, or employees get frustrated and quit."

"Or customers head for the competition."

"Right. Managers here don't seem to observe the problems that the cashiers are having, and they don't seem to take notice of the amount of time that customers spend in line. I've been avoiding this chain for years because I can depend on having to wait more time than I think is reasonable. I only come here when I'm desperate."

"Me too," he said. "Don't managers realize how much that costs them?"

"Well, people say that missed opportunities are hard to measure. Plus, it's hard to evaluate things when you don't see how they interact with the rest of the system. Most testers sit in front of computer screens, testing the software but not the process that it's designed to support. If they tested the whole system, good testers would discover more important weaknesses, and they'd be able to tell better stories about how the problems threaten value. Smart developers and smart managers would notice possibilities for increased value if the system worked better. I can't say much about the testing, but I can pretty much guarantee that the system hasn't been developed, or managed, very well."

It was finally my new friend's turn to go through the checkout. He did so without incident. He looked over his shoulder and grinned as he left, "Good luck."

My turn. I realized to my horror that my shelf bracket didn't have a sticker on it. "It costs a dollar ninety-seven," I said weakly. The cashier asked me if I could wait for someone to do a price check. I had noticed that staff members were constantly being paged for price checks, and that people were waiting a long time for the information to come back. I decided to go and get the information myself.

There was a label on the shelf with a description, a bar code, and two numbers: 71924-20 and A434-300. There was no picture to match the product with the code, but most of the brackets in the same bin looked the same as the one I had. The description of the product was easy to remember. I fumbled for my notebook, but I had left it in my other jacket. I had a receipt in my pocket, but no pen and no staff around to lend me one. I memorized the numbers using some mnemonic tricks (7pm is 1900 hours, of which there are 24 in a day, 20 is easy. A's the first letter of the alphabet, 434's a palindrome, and 300 is a lousy movie.) I took the long walk back to the cashier.

"71924-20," I said. She typed it in.

"Wait. That's too many numbers," she said, clearly frustrated.

I said, "The only things that work in this place are the people. I really do appreciate that you're struggling here." She smiled. "What about A434-300?"

She tried it. "Nope. Six numbers is right, but it can't start with a letter." I looked all over the item. There was a number on it—but only four digits.

"What about dropping the A?" I suggested. "434-300?"

She tried backspacing, but that displayed only equal signs in the text field. Shift-backspace allowed her to backspace over the whole field, and she typed in the number. "Ah, that works. Shelf bracket. A dollar ninety-seven."

That was a lot of work for a two-dollar part, and the system didn't do very much to help. Why are there two numbers on the bin label? Why is neither number identified? Why does the software accept neither and reject both? Why isn't the acceptable number printed on the label? When I mistype something in the Google search window that doesn't match an entry in its database, Google offers a plausible suggestion—a close-but-not-exact match. Couldn't the store's system have tried a database search based on the numeric portion of the number? Couldn't it have cross-referenced one catalog with another to offer at least a choice or a guess as to what the product might be? Meanwhile, the clerk had an idea about the correct number format, but not how to deal with an incorrect number. Had management realized this as a training issue?

I'll guess that when the developers for this product designed, built, and tested it, they thought in terms of confirming that it worked. I'll guess that when they tested it, they used all kinds of boundary testing and field validation checks to make sure that the software accepted properly formatted numbers and rejected improperly formatted numbers. Those things are important. But I'll also guess that they tested the software in isolation from the environment in which it was intended to operate, and that they looked for functional correctness without testing the product in the field and asking, "Is there a problem here?" And finally, I'll guess that, to this day, they don't know about the frustrations that customers and cashiers alike are having with the system.

Systems and software aren't just about correctness; they're also about solving problems for people. One principle of the context-driven software testing movement says that if the problem isn't solved, the product doesn't work. Testers: Could we find more bugs—and *more important* bugs—in our systems by observing something other than the software itself? And managers: What problems could we prevent by letting our testers see more than some screens and some specifications? **{end}**

How would you answer the questions above? What additional questions should we be asking?

Follow the link on the **StickyMinds.com** homepage to join the conversation.

26TH ANNUAL
PACIFIC NW SOFTWARE QUALITY CONFERENCE
October 13 -15, 2008
Portland, OR

World-class quality does not happen in a vacuum. Agile-inspired collaboration spans levels, disciplines, and industries.

2008 keynote speakers Ron Jeffries and Chet Hendrickson will address *Quality Dynamics in Agile Software Development*.

www.pnsqc.org

The Art of Persuading Management

by Naomi Karten

Stew was venting. "I-give-up-I-give-up-I-give-up."

"OK, take a deep breath and slow down. Now, what's the problem?" Bruce loved being a mentor. Stew had been a great developer, but as a new project manager, he was still a little rough around the interpersonal edges.

"It's Jake," Stew said. "You'd think a manager would want his teams to succeed. But he just won't listen."

"Listen to whom about what?" Bruce asked.

"To me, of course, about our next project. I want my team to be able to work entirely from home. It's a great idea, but Jake didn't want to hear it."

"Are you saying he wouldn't listen to you at all, or that you made a case and he rejected it?"

"Oh, I made a case, all right, and yeah, he rejected it. Outright. Just like that." Stew slapped his hands together, squash-a-mosquito style.

"OK," Bruce said, "what did you say to try to persuade Jake?"

"I told him how much more productive team members would be if they worked at home. No commuting, no time wasted in meetings, no interruptions. Ergo, happier employees. Everybody's doing it."

"Whoa!" Bruce signaled Stew to stop. "Who's everybody?" Stew was ready with an answer: two telecommuting friends of his who worked almost entirely at home.

"How has it worked out, not only for them but also for their team and their projects? And how has it worked out for their company?"

Stew started to answer, then stopped short. "Uh, well, I know they like not spending half their lives in rush hour traffic. Other than that, I guess I don't know."

Bruce raised his eyebrows. "A good starting point in being persuasive is to know what you're talking about. What is the impact of your friends' telecommuting on project timeframes? Costs? Security risks? How do team members stay in contact and in synch? And what's involved in transitioning to home-based development? It's great that your friends are happier, but Jake needs to know the impact on the project and the business."

ISTOCKPHOTO

"So, I need to check out the facts?" Stew asked sheepishly.

"Sure do," Bruce said. "Claiming that 'everybody's doing it' is a mistake. Obviously, not everyone is doing it, *whatever* it is, so speaking in absolutes—always, never, everybody, nobody—is automatic points off.

"Instead," Bruce continued, "find compelling examples to support your case. In addition, list the potential pitfalls and how to avoid them. The fact that you've considered both sides of the issue will show Jake that you've given the matter serious thought." Bruce had a way of hitting nails on the head.

"Another thing: Make sure you consider Jake's perspective. How will he benefit from your idea? How will it help him be seen as successful by his superiors? And what does he emphasize when *he* seeks to persuade? For example, if he stresses facts and figures, do the same. If he focuses on the impact on people, try to orient your points similarly. The better you relate your case to Jake's priorities and preferences, the better your chances of winning him over.

"Now tell me," Bruce said, "when did you try to present your case to Jake?"

"Right after he returned from the off-site," Stew said, certain that this, at least, he had done right. "I wanted to be the first one to get to him his first morning back."

Seeing the "oh no, you didn't" look on Bruce's face, Stew realized he was about to learn something that should have been obvious. "When you're trying to be persuasive, timing matters. Piles of problems probably awaited Jake's return. If you want his undivided attention, don't drop in unannounced. Schedule a meeting at *his* convenience.

"One other thing," Bruce added. "If you're going to try to make a persuasive case in person, practice your delivery. Prepare as if you're about to give a formal presentation—which, actually, you are."

Stew felt emboldened by Bruce's advice. "So you're saying that if I do these things, Jake will give me the thumbs up?"

Bruce shook his head. "Not at all. By

STORY LINES

- **To be persuasive, take a methodical approach. Don't be haphazard.**
- **Gather data so you can be specific and concrete.**
- **Consider the perspective and priorities of those you want to persuade.**
- **Take timing into account. People preoccupied with other priorities won't be predisposed to hearing your case.**
- **Before making a case to management, practice—so you come across as poised, polished, and professional.**

doing these things, you'll stand a better chance of having your ideas heard and considered. But no matter what you do, you won't win them all—Jake has issues of his own that will influence his decision. And for a major proposal like this one, it may take time for him to come around. Maybe it'll be for the next project instead of this one—or the one after that. Getting buy-in for a major change takes patience and quiet persistence.

"The most important thing to remember: If you want to be persuasive, avoid doing things that will make you *un*persuasive."

"Ah," Stew said, "you've persuaded me." **{end}**

Which of Stew's mistakes offered lessons from which you can learn? What steps will you take next time you want to persuade management?

Follow the link on the **StickyMinds.com** homepage to join the conversation.

CALLING ALL SOFTWARE MANAGERS

Every manager has a story to tell, and we'd like to hear yours!

The editors of *Better Software* magazine are looking for a software manager with a literary flair, to write an installment of our monthly Management Chronicles department.

For more information, email a query to **editors@bettersoftware.com**. Subject: Management Chronicles.

Better Software Has Gone Digital!

Read your favorite magazine straight from your desktop before it hits the street—check out the digital edition of this issue online today at www.StickyMinds.com/April

Want to switch to the digital edition? Contact our subscription department at 1-800-450-7854 or info@BetterSoftware.com to find out how.

IBM, the IBM logo, Rational and Take Back Control are trademarks or registered trademarks of International Business Machines Corporation in the United States and/or other countries. © 2007 IBM Corporation. All rights reserved.

_INFRASTRUCTURE LOG

_DAY 72: We wrote our software but didn't build it to fit with the broader IT architecture requirements. Now we don't have the flexibility to reuse our assets. We're not moving forward. Why did we lock ourselves in like this?

_I never knew being stuck at work could be so literal.

_DAY 73: Here's something less confining. IBM Rational unifies all aspects of our SOA software design and development. Now we can ensure global architecture integrity using a new, simpler, modular systems development approach. And we're speeding our results with sound architectural design and automated service delivery and maintenance.

_I'm glad we're free. Was never sure where to put "stuck to my coffee cup" on my time sheet.

Rational.

IBM.COM/**TAKEBACKCONTROL**/INNOVATE

Incremental and Iterative Developmen

HOW they are different and WHY you should be doing both

by Alistair Cockburn

People still get wrapped around the axle trying to understand the difference between incremental and iterative development. The Unified Process authors in the 1990s didn't help by indiscriminately calling everything iterative development. The two are different and must be managed differently. Successful teams do both at the same time, usually without thinking about it. Then someone starts thinking about it and does one without the other. Bad news follows.

Current agile practice doesn't seem to make it important that the users get to see what's being built and have a chance to change it. Iteration lengths are so short that the programmers barely have time to program up the basics before the end of the iteration, so there's no time for the users to show up and say, "No, that's not actually what I want."

I heard one Extreme Programming (XP) customer ask others in a multi-project exchange:

> Is it just me, or do you also have to *get it right* the first time? On our project, if I request a change, they tell me that story will go to the back of the queue and the project timeline will be delayed. As a result, I have to get it right the first time. Isn't this against the very spirit of agile development?

This person is right on all counts—it happens, it runs against the spirit of agile development, and it is bad business.

This problem is the opposite of what we wrestled with back in the 1990s. Back then, we struggled to get people to do incremental development. Getting people to break work into small-ish pieces and build a piece at a time was all we aspired to. Many people understood that it was good policy to show the results to users several times, and so, oddly enough, we sometimes saw iterative but not incremental development!

If you create smaller pieces but don't show them to real users *and incorporate their responses*, then you revert back to the project failures of the 1980s, when users complained that what got delivered to them didn't satisfy their needs.

How Did We Get into this Mess?

Back in 1991, my assignment with the IBM Consulting Group was to investigate development methods for object-technology projects. In my research, I ran into two terms—*incremental* development and *iterative* development.

I read articles and interviewed people to learn what they meant by these two phrases and learned that they refer to distinct *staging* and *scheduling* strategies. They are choices made by the project team on how to slice, integrate, and revise the work. We learned that:

> **Incremental development** is a staging and scheduling strategy in which the various parts of the system are developed at different times or rates and integrated as they are completed. The alternative is to develop the entire system with a big bang integration at the end.
>
> **Iterative development** is a rework scheduling strategy in which time is set aside to revise and improve parts of the system. The alternative development is to get it right the first time (or at least declare that it is right!).

Please notice that these strategies don't imply, require, or preclude each other. It is possible to work incrementally without iterating (the product manager's complaint at the start of this article), to iterate without incrementing, or to do both (the recommended strategy).

In the next section, we'll look at the definitions more closely along with how to manage the differences, but first: How did we get into the situation where the two became so confused?

Back in the 1990s, the authors of the Unified Process—perhaps so excited to be able to revisit the requirements and design portion of the process at all—decided to call everything "iterations" and "iterative development," erasing the line between the two. At that moment, the difference in the two strategies got lost. Since then, projects have failed by doing one without the other. You need both, and you need to manage them differently.

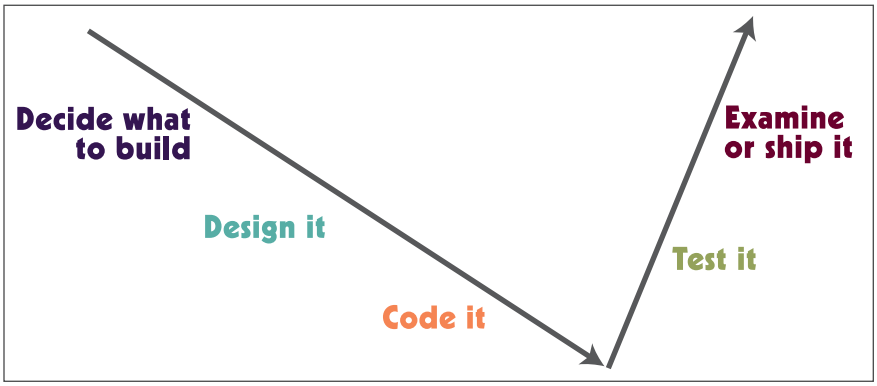


Figure 1: The Validation V is a fact of life.

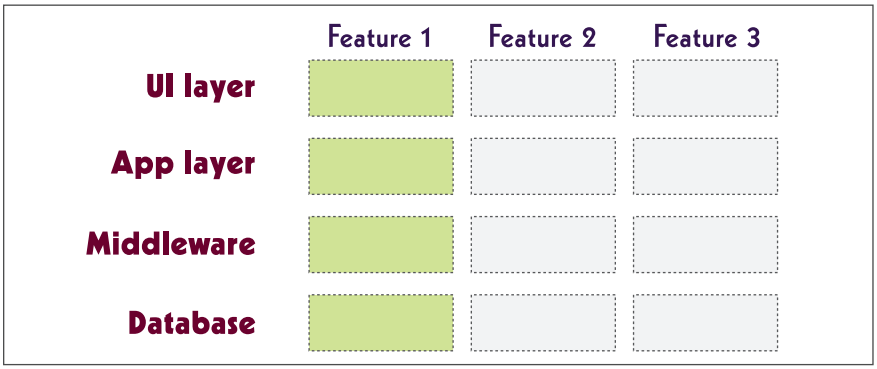


Figure 2: The first increment delivers one slice of functionality through the whole system.

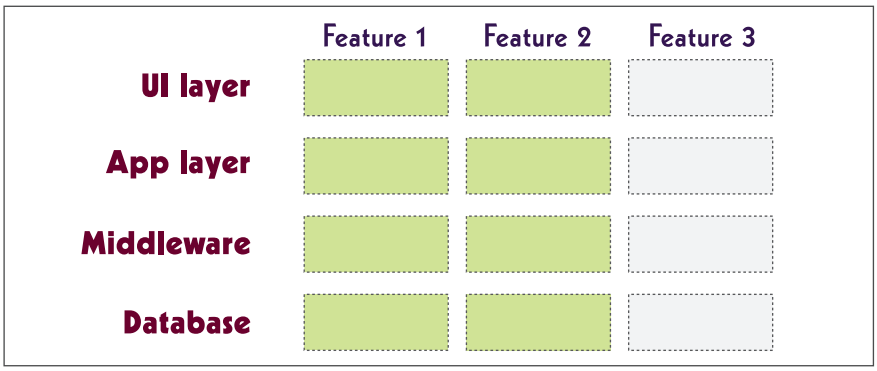


Figure 3: The second increment delivers another slice of functionality through the whole system.

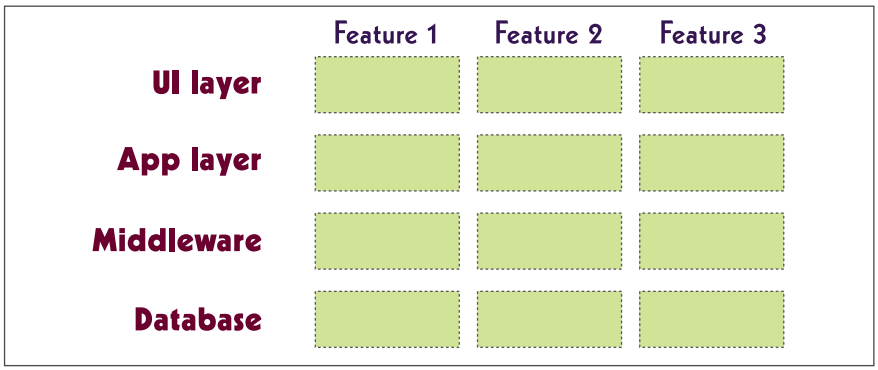


Figure 4: The third increment delivers the rest of the system.

Incremental Development

Incremental development is easier to explain and to manage, so let's start there.

First, let's get comfortable with the "Validation V" shown in figure 1. Read the figure from right to left, to see it as the dependency diagram it really is: We can't ship our system until it's been integrated, tested, and validated. It can't be tested until there's code, and we can't design and code it until we've decided what to build.

Some people see this diagram and immediately shout, "Waterfall!" (holding up the sign of the hex, or whatever). However, it is a simple fact of life, and we will take advantage of it in learning how to work with both incremental and iterative development.

You can see the Validation V in action in figures 2 through 4, illustrating *incremental* growth of the system—multiple passes through the Validation V for each new section of the system to be built.

In figures 2 through 4, the columns represent different features we wish to implement; the rows represent the different layers: user interface, application, middleware, and database.

Figure 2 shows that the first feature is completely available after the first increment, figure 3 shows that the second feature is completely available after the second increment, and figure 4 shows that the last feature is completely available after the third increment.

The figures don't show that the application or the database might have been refactored during the second or third increments. Those are iterative (rework) issues. They show the perfect incremental situation, in which additional features are added without having to revise the pre-existing features. It is rare that we get to work in such a pure incremental fashion in software development, which is why we also need to understand and manage the iterative strategy.

Incremental development is a very successful staging strategy that has been used as long as software has been built. The tragedy is that people forget to leave time to check for mistakes in understanding from the requirements gathering periods, and they forget to leave time to refine their designs. For these, they need *iterative* development.

Iterative Development

Iterative development is all about allocating time to improve the features requested, the user interface, and the performance characteristics.

Iterative development also follows the Validation V. However, instead of releasing the system at the end of the V, we examine it from various angles, particularly the three listed above.

Figure 5 shows the minor adjustment to figure 1 needed to obtain iterative development.

Figures 6 through 8 depict iterative development. These figures show iteration (revision) happening within an increment. Other strategies are possible, but this is the most common.

In constructing an iterative plan, two different strategies (and their blends) are possible. Each will work or backfire depending on the situation at hand. The art of iterative development is to guess well which strategy applies best at any given moment. The first strategy, shown in figures 6 through 8, is to develop everything as completely and correctly as possible the first time. The second strategy, illustrated in figures 9 through 11, is to develop as little as possible until things stabilize.

The idea in the first strategy is: If we can get it mostly correct the first time, it will take the least amount of time to revise just the few parts needing correction, and then the whole thing can be released. Musicians use this strategy when laying down mixes, patching only the faulty bars.

The idea in the second strategy is to expose early the places most likely to need rework, without having gone too far in the wrong direction.

To illustrate the second strategy, let's look at the art world, starting with Michelangelo's painting of the Sistine Chapel [1].

Michelangelo knew that he would be perched on a scaffold each day, staring up at a small section of the vast ceiling, mixing his paints in wet plaster and trying to get a section complete before the plaster dried (his use of incremental development!). He knew that he would not have time to paint a section of (for example) a sun-lit body, climb down from the scaffold to see how his lighting angles

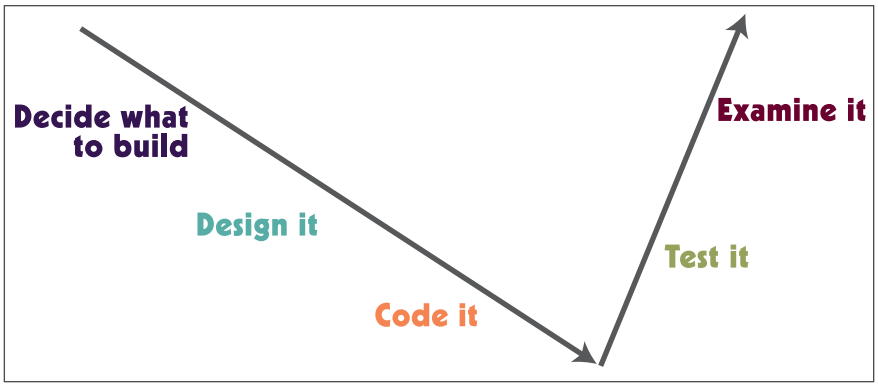


Figure 5: The Validation V for iterative development.

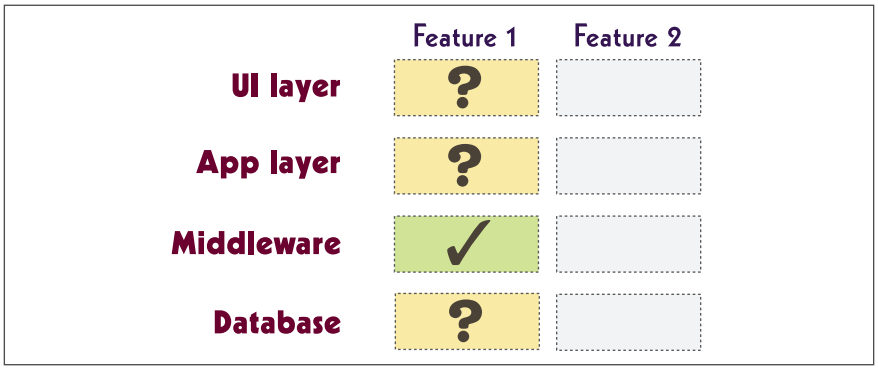


Figure 6: The first iteration delivers enough of feature 1 to evaluate what is correct and what needs revision.

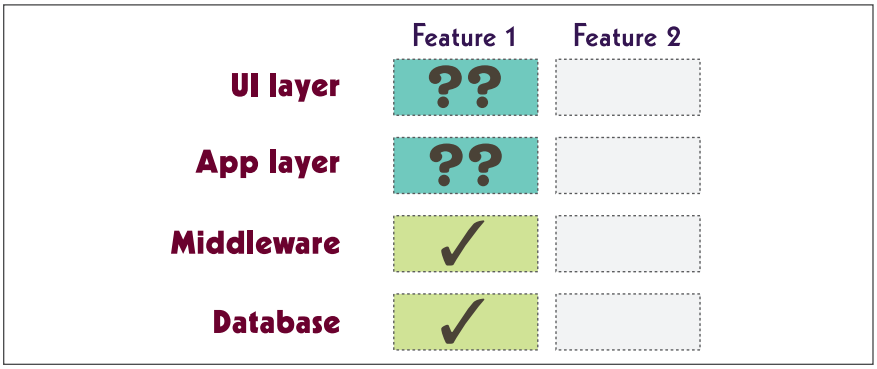


Figure 7: After the second iteration, some revised parts still need improvement.

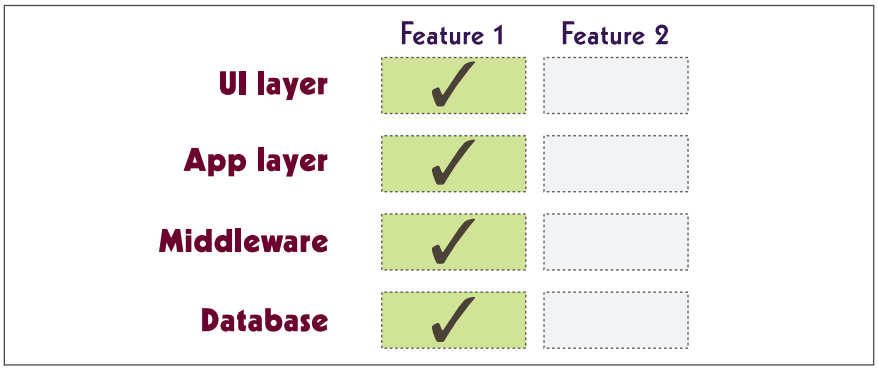


Figure 8: The third iteration produces the final and stable feature.

looked, and climb back up to the scaffold and repaint the body. This would be a bad use of time and energy—that is, a poor iterative development strategy.

He made a full-scale, rough-draft "cartoon" of the whole ceiling, so that he would know what to put in each section of the ceiling. He also made full-sized sculptures of certain human torsos, which he lit so that they would mimic the lighting from the windows in the Sistine Chapel. He sketched the lit torso, and then carried the sketch up the scaffold, so he could paint the plaster correctly the first time.

This was iterative development involving *throw-away prototypes*, a version of the second strategy.

To show more complete use of the second iterative strategy, I must invent an imaginary progression of work resulting in Leonardo da Vinci's *Mona Lisa*. Jeff Patton started this comparison [2]; I am modifying his progression to more clearly demonstrate the energy-saving value of the early iterative strategy, and to show the rework done.

Leonardo, experienced in the way patrons change their minds, first draws only a sketch, as shown in figure 9. He asks the patron for his thoughts.

The patron comments (we imagine): "Oh, no! She can't be looking right! That's not her good side. She has to be looking left!"

Happy only to have done a sketch, Leonardo proceeds and blocks out the figure and the background—see figure 10—so the shape, the colors, and the background can all be seen. He is, perhaps, one-third of the way done by time and cost, so this is a good time to get more feedback.

The patron looks at the painting so far and complains again: "You can't make her head that big! Shrink it to her body size!"

Leonardo, assessing that all the major obstacles have been met (and probably also thinking to himself that he doesn't want to give the patron any more leeway in how it looks in the end), finishes the painting without showing it again, see figure 11.

Figure 9: Iteration 1 of the *Mona Lisa*

Figure 10: Iteration 2 of the *Mona Lisa*

Figure 11: Iteration 3 of the *Mona Lisa*

We can imagine that the patron grumbles a bit at the very end, but looks at what he has spent and says, "I'd rather have her eyes bigger ... but let's call it done."

This second strategy—"Do the least work before getting feedback"—is a popular one in the software world, because of the frequency of changes and the cost of those changes. However, either of the two strategies is valid.

Managing Incremental and Iterative Development Combined

The two fit together quite naturally in what I call "VW Staging"[3], which can also be thought of as "cycle unrolling"[4]. Figure 12 illustrates. In the figure, the letter "S" means that the system ships at that point; the letter "E" means that the system only gets examined.

The figure shows that it is common to revise one subset of functionality several times before considering it good enough to ship. This is quite normal, but otherwise difficult to show on the schedule. Figure 12 shows it in an easy-to-read manner, and one that maps easily to a sponsor's calendar.

Of the two, iterative is the harder one to manage, because the question always arises as to how many times to iterate and how much to allow to change. It's not that you won't do rework if you don't schedule it—you would just like to have a way to forecast and allocate that time in advance.

On Project Winifred (see the sidebar), we developed a useful heuristic, which I recommend to this day: *Allocate two revisions for every piece of functionality*.

For the first revision, allocate one-third of the original development time. For the second, allocate half of that. These numbers are, of course, heuristic—a starting guess—but they are better than nothing. Tune the numbers for your team over time, as you gain experience.

These days, agile projects like the one at the start of the article make the mistake of not allowing for user review and correction. Let's revisit that project, which we'll call Project Laddie.

In Project Laddie, development cycles were two weeks long. Their incremental development strategy was standard XP and Scrum: Put the user stories into a list, work from the top down, and show the customer what has been built at the end of the iteration.

Their iterative strategy was to have the customer say whether to put the completed-but-not-satisfactory story back on the top of the work queue, to put it somewhere down into the work queue, or to accept it. This is normal XP/Scrum behavior.

What I learned from talking to the customers on this project was that this is not as good a strategy as I had once thought. Very few customers (or product owners) are really strong enough to say, "Return it to the front of the queue—don't work on anything new until it's satisfactory." Instead, they feel the pressure to deliver so strongly that they are torn whether to deliver on time something that is not right, or to delay the project to make it better.

What these customers were missing, and I have since come to appreciate, is that they should have several chances to view and correct the growing piece of software as it is being developed. In other words, what I am coming to call the "user's bill of rights" is that they automatically get two chances to correct

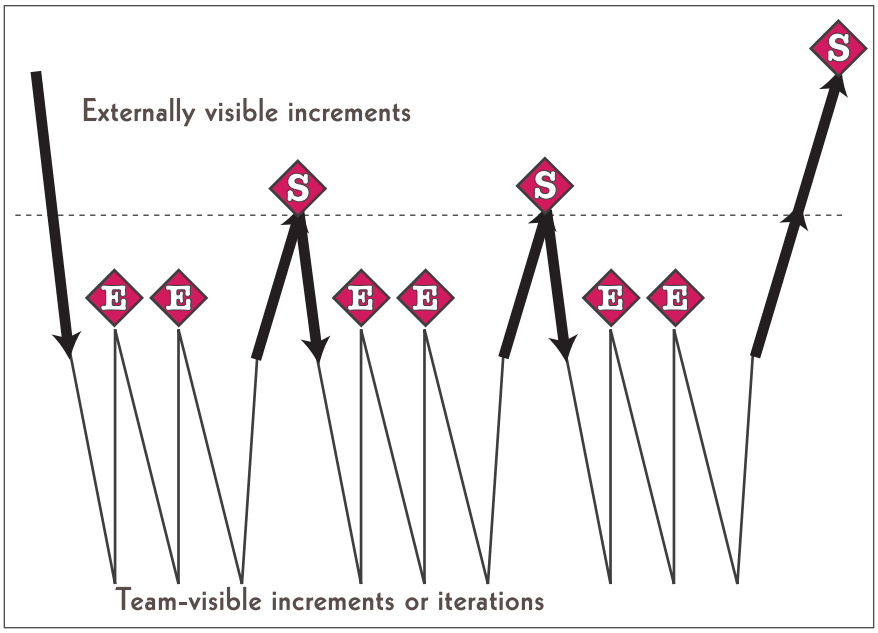


Figure 12: Integrating incremental and iterative development

the work before it gets pushed either out to the users or back into the queue[5].

In contrast to Project Laddie, Project Winifred—a forty-five-person, multi-technology project—successfully incorporated both increments and iterations.

I was pleased, recently, to run across an organization using Scrum across the company that applied this exact strategy, but within their one-month sprints. This company showed that they understood the use of both increments and iterations, and were pleased to get the benefit of both.

The Punch Line

Remember:

- The word *incremental* fundamentally means *add onto*.
- The word *iterate* fundamentally means *re-do*.

Your process, your feature set, and your product quality all need improvement. Use an incremental staging and scheduling strategy (and reflect between increments!) to improve your process and the feature set. Use the iterative, rework-scheduling strategy to improve the quality. **{end}**

References:

[1] Zipser, Karl. "Drawing, Sculpture, and the Sistine Chapel Ceiling." zipser.nl/michelangelo.html.

[2] Patton, Jeff. "The Neglected Practice of Iteration." StickyMinds.com. December 2007. www.stickyminds.com/s.asp?F=S13178_COL_2.

[3] Cockburn, Alistair. "Using VW staging to clarify spiral development." Humans and Technology Technical Report HaT TR 1997.02. alistair.cockburn.us/index.php/Using_VW_staging_to_clarify_sprial_development.

[4] Cockburn, Alistair. "Unrolling cyclic development processes." Humans and Technology Technical Report HaT TR 2004.03. alistair.cockburn.us/index.php/Unrolling_cyclic_development_processes.

[5] Cockburn, Alistair. "Three cards for user rights," Humans and Technology Technical Report HaT TR 2007.03. alistair.cockburn.us/index.php/Three_cards_for_user_rights.

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

- Links and further reading

PROJECT WINIFRED

In Project Winifred, development cycles were three months long. The incrementally growing software was released to the sales force to use at the end of each business quarter.

Their iterative strategy was to have key users visit the project twice each quarter, at about six weeks through the quarter, and again two weeks later. When the users first visited, they were allowed to change absolutely anything about the features being developed, since the system was, after all, for their internal use. On the second visit, only minor corrections were allowed. Winifred was a fixed-price project, and this rule was put into place to balance the needs of user revision and cost containment.

The users were shown real, running software (not prototypes), due to the skepticism of the project sponsors toward the contractors. Because it was the real system, it wasn't completely developed at the time of the first visit—it was perhaps only about two-thirds complete by functionality. Most of the functionality was complete at the time of the second visit. (Note the use of the *Mona Lisa* strategy!)

DEVELOPERS... START E

Rev Up Reuse with Source

VEER IMAGES

BY ALAN BERG

YOUR ENGINES

Code Search Engines

As a developer, I often have the feeling that I am writing code that others have previously written. How often have you built a standard application structure such as a GUI to modify or retrieve information from a database, built code to iterate through a sequence of objects, or scratched your head trying to determine the most efficient approach for storing data in a Perl hash? We shouldn't have to create such common code from scratch.

Many approaches are available to enable code reuse. Perhaps the most widely used at present are libraries containing frameworks, such as those for logging, presentation, or database-connection pooling. But just keeping track of the growing number of potentially viable frameworks is a headache for developers everywhere. Being heavily involved with Java, I find this burden increasing rather than decreasing over time. The Java library, useful as it is, is large enough that it is not always easy to find what I need.

Software search engines may be the answer to finding chunks of reusable code. This article reviews some popular source code search engines in common use (links are located in the StickyNotes).

What Are Source Code Search Engines?

Source code search engines have been adapted from generic search engine products. There are a significant number online, mostly commercial and paid for by advertisements. The prime example of this genre is the Google Code Search shown in figure 1. The interface is very easy to use.

Search engine spiders crawl across the Web for information, index it, and display it in ranked order. For source code search engines, the search material is code gleaned from repositories such as sourceforge.net, which houses more than 100,000 open source projects; Apache projects; and other well-known open source applications. The spiders index not only source code but also Javadoc comments and POD for Perl, as well as documentation and code-specific wikis such as Wikicodia.

Source code engines differ from generic text search engines in how they organize the results. Since code is organized into functions, classes, packages, etc., source code engines display results in a way that reflects that organization. To remain competitive, each search engine has slightly different emphases. The Krugle search engine, for example, provides excellent scoping via search-by-language and licensing options and by differentiating between comments, function definitions, function calls, and class definitions.

Source Code Search Plug-ins

Browser-based search engines target a general audience and play well to the use case of ad hoc searching. Developers,

Figure 1: The main page for Google Code Search

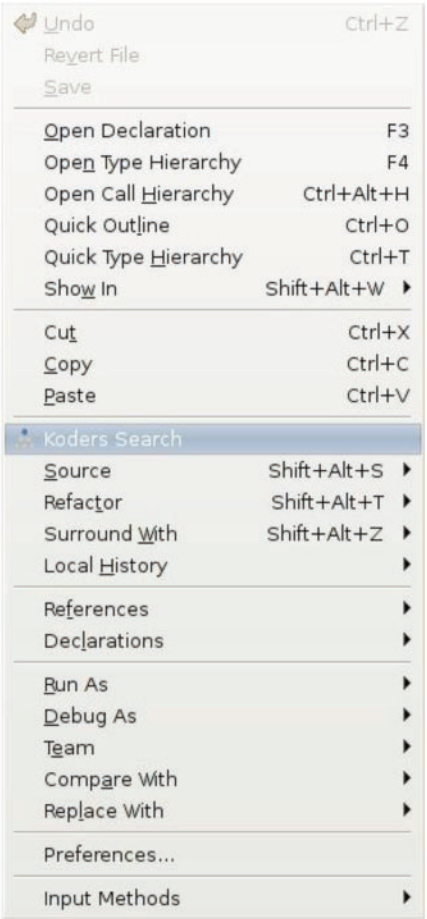


Figure 2: Koders' context-sensitive search with Eclipse

"Source code engines differ from generic text search engines in how they organize the results. Since code is organized into functions, classes, packages, etc., source code engines display results in a way that reflects that organization."

however, spend most of their time in their favorite IDE and conduct more focused searches. IDE plug-ins, therefore, can help simplify the task of finding relevant, reusable source code during development.

Eclipse has a plug-in framework that makes creation of extensions relatively straightforward. Koders has taken advantage of this by giving away a free Eclipse plug-in that adds context-sensitive searching. Installation is simple and involves unzipping a package at the top-level directory of Eclipse and then restarting the IDE. Later, as shown in figure 2, right-clicking a verb in the code pops up the usual context menu, plus a link to Koders. On selection, the online Koders site sends back search results to the IDE in the main pane, as shown in figure 3.

Google Code Search also offers an API for integrating its services into an application. For source code search plug-ins to become a commodity, however, a standard API is needed. Perhaps an XML-based solution will emerge.

The Aston plug-in for Eclipse shown in figure 4 allows for the creation of Java classes based on templates for various well-used patterns, such as Singleton and Factory Method. Installation is just a matter of dumping a jar file into the plug-in directory. After restarting Eclipse, a number of new templates will be available. Better still, if you want to build portlets or JSP/JavaScript templates, then the plug-in allows for the creation of self-defined templates to fulfill these regular creation chores.

My Own Eclipse Search Engine

I prefer tailoring existing search engine syntax to the way I think. To this end, I have installed a browse-by-query plug-in in my local Eclipse for experimentation purposes (browsebyquery.sourceforge.net/). To install it, use the typical software update method under the Help/Software Updates/Find and Install menu option. The plug-in is activated for a given project by right-clicking the project and enabling browse and query. Once activated, an agent indexes the code within the project.

If you wish to test drive the querying language before installing it in Eclipse, use the demo link on the homepage shown in figure 5. For example, to query for classes in a specific package—in this case the package "org.eclipse.core.runtime"—type:

(count of (arguments of) > 3) methods in classes in package "org.eclipse.core.runtime"

Once satisfied that the query language is expressive enough, you can activate the querying view in Eclipse and select the menu option Window/Show View/Other. A dialog similar to figure 6 will pop up, and you will need to select Browse-By-Query View from the BBQ Category.

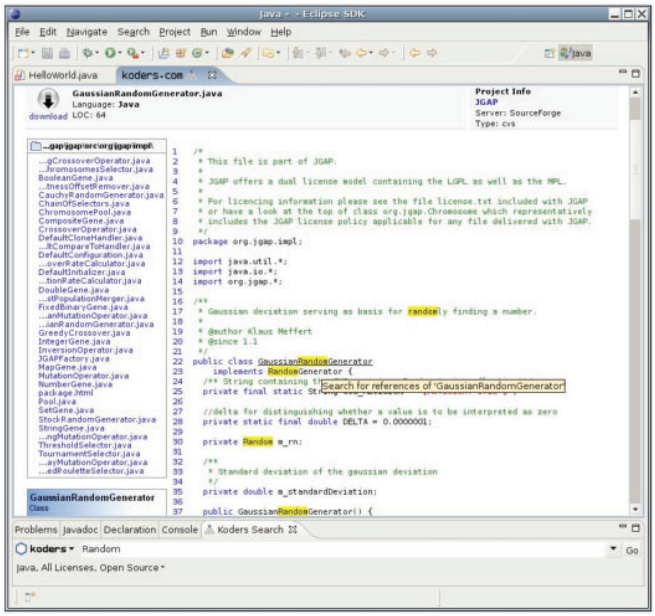


Figure 3: Search results as seen from within Eclipse

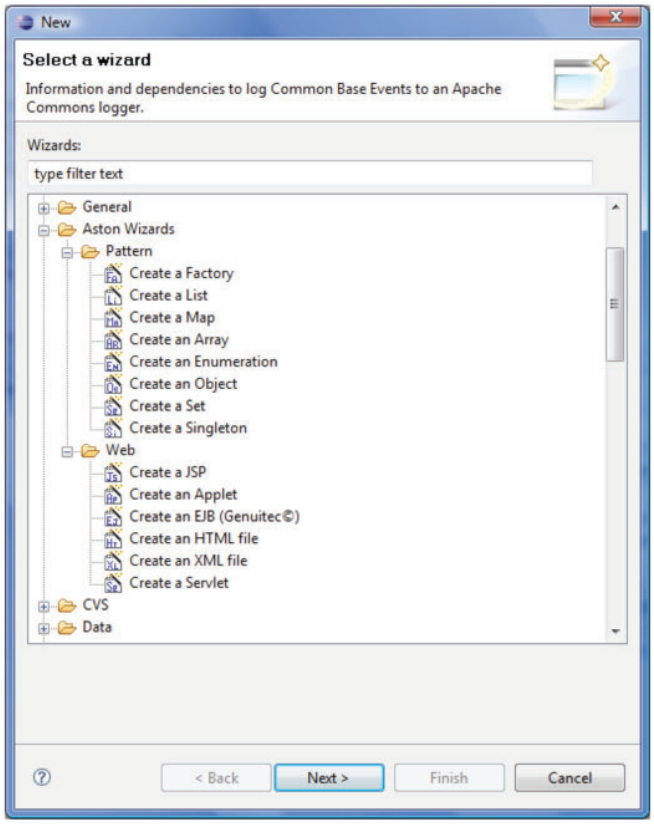


Figure 4: A list of Aston templates available from within Eclipse

browse-by-query home | antlersoft free software | sourceforge project page

Browse-by-Query Live Demo

This demo lets you query a large Browse-by-Query database, which was built from all the Eclipse 3.1 base and Java development jars as well as the Browse-by-Query code, totaling more than 200,000 methods in about 24,000 classes.

If you were using an IDE-integrated version of Browse-by-Query, you could click on one of the results to open that file and position in your editor.

Enter query or choose a sample query below:

```
(count of ( arguments of ) > 3) methods in classes in
package "org.eclipse.core.runtime"
```

Run Query

Query history:

(count of (arguments of) > 3) methods in classes in package "org.eclipse.core.runtime"

Saved Values: (no stored values)

Sample Queries:

(select one)

Results

org.eclipse.core.runtime.Status:<init>(int, String, int, String, Throwable)
org.eclipse.core.runtime.Platform:void addAuthorizationInfo(java.net.URL, String, String, java.util.Map)
org.eclipse.core.runtime.MultiStatus:<init>(String, int, org.eclipse.core.runtime.IStatus[], String, Throwable)
org.eclipse.core.runtime.MultiStatus:<init>(String, int, String, Throwable)
org.eclipse.core.runtime.Preferences.PropertyChangeEvent:<init>(Object, String, Object, Object)
org.eclipse.core.runtime.PluginVersionIdentifier:<init>(int, int, int, String)

Figure 5: A demonstration of the power of a problem domain-specific querying language

Page Ranking Static Code Reviews

Static code analysis is the reviewing of source code offline for bug patterns otherwise known as anti-patterns. Tools such as PMD and Findbugs (see the StickyNotes for links) are effective in finding common defects such as unsafe practices and unduly complex or dense code regions. For example, a common bad practice in Java is checking for string equality when one of the string objects may be null. If you call `equalsIgnoreCase` on a string object, it may throw an exception, as the following code snippet illustrates:

```
String notGood=null;
if (notGood.equalsIgnoreCase("testValue")){
      System.out.println("Matches");
}
```

A better coding pattern is to place the constant string first:

```
"testValue".equalsIgnoreCase(notGood)
```

This statement returns false instead of interrupting the smooth execution of the program.
Key problems with static code analysis include "finding" a bug that isn't there and differentiating trivial and critical mistakes. This is a pity, as one would assume that static code analysis and bug density would be logical ways to rank source code by quality.

Another weighting factor for quality is the density and code coverage of related documentation and comments. For example, one would consider Java code that has a decent proportion of relevant Javadoc comments to be of higher quality than the same code without comments.

If source code search engines are to be used to automatically measure the quality of a product before making buying decisions, then page-ranking manipulation to improve search positioning will generate enhanced recognition. If improving ranking involves removing perceived defect densities found during static code reviews, then search engines will push quality upward.

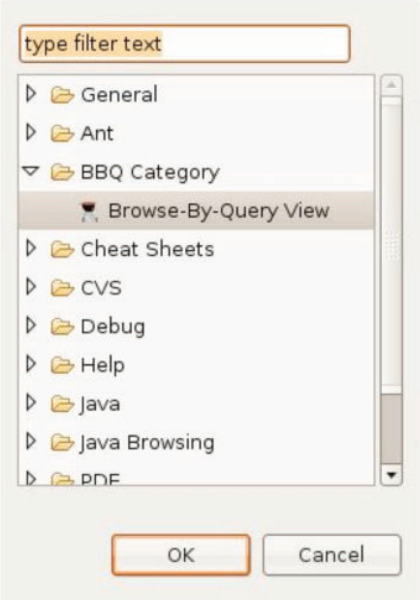


Figure 6: The Window/Show View/Other menu option

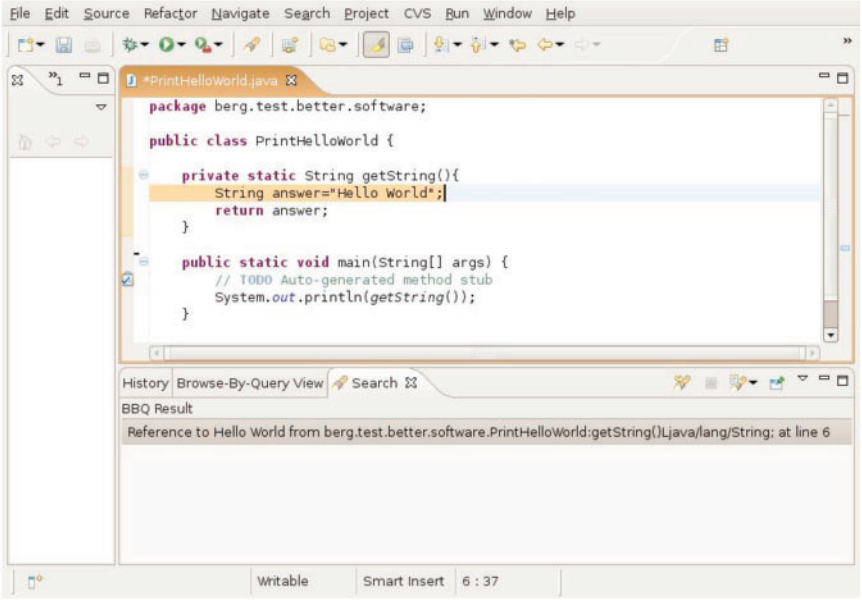


Figure 7: The main Eclipse pane with the results of a search action

Once in the view, you can query in the Browse-By-Query pane at the bottom of the Eclipse IDE shown in figure 7. A second search pane will pop up with results. Clicking a result will send you directly to the appropriate location in the relevant class, ready for editing.

A Brief Warning about Security

At this point you may be thinking about "advertising" your application indirectly by allowing the code to be indexed by a mainstream search engine. Google, for example, has a facility for uploading your code for indexing. Before uploading, I would advise you to think through your security policy if you have not already done so. Applications contain bugs and security vulnerabilities. Placing a vast amount of open source code within searchable reach allows the "black hats" to search efficiently for known attack vectors. There are well-known security anti-patterns such as trusting input from the Internet without tainting, weak or no encryption of passwords, relying on client-side checking alone, and poor session management. The open source community has responded to such threats with frequent patching. In the long run, open source search engines can help push quality upward (see the side bar Page Ranking Static Code Reviews).

Conclusion

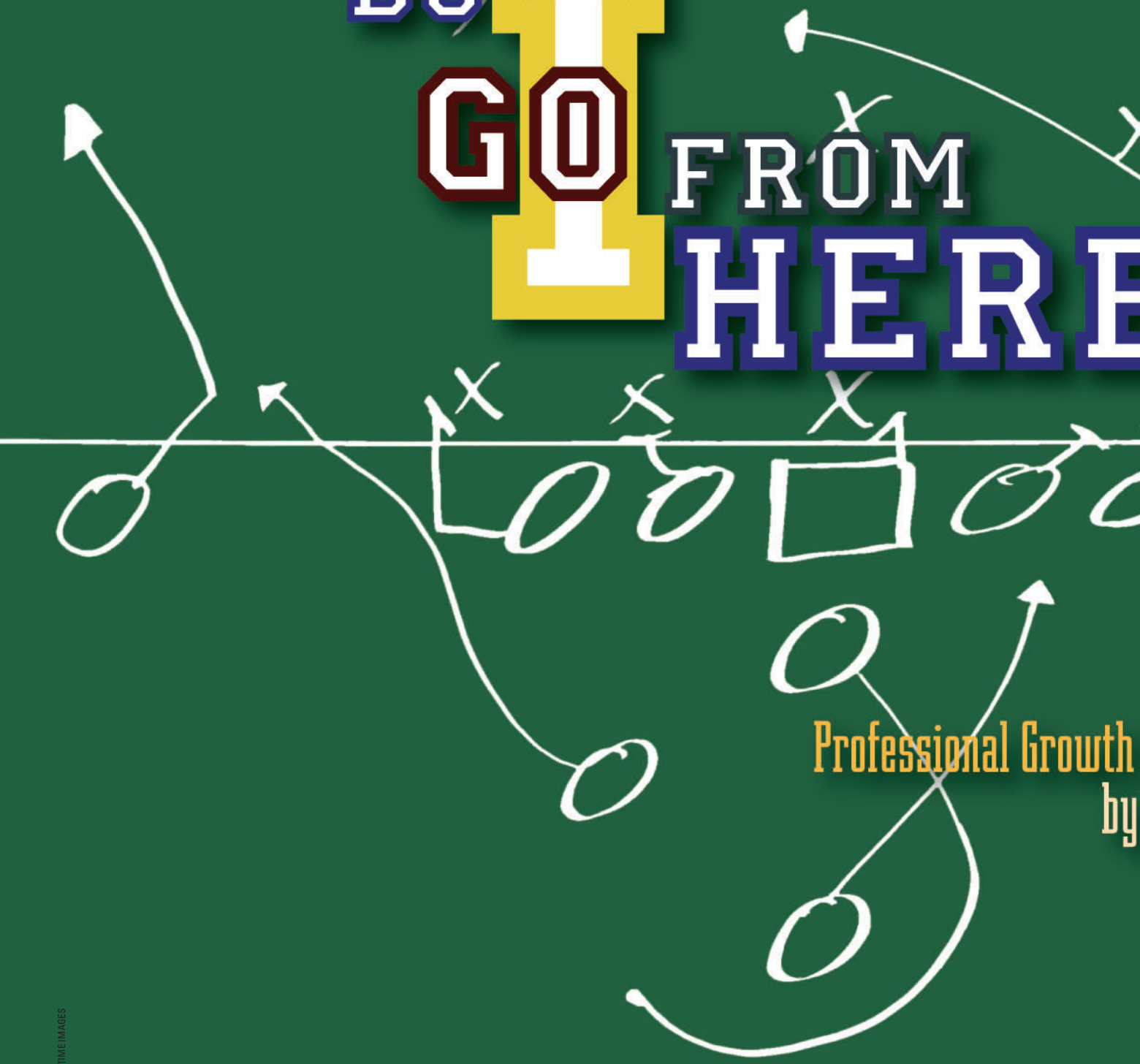
Source code search engines are positioned to be an effective tool for software developers. Since they integrate with popular IDEs, they are easily accessible during software construction. Combining their functionality with community-building technologies such as wikis and Google can make our coding efforts even more effective. **{end}**

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

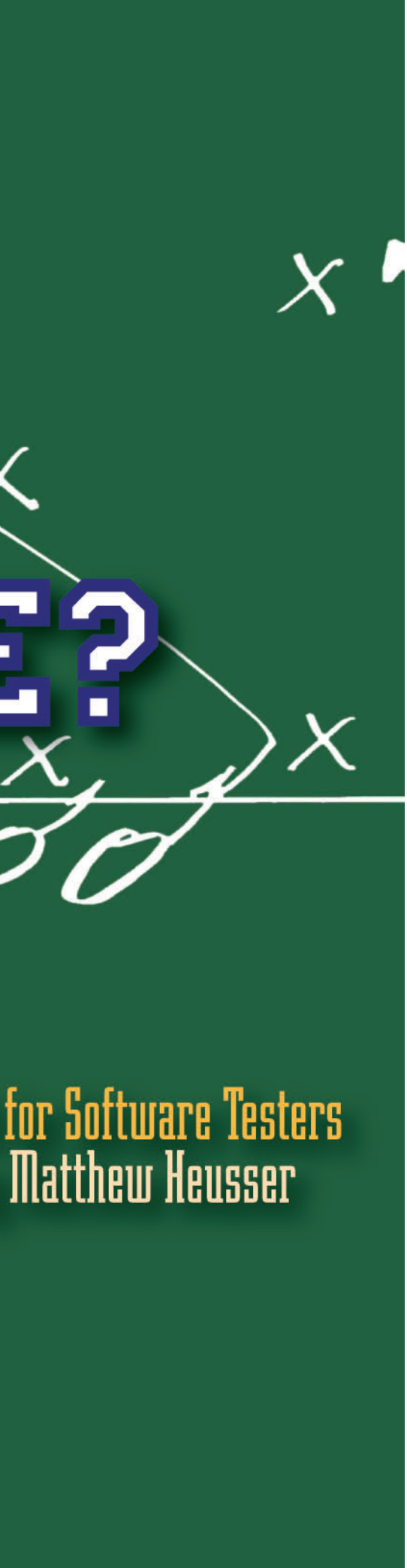
- Links

WHERE DO I GO FROM HERE



Professional Growth
by

DREAMSTIME IMAGES



for Software Testers

Matthew Heusser

I sort of fell into the testing field: I learned testing on the job from some really good people. I used to think that was exceptional, but over the years, I realized that very few people actually plan to be testers. Sometimes people use it as a stepping stone to development, but very few people set out to test as the end goal. Then, once you become a tester, the next step is unclear. The perception often seems to be the choices are stagnate or go into management.

- Ben Simo, Software Tester

Sadly, Ben is not alone in his dilemma. Where accountants and professors have a detailed professional career ladder, in many test organizations the ladder only has one rung. Ben is quick to add, "In my mind, this isn't just a problem for testers, but often for all of IT. Often the IT managers are the least qualified managers in the organization. They chose management as the default, because they saw no where else to go."

Shrini Kulkarni, a popular blog author and software tester, recently pointed out that a management job often leads to dealing with personnel evaluations, invoices, payments, recruiting, budgets, and other aspects of running a business that may not seem interesting to technical people. Many testers prefer solving technology-related problems. Therefore, many testers are more interested in a technical career ladder, but they may not see one with their current employer. If we are responsible for our own professional development, as Alicia Yanik pointed out in the December 2007 issue of *Better Software*, we need to develop our own professional growth strategy. In other words, if your company does not provide a professional development system, one option is to build your own ladder—figure out where you want to be and work with the company to recognize that growth.

ANOTHER WAY TO LOOK AT IT

Software testing is a business, and you are the product. Your goal is to increase the value of the product. There are three common strategies [1] that we can use to accomplish this goal: cost reduction, market focus, and differentiation. In other words: charge less per hour, know your customer better, or be "special."

Charging less per hour may not be feasible, but if you do, then everyone competes in this fashion and the result is a price war. The winner of a price war is whoever has the lowest cost structure.

A market-focused tester limits himself to a particular niche, such as Department of Defense contractors, medical device manufacturers, pharmaceutical companies, and so on. This can work for some contractors and job-hoppers, but if you want to stay in the same company, it doesn't help much. Everyone at your company is a specialist in that field.

The third approach is differentiation. People willingly pay twice as much for a bottle of Coca-Cola than a bottle of the generic stuff—and the same goes for automobiles, laundry detergent, macaroni and cheese, and even hotel rooms. For example, Donald Trump has managed to make his name synonymous with "expensive … and worth every penny." In one episode of his television show, he essentially slapped his name on bottles of water and doubled the price. There are so-called "name brands" in software testing, too—people who have made an effort to be better known than their peers.

This "branding" style of differentiation is reputation and perception, but sometimes the product actually can do something unique and different. For testers, this means specialization, often in a sub-field that your company does not currently have. By making this specialization explicit (defined in our job descriptions), we can demonstrate added value to the company and then build our ladder.

BUT … WHAT SPECIALTY?

Inventing a completely new specialty can be scary. It not only must be an area in which you are interested but also must be a role that your organization needs filled. But if the organization needs it, why isn't it already being done by someone else?

Jerry Weinberg suggests generating ideas for new specialties by taking two existing things and intersecting them [2]. Figure 1 shows a few examples of roles

Non-Testing Group	Required Testing Expertise	Area of Intersection - A niche to hang on
Network Admin	Zero	Performance Testing
Security Team	Little	Security Testing
Technical Writer	Zero	Documentation for the Test Team
Business Analyst	Little	Analyst/Tester
Developer	Moderate	Automation and Frameworks
Project Manager	Little	Test Estimation and Status Reports
Technical Support	Zero	Triage/Investigation of Support Tickets
DBA	Little	Database Testing

Figure 1: Overlapping roles and possible opportunities

that can be formed by intersections between testing and non-testing groups.

In each of these specialties, we find a real business need that is served by two groups. The non-testing group may not understand or appreciate what your manager needs, or it may not be serving the needs of the testing organization well. Perhaps you just like what that group does and find it interesting. By investing a little bit of time exploring the intersection of testing and other roles, you can find out enough to better fill a need. You may learn enough about the role to fill in when Bob's sick or to have a backup job in case of a layoff.

Once you have identified a specialty, work at it for six months. Get enough experience to have something to list on your annual review. If you really enjoy the work, you might want to think about where that could go in five or ten years and set out some really high-level goals. In order to have a serious ladder, the role will have to carry significant responsibility and be of real value to the company.

FROM CONCEPT TO CASH

Deciding on a field will give you a concrete direction. While having a direction is great, do you know your long-term goal or destination? Is it clear and concrete or just warm and fuzzy?

One way to develop longer-term career goals is to work backward from an envisioned future—or, in the words of business guru Stephen Covey, "Begin with the end in mind." For example, imagine that some coworkers and friends are discussing you around the water cooler five or ten years from now. What would you like them to say about you? What role are you in? While "Where do you see yourself in five years?" might be a clichéd interview question, "Where do I want to be in five years?" is a much more serious one.

Once you have goals that are specific and realistic, you can time-box them—which is as simple as creating your own deadlines to provide motivation. With time-boxed goals, you can create subgoals, or tactics, that are smaller and smaller until the pieces are manageable enough to accomplish. We do this every day in testing and project management; it really isn't that hard to apply to career growth.

If your goal is to be the go-to guy, you might sit down with your manager and ask for additional responsibility on a highly visible project. If your goal is to be a developer/tester, the first step might be to go to the library and check out two or three books on development. If your goal is to earn a college degree, the first step might be to find a program that is a possibility, and then take a single college course.

Along the way, your goal might change. You might find that you dislike the pressure of constant delivery, or that college course might lead you to a second career as a journalist. The idea is to do something that moves you in the right direction and then adjust your plan as you go. If you are an "agile" type, you could think of each of these steps as an iteration. Each step provides you with value, feedback, and—because the iterations are frequent—the opportunity to adjust.

LOVE THE ONE YOU'RE WITH

If your company doesn't have a technical career ladder, it is possible to develop one, especially at larger companies with an HR department that employs a compensation analyst. Microsoft, for example, has a defined ladder for every technical role, and the higher levels are well compensated. Alan Page, a test architect in Microsoft's Software Engineering Excellence group, explains that "The split between management and non-management typically happens around the 'senior' level. For example, for software development engineers in test (SDET), we have SDET I, II, senior SDET, and principal SDET. A principal SDET may also go by the title 'test architect.' Thus, our newest managers have several years of testing experience, and people who choose to remain technical are given a few more steps on the ladder to climb."

Many companies can stand to benefit from defining detailed career paths. For example, consulting companies can charge a premium by offering higher-level consultants, who have more perceived value. If you work for a company

like this, you can demonstrate how career pathing can make the company more money. At the same time, a professional ladder will mean more money for those people willing to climb to the top. (Hint: That's you.)

THE CAREER PATH LESS TRAVELED

Without a non-management career ladder, testers have a false dilemma. Again, Ben Simo tells the story: "I knew one excellent tester who was pushed into management, and he hated it. The truth was that he wanted to do excellent testing, not personnel evaluations and staffing with Excel spreadsheets. So, after some time, he convinced the company to create a test architect role."

If your company doesn't have such a path, it is up to you to carve one out for yourself. One integral part of any career management plan is marketing. As odious as that sounds to many technical people, no matter how much you know about software testing, if other people do not realize it, then no one will be knocking down your door asking to promote or hire you.

The StickyNotes for this article are full of ideas that will help you grow professionally and grow your reputation. The challenge of this kind of growth is that you want the improvement to be real, and you want the credit for it. That is the "marketing" part. To get that credit, we may need to improve in areas beyond hard technical skill.

One simple model for this is to think of inside skills (coding, exploratory testing, test automation) and outside skills (verbal or written communications, giving and receiving feedback, or supervision). While we don't have time in this article to explore all these areas, I feel obligated to mention that the gap between your current skills and where you want to be might be technical and heads-down, or it might be how you interact with the world. In sales, management, liaison, and consulting roles, outside skills can far outweigh inside skills. Even the best technician is stuck if he can't work with people or communicate his value.

Getting the credit for our work can be very challenging. When I interviewed John McConda, a performance tester for Mobius Labs, he pointed out that it is very hard, if not impossible, to put "I read a book" on a résumé. Even if you read and applied the ideas in the book, it can be hard to take that credit. John's suggestion is to learn new technology, then find opportunities to use those technologies in the workplace. Or, if you enjoy reading, then form a book club, host a brown-bag session, or champion some other form of distributed learning. Then not only can you claim it on your résumé but you also are building personal relationships, which are the best kind of recommendations.

The biggest, most positive career moves I ever made were in building things. In 1998, I founded the West Michigan Perl User's Group ("Perl Mongers"), and in 2006, I co-organized the first Great Lakes Software Excellence Conference (GLSEC.org). These led directly to jobs and interviews from hiring managers.

Building events and organizations follows the general pattern of giving things away. When I say giving things away, I don't mean a pen or a key chain, but rather ideas and your time. Start a blog or a brown bag. Learn a technique, do it, then bring the rest of your team along. It may take time, but people will notice, and when it's your time to ask for help, they may be in a position to help you. My advice is to pick two to four things that provide the best return on investment and pursue them aggressively.

TOMORROW

If your organization has a professional development ladder, it is probably vague by design. Sit down with your manager and find out specific, measurable goals to demonstrate accomplishment of the next level. If there is no ladder, the first part of the conversation is how to create one. If the organization balks, you can figure out where you want to be, what to do to get there, and how to make sure the effort is noticed. Earlier I mentioned redefining your role in a way that adds unique value to the company. After a few months of effort, you can ask that this new role be designated "tester II," "senior tester," "performance test specialist," or so on. There are several Web sites that recommend how to redefine

NETWORKING

Networking is an extremely common practice, but it amazes me how it is so poorly understood. Yes, you can go to meetings, collect business cards, and aggressively pursue relationships with people who are in a position to help you. It is even possible that this process works! It just feels a little mercenary to me. I have colleagues who "network" this way, and when I answer a call from them, the first question out of my mouth is usually something like "What do you want this time?"—only a little more polite.

When I think of networking, I think of the opposite. Instead of trying to get, I try to give—finding out what I can do to help the other person. For job seekers, that might mean an introduction. For volunteer organizations, that might mean stepping in and filling a role. At conferences, it can be as simple as listening and offering advice or, even easier, listening and offering sympathy.

A few of my colleagues also network in this way. When I receive a call from them, I am quick to answer it, because I know the conversation will be about an opportunity that is to my advantage.

Now, when 90 percent of our phone conversations are to my advantage, and the one time comes when you need my help … that is when the professional network you've built can come to your aid.

StickyMinds.com

Map out your project success using StickyMinds.com as your resource guide

Benefits of Joining Stickyminds:

- Access daily news articles geared to our industry
- Have one of our six eNewsletter delivered straight to your inbox
- Podcasts and Videocasts featuring industry experts
- Find software solutions in the comprehensive Tools Guide
- Search and read book reviews before you buy
- Post a question to your colleague on our Discussion Board
- Find the resources you need for building better software
- Submit articles or technical papers for others to download

StickyMinds.com is the Web's first and most popular interactive community exclusively engaged in improving software quality throughout software development.

Membership is free, so sign up today and start your journey to building better software! www.StickyMinds.com/join

StickyMinds.com

your job or role formally in order to advance technically; I am suggesting that you actually do it informally first, so you'll have a leg to stand on.

I would go so far as to suggest that additional work beyond forty hours a week should be invested in you, not project work. Stephen Covey refers to this as "sharpening the saw"—doing personal work to make your professional work go faster. This could be contributing to open source projects, experimenting with new technologies and techniques, or just keeping up with the software testing literature.

This research and experimenting can create some real options beyond going into management, which brings us back to our original dilemma and my colleague Steve. With nearly thirty years of software experience, Steve has worked for a large government agency, and multiple Fortune 500 companies, and earned a master's degree from Johns Hopkins University along the way. In his words, "The choice to stay technical is not a quick and easy way—in fact, you have to build your own way. Staying current is hard. I strive to add something to the résumé every six months, live a thrifty life, save and invest some of my income, maintain relationships, and contribute a little bit to my peers and my community."

Steve is quick to add: "There is nothing wrong with going into management, but my choice has been to avoid it. The problem is when you make the decision by default, or let someone else make it for you." **{end}**

REFERENCES:

1] Porter, Michael J. *Competitive Strategy*. Free Press, 1998.

2] Weinberg, Jerry. *Becoming a Technical Leader*. Dorset House, 1986.

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

- Professional resources

USER GROUPS AND PROFESSIONAL ASSOCIATIONS

Working in technology— especially software testing—can feel isolating, but believe it or not, there are people near you who struggle with the same thing. Every large metropolitan area (and most medium ones!) has formal user groups that meet regularly. User groups typically meet monthly, at lunch or dinner meetings; some may even have an annual conference or other large event. These organizations are always looking for board members, speakers, and attendees.

If you aren't the "toot my own horn" speaker type, there are a number of non-glory jobs like secretary or program chair. People appreciate and notice these contributions, especially when they are done well. Some of my strongest relationships—and a job or two—came through working as a volunteer for my local Perl user group and Extreme Programming group.

Here are a few large, national groups that may have a chapter in your area. If there is no user group in your area, the easiest way to become a recognized "name" in town is to start one.

- The Agile Alliance is an informal group that specializes in agile development. **www.agilealliance.org/show/1641**
- The Association of Information Technology Professionals (AITP) leans more toward hardware and networking, but each chapter has its own flavor. (The AITP was originally the Data Processing Management Association, so it has been around for some time). **www.aitp.org/index.jsp**
- The American Society for Quality (ASQ) has chapters all over the United States, and includes representatives from all aspects of business and government. **www.asq.org/sections/find.html**
- The Quality Assurance Institute (QAI) is a very software-focused version of the ASQ. **www.qaiworldwide.org/chapters/index.html**
- The Association for Software Testing (AST) is dedicated to software testing and just began a local chapters program. **www.associationforSoftwareTesting.org**

Code Review

No Meetings. No Paper. No Kidding.

http://localhost:8080 - Review #21: //depot/demo/primes/PrimeUtils.java - Mozilla Firefox

Accept | Mark Read | Comment | Add Defect

Line 40:

JC: What about n==2 and n==3?
SB: Oh yeah, you're right.
JC: Created Defect D10: Handle cases n==2,3
JC: ** New version uploaded **

Accept | Mark Read | Comment | Add Defect

Comment:

Submit Comment

D10 [Mark Fixed] [Edit] [Delete]
Major / Algorithm
Added 2006-12-12 19:53 by JC:
Handle cases n==2,3

```
28   *
29   * @param n the number to check
30   * @return true if <code>n</code> is prime, fal
31   */
32  public static boolean isPrime( int n )
33  {
34    // Initial check for divisibility by 2 or 3 t
35    // of 2/3rds of the cases!
36    if ( (n%2) == 0 || (n%3) == 0 )
37      return false;
38
39    // Don't have to test divisors all the way up
40    // Biggest possible divisor is sqrt(n).
41    final int maxDivisor = (int) Math.sqrt( n );
42
43    // Scan for non-trivial divisors of n.
44    for( int k = 5 ; k <= maxDivisor ; k++ )
45    {
46      if ( ( n % k ) == 0 )   // if anything divi
47        return false;
48    }
49    return true;
50  }
```

```
28   *
29   * @param n the number to check
30   * @return true if <code>n</code>
31   */
32  public static boolean isPrime( int
33
34
35
36    {
37      if ( ( n % k ) == 0 )   // if
38        return false;
39    }
40    return true;
```

Supports Best Practices and Company Processes
Automatically capture review metrics such as kLOC, person-hours, defect density, and inspection rate. No stopwatches or line-counters. Reports ensure reviews are really happening.

Improves Efficiency
Click-and-comment, chat-style interface makes code reviews fast, efficient, and even fun! Upload files from version control with a single click.

Encourages Collaboration and Builds Teams
With a framework to facilitate discussion and learning, developers work together instead of in isolation, even if separated by an ocean.

Code Collaborator

Tool-assisted peer code review with Code Collaborator means no more busy-work, marked-up print-outs, or meetings. Start a free trial today and cut up to 75% of the time out of code review.

STARTING AT
$199

www.CodeCollaborator.com

Learn how with this FREE book

Ten essays about code review including the largest case study of lightweight peer code review.
www.codereviewbook.com

SmartBear SOFTWARE

Another great tool from SmartBear, the experts in code review.

www.SmartBearSoftware.com
(877) 501-5651

BUILD-YOUR-OWN
TRAINING WEEK

Maximize the impact of your training by combining courses in one location to create a customized training week. Pair two courses and save up to $300. For a complete list of courses available, visit www.sqetraining.com or call 888-268-8770 or 904-278-0524 for pairing discount options.

SQE TRAINING
www.sqetraining.com

Improve your skills and help your organization increase its performance through targeted high-value training. Delivered by top software consultants, training through SQE Training is one of the best investments you can make to meet your business objectives.

TRAINING WEEK LOCATIONS

April 14–18, 2008	*Boston, MA*
June 2–6, 2008	*Chicago, IL*
September 8–12, 2008	*New York/NJ*

PMI Project Management Institute R.E.P.

The following courses are PMI Certified:

- Software Testing Certification
- Systematic Software Testing
- Mastering Test Design
- Test Management

Choose from 12 specialized training courses:

THREE-DAY COURSES (Monday - Wednesday)

- Systematic Software Testing
- Software Testing Certification
- Test Management
- Writing Testable Requirements
- Just-in-Time Software Testing

TWO-DAY COURSES (Thursday - Friday)

- Mastering Test Design
- Performance, Load, and Stress Testing
- Lean-Agile Testing Practices
- Requirements Based Testing
- Software Security Testing and Quality Assurance
- Exploratory Testing in Practice

ONE-DAY COURSE (Thursday)

- Test Process Improvement

ON-SITE ADVANTAGE

Let SQE Training come to you. For more information about on-site training courses, contact SQE Training at 904-278-0524 x212 or 888-268-8770 or email onsitetraining@sqe.com.

Klocwork Insight

BURLINGTON, MA—Klocwork Inc. announces the availability of Klocwork Insight.

Klocwork Insight combines the productivity benefits of fast, local desktop source code analysis with the accuracy of system-wide analysis, transforming source code analysis from a build audit report to a true, developer-support tool. Klocwork Insight enables organizations to find the most complex cross-system bugs within a developer's local build, ensuring vulnerabilities and defects are never propagated in the code stream.

Klocwork is available in two versions: Klocwork Insight for C/C++ and Java, and Klocwork Insight for Java.

Visit www.klocwork.com for additional information.

TeamCity 3.0

PRAGUE, CZECH REPUBLIC—JetBrains announces the free professional edition of its newly released TeamCity 3.0, which delivers multiple improvements in different areas and extends its support for both Java and .NET software development teams.

New features include:

- Per-project access rights with project roles—an exclusive feature of the Enterprise edition
- Build statistics charts, with declarative pluggable charts for user-defined metrics
- Pre-tested commit from Visual Studio for Subversion
- .NET Duplicates finder for catching similar code fragments of your C# and Visual Basic .NET code
- Java Inspections and Duplicates for Maven2 projects
- Version Control labeling for Subversion, CVS, Perforce, StarTeam, and ClearCase
- "Hanging" builds auto-detection and thread dump capturing, for quick feedback on Java and .NET build problems
- Display of build start/finish estimations in the build queue
- Build tags for organizing and quickly filtering builds

Visit www.jetbrains.com for additional information.

VMware Stage Manager

PALO ALTO, CA—VMware, Inc., unveils a new management and automation product that streamlines and accelerates the process of bringing new or modified applications and other IT services into production. Building on the management capabilities of VMware Infrastructure, VMware Stage Manager automates management of complete, multi-tier application environments—including the servers, storage, and networking systems that support them—as they move through various stages from integration to testing to staging and user acceptance before being released into production.

By deploying VMware Stage Manager, application owners can:

- Manage pre-production infrastructure assets utilizing them only when needed, thereby reducing server sprawl and increasing the resource utilization efficiency of IT services hosted on VMware Infrastructure
- Rapidly build up a preproduction image of complex production environments, thereby mitigating risk of errors and potential downtime
- Systematically propagate complex system changes through all phases of the process while more easily enforcing change and release management procedures
- Accelerate the completion of requests for changes to production systems

Visit www.vmware.com for additional information.

Redstone Software Puts Eggplant on the Menu for Windows

NEW YORK, NY—Redstone Software Inc., makers of Eggplant, announces it will offer its Eggplant software as part of a self-administering appliance. Designed for Windows users, Redstone's integrated Test Appliance (RiTA) enables testers to run Eggplant from their Windows PC.

Eggplant features include:

- Image Collections enable a single script to test an application or process across multiple platforms.
- Image Doctor simplifies update and maintenance of image-based test scripts in rapidly changing UI environments.
- Image Import accelerates test script development by leveraging existing image resources.
- Text Image Generation simplifies interacting with text on the screen of the system under test, including dynamic content not known until test execution time.
- Readable SenseTalk scripts and a powerful interactive debugger simplify even complex automation development and script maintenance tasks.

RiTA builds on Eggplant's capabilities, advances automated testing, and eliminates the need for testers to work on the Apple Mac platform. It provides a central facility for Windows users to build, store, manage, and execute Eggplant automated test cases from their Windows desktop. Irrespective of the platform to be tested (Windows, Linux, Unix, Symbian S60, Mac, etc.), RiTA offers a comprehensive testing solution delivering all of the capabilities of Eggplant to the Windows platform.

Email info@redstonesoftware.com or visit inwww.redstonesoftware.com for additional information.

Mindreef Enhances SOAPscope Server

HOLLIS, NH—Mindreef, Inc. announces the availability of SOAPscope Server 6.1, the latest offering of the company's flagship collaborative team-based solution for testing and verifying the quality of service oriented architectures (SOAs). Highlights include support for several OASIS WS-Security standards requested by customers. The company also introduces three new desktop products: SOAPscope Architect, SOAPscope Tester, and SOAPscope Developer.

The Mindreef SOAPscope Server product family allows teams and individuals to choose the most effective mix of product features and pricing based on their needs and those of their SOA project. The new and enhanced products combine valuable Mindreef features into targeted offerings:

AGILE SOFTWARE DEVELOPMENT TRAINING

Accelerate Your Career & Empower Your Team

BUILD-YOUR-OWN TRAINING WEEK

Maximize the impact of your training by combining courses in one location to create a customized training week. Pair two courses and save up to $300. For a complete list of courses available, visit www.sqetraining.com or call 888-268-8770 or 904-278-0524 for pairing discount options.

SQE TRAINING
www.sqetraining.com

Improve your skills and help your organization increase its performance through targeted high-value training. Delivered by top software consultants, training through SQE Training is one of the best investments you can make to meet your business objectives.

AGILE SOFTWARE DEVELOPMENT TRAINING WEEK LOCATIONS

May 12–16, 2008	*Boston, MA*
June 2–6, 2008	*Seattle, WA*
September 8–12, 2008	*Washington, DC*

Choose from 4 specialized training courses:

THREE-DAY COURSES
(Monday - Wednesday)

- Scrum Master Implementation Workshop
- Practical Test-Driven Development

TWO-DAY COURSES
(Thursday - Friday)

- Design Patterns Explained
- User Stories and Estimation in Agile Development

ON-SITE ADVANTAGE

Let SQE Training come to you. For more information about on-site training courses, contact SQE Training at 904-278-0524 x212 or 888-268-8770 or email onsitetraining@sqe.com.

- SOAPscope Server 6.1 encapsulates all of the features that Mindreef provides for architects, developers, testers, and support engineers into a single, server-based solution that allows entire project teams to collaborate effectively while delivering well-tested, scalable, and policy-compliant Web services and composite applications.
- SOAPscope Architect 6.1 is a design-time governance and SOA quality and testing platform for policy rules authoring, design-time support, prototyping, change-time, and run-time support.
- SOAPscope Tester 6.1 provides an integrated set of tools for Web services performance and load testing, design-time support, prototyping, change-time, and run-time support.
- SOAPscope Developer 6.1 is a cost-effective desktop platform with integrated tools for problem diagnosis and resolution, unit testing, and supporting service consumers.

Visit www.mindreef.com for additional information.

SOAPSonar Version 3.5 for SOA Testing

BOSTON, MA—Crosscheck Networks, Inc. announces the immediate availability of SOAPSonar version 3.5. The following features enable testers to easily validate increasingly sophisticated SOA Web services deployments:

- Direct database integration for end-to-end SOA testing
- Dynamic on-demand binding for abstract types
- Setting of global policies that can be assigned to custom test groups
- JUNIT integration that enables SOAPSonar to leverage JUNIT framework
- Direct native Java key store PKI support
- Optimization techniques to parse jumbo WSDLs and schemas

Based on patent-pending XSD-mutation technology, SOAPSonar delivers on the four pillars of SOA Web services testing: functional regression, performance testing, security assessment, and compliance validation. These four pillars ensure robust, scalable, reliable, and secure Web services are deployed within an enterprise.

Visit www.crosschecknet.com for more information.

Dynamic Leak Check for HP-UX Itanium

NORTH HAVEN, CT—Dynamic Memory Solutions announces the latest release of the software debugging tool Dynamic Leak Check for HP-UX Itanium and introduces support for 32-bit applications.

Dynamic Leak Check for HP-UX Itanium is a high-speed memory leak and heap corruption detector. Dynamic Leak Check detects many difficult to diagnose software errors in C and C++ applications. It now supports both HP-UX 64-bit and 32-bit applications running on Itanium architecture.

For additional information, visit www.dynamic-memory.com or call 877-293-4144.

Response Time Instrumentation

FAIRFAX, VA—Response Time Instrumentation (RTI) for Internet Explorer, from OC Systems, Inc., measures system response times as experienced by the user during transactions initiated within Microsoft Internet Explorer. RTI monitors the start of an action (i.e., when the user requests a page or submits data) and when the result is returned and rendered. This data is used by performance engineers to reduce application slowdowns and improve the user experience for Web-based applications or applications that have a browser-based front end.

RTI can be used in a standalone manner or integrated with an application management system. When used in standalone mode, RTI includes its own agent to collect and transmit data. The information RTI provides can be easily understood by help desk and operational users, while the available context data provides actionable information to the development teams when production problems occur.

Out of the box, RTI also integrates with any agent or management system that uses the established and open industry standard for Application Response Measurement (ARM) to provide immediate transaction tracing and decomposition from a user's PC through the back-end servers supporting the request.

Visit www.ocsystems.com for additional information.

To load test your website, you could type this:

```
Definitions
  ! Standard Defines
  Include "RESPONSE_CODES.INC"  Include "GLOBAL_VARIABLES.INC"
  CHARACTER*512 USER_AGENT  Integer USE_PAGE_TIMERS  CHARACTER*
  CHARACTER*1024 cookie_2_0  CHARACTER*1024 cookie_2_1 Timer T_
Code
  !Read in the default browser user agent field
  Entry[USER_AGENT,USE_PAGE_TIMER  Start Timer T_OBFUSCATED
  PRIMARY GET URI "http://yahoo.cHTTP/1.1" ON 1 &
  HEADER DEFAULT_HEADERS &
  ,WITH {"Accept: image/gif, image/xbitmap, image/jpeg, image/p
  "application/x-shockwave-flash, application/msword, */*", &
```

or this:

www.webperformanceinc.com

Why code every test case by hand, when our unique software detects and automatically configures the test cases for you – quickly and accurately, then gives you superior reports that are easy to understand? With Web Performance automatic load testing, the time and money you save could increase productivity as much as 500 percent.

For more information about how you can increase performance and productivity using Web Performance automated load testing, visit **www.webperformanceinc.com**

10 Things You Might Not Know About Sources of Testing Ideas

by Robert Sabourin

I am always on the lookout for new testing ideas. Some of the most valuable test ideas come to mind when I am actually testing. Here is a list of ten sources of test ideas I use on Just-in-Time (JIT) testing projects.

1 Capabilities. Capability-based test ideas focus on confirming that an application does what it is supposed to do. Requirement and functional specifications can be used as a source of capability-based testing ideas.

2 Failure Modes. Failure mode-based test ideas are "what if" questions that often are inspired by how a system is designed. I look at all of the objects, components, and interfaces in a system and ask myself, what if they break or exhibit some sort of unanticipated failure. Failure modes can be the result of harshly constrained system resources or forced error conditions.

3 Quality Factors. Quality factors are characteristics of a system that must be present for the project to be successful. Quality factors are the "ilities" including: usability, reliability, availability, scalability, and maintainability. Quality factor test ideas often involve experiments to determine if a quality factor is present. Examples include performance, load, and stress testing.

4 Usage Scenarios. A usage scenario test idea challenges whether a user can achieve his tasks with the software under test. To paraphrase the Kennedy inaugural address—we ask not what the software does for the user, but rather we ask what the user does with the software. Usage scenario-based test ideas involve identifying who is using the system, what he is trying to achieve, and in what context.

5 Creative Ideas. Creative test ideas come from many sources. I often use deliberate lateral thinking techniques (for example, Edward De Bono's Six Thinking Hats) to come up with cool and effective tests. I also use metaphors to come up with testing ideas. I wonder what would happen if the Tasmanian Devil used the system. Perhaps a Dr. Seuss story inspires some testing ideas. Perhaps great detectives or movies. Pretty much anything goes here.

6 States. When testing an application, a state model helps me come up with test ideas. For example, a transaction goes through many states of existence from creation through approval, payment, and delivery. I use state models to identify test ideas such as getting to states, exercising state transitions, and navigating paths through the system.

7 Data. Data is a rich source of testing ideas. Data flow paths can be exercised, different data sets can be used, data can be cooked up and built from combinations of different data types, and stored procedures can be verified. Test ideas can be developed to create, update, and remove any persistent data.

8 Environments. Exploring how the application behaves in different operating environments is a rich source of testing ideas. Environment test ideas can relate to varying the platform, hardware, software, operating system, co-resident third-party software, and locales.

9 White Box. White box test ideas come from reviewing low-level design, code, and data schemas. White box test ideas include exercising the code, decisions, and paths through the program. White box test ideas look at the work products of developers to find ways to challenge the application under test.

10 Taxonomies. One of my favorite sources of test ideas is bug taxonomies. These are organized, documented collections of bugs. I peruse taxonomies randomly in search of interesting bugs. If the bug could happen in my application, I come up with a test idea to expose it. Three taxonomies I often use are:

- Appendix A of *Testing Computer Software* (Kaner, Falk, and Nguyen)
- Boris Beizer Taxonomy (Otto Vinter)
- Shopping Cart Taxonomy (Giri Vijayaraghavan)

Software Quality and the Prisoner's Dilemma

by Paco Hope

Twice recently I have read the "prisoner's dilemma" applied to unusual and very different situations. For those of you unfamiliar with it, the explanation of the "classical" dilemma at Wikipedia.org (see the StickyNotes for a link) is sufficient:

Two suspects are arrested by the police. The police have insufficient evidence for an outright conviction, so they visit each prisoner to offer the same deal: If one testifies against the other and the other remains silent, the betrayer goes free and the silent accomplice receives the maximum ten-year sentence. If both stay silent, both prisoners are sentenced to only six months in jail for a minor charge. If each betrays the other, each receives a five-year sentence. Each prisoner must make the choice of whether to betray the other or to remain silent. However, neither prisoner knows for sure what choice the other prisoner will make. So this dilemma poses the question: How should the prisoners act?

This mental game gives us remarkable insight into why software managers who forego quality in order to reach a short-term market advantage are actually acting rationally. Equally remarkable is how irrational that short-term sacrifice seems when we realize that we're not playing the classical dilemma in the software industry. Before we look at the software industry, let's consider a couple of other ways the prisoner's dilemma plays out in the world.

An article in *The Economist* [1] discusses how most nation-states are treating climate control as a sort of prisoner's dilemma. The author writes: "Rational leaders will always neglect the problem, on the grounds that others will either solve it, allowing their country to become a free-rider, or let it fester, making it a doomed cause anyway. So the world is condemned to a slow roasting, even though global warming could be averted if everyone cooperated."

What does climate change have to do with software quality? Vendors and software makers are in positions very similar to nation-states. If you replace "global warming" with "software quality" and replace "nation-states" with "software maker," you can see the analogy. Consider the state of software quality: It is universally bad. Rational software managers can ignore quality, assuming that something will come along and solve their problems (a new tool, a new methodology, the next manager on the project), or they can recognize poor quality but let it fester—because when compared to their competitors' software quality, everybody's products are equally bad.

In an unrelated story, *Smart Money* magazine [2] invokes the prisoner's dilemma with respect to retail banks. "Forget coffee joints and nail salons. The new scourge of Main Street is … bank branches." The author reports that in retail banking, if a bank gets a slim majority of branches, say, 51 percent, the bank can experience disproportionately high majorities in deposits, say, 89 percent. If a bank does not open as many new branches as its competitors, it is sure to be squeezed out of the market. If it keeps up with its competitors, however, it only ensures mediocre performance for everyone. In fact, the situation has gotten so bad that many individual branches do not make a profit.

> "As long as we view the tradeoffs in software quality to be isolated, individual rounds of the prisoner's dilemma, we will make the rational choice to emphasize short-term revenue over long-term quality."

Now look at software from the banking perspective. This time replace "bank" with "software maker" and replace "branches" with "features." Marketing likes to remind us that new features drive new sales. It's true that a few key features might capture a disproportionately large market share. Since we know "more code means more bugs," however, we know that this race for features can cost us in quality in the long run. Sacrificing quality to get those last few features won't actually ensure us a disproportionate share of the market, because all of our competitors are sacrificing quality to get their features faster. So, similar to the retail banking situation, this approach to quality ensures uniformly poor quality across the industry.

So a rational player will always choose the middle outcome that is bad both for him and all the other players because it is clearly in his best interest. That fact is only true, however, in the classical dilemma where the consequences are meted out just once and the "prisoners" cannot communicate. *The Economist* points out that game theorists have long known if the same players play an unbounded number of rounds of the dilemma, and if they know what the other players have chosen in prior rounds, justice can be served and cooperation is encouraged. Aggressive play is punished by cooperative players in subsequent rounds.

Thus, it can be with software quality. If poor quality is punished through market forces, the manager or company who neglected quality to achieve a perceived short-term market advantage will be punished in subsequent rounds when sales of buggy software lag or schedules lengthen to spend time fixing bugs. As long as we view the tradeoffs in software quality to be isolated, individual rounds of the prisoner's dilemma, we will make the rational choice to emphasize short-term revenue

over long-term quality. When we realize that we are playing a long-term game, the opposite trade-off becomes the clearly preferred choice.

This analogy of software and the prisoner's dilemma is not flawless. Note that playing nicely is only the right strategy if the number of rounds you play is unbounded. If we know we're playing *n* rounds, then the obvious strategy is to cooperate until round *n-1*, then play selfishly. However, since everyone will do *that*, you have to cooperate until round *n-2* and then play selfishly. But, since everyone will do that, a bounded game of the prisoners' dilemma ends up degenerating to the same level as a one-round game. A rational player will only play cooperatively if he believes the game is unbounded.

It is tempting to believe that our fast-paced, short-lived software project has a small, finite number of rounds before the software itself has no life left. That leads us back to our one-round, selfish strategy. The flaw in that reasoning is that every business need—every reason for writing software—is usually long-lived, even if our software is not. Thus, solving the business need is playing the unbounded game. Sacrificing quality in today's software is playing selfishly for short-term gain and will be punished in the next round of the long-term game. Those who solve the business need and maintain quality will know how to do that again when it's time to retire the current solution and build the next. We all need to remember that we're playing the game for an unbounded number of rounds, and the natural outcome of playing the selfish strategy this round is to be punished in the next. **{end}**

REFERENCES:

1] *The Economist*. Playing Games with the Planet. September 2007.

2] *Smart Money*. Tough Customer. November 2007.

Sticky Notes

For more on the following topic go to www.StickyMinds.com/bettersoftware.

- Prisoner's dilemma

HAVE THE LAST WORD!

If you have a point to make or a side to take on issues and trends that affect the industry, we want to hear from you.

We are looking for insightful, thought-provoking commentary for possible use as a Last Word column.

Please send your submission to **editors@bettersoftware.com.** Subject: The Last Word.

You will be notified if you are being considered for publication.

Index to Advertisers

AutomatedQA	www.testcomplete.com/try	Inside Back Cover
Better Software magazine	www.stickyminds.com/april	21
Empirix	www.empirix.com	1
Hewlett-Packard	www.hp.com/go/software	Back Cover
IBM	www.ibm.com/rational	22
Net Objectives	www.netobjectives.com	Inside Front Cover
PNSQC	www.pnsqc.com	19
Rally Software	www.rallydev.com/bsm	16
SmartBear	www.codecollab.com	41
SQE Agile Training	www.sqetraining.com/Agile	44
SQE Better Software Conference	www.sqe.com/BetterSoftwareConf	17
SQE Testing Training	www.sqetraining.com/Testing	42
STAR*EAST* 2008	www.sqe.com/StarEast	5
StickyMinds.com	www.StickyMinds.com	40
TechExcel, Inc.	www.techexcel.com	2
Web Performance, Inc.	www.webperformance.com	45

Display Advertising
Shae Young syoung@sqe.com

All Other Inquiries
info@bettersoftware.com

Better Software (USPS: 019-578, ISSN: 1532-3579) is published ten times per year. Subscription rate is US $49 per year. A US $35 shipping charge is incurred for all non-US addresses. Payments to Software Quality Engineering must be made in US funds drawn from a US bank. For more information, contact info@bettersoftware.com or call (800) 450-7854. Back issues may be purchased for $15 per issue (plus shipping). Volume discounts available.

Entire contents © 2008 by Software Quality Engineering (330 Corporate Way, Suite 300, Orange Park, FL 32073), unless otherwise noted on specific articles. The opinions expressed within the articles and contents herein do not necessarily express those of the publisher (Software Quality Engineering). All rights reserved. No material in this publication may be reproduced in any form without permission. Reprints of individual articles available. Call for details.

Periodicals Postage paid in Orange Park, FL, and other mailing offices. POSTMASTER: Send address changes to Better Software, 330 Corporate Way, Suite 300, Orange Park, FL 32073, info@bettersoftware.com.

SICK AND TIRED OF MANUAL TESTING?

TestComplete™

Sensible price. Superior automated testing.

Test **.NET** Delphi **Java**

Web Automated **Windows** Test Desktop

Load Easy **Vista** Test Synchronized

Fast Distributed Powerful Client/Server

✓ NEW! Record Test Checkpoints

✓ NEW! Better & Faster Web Tests

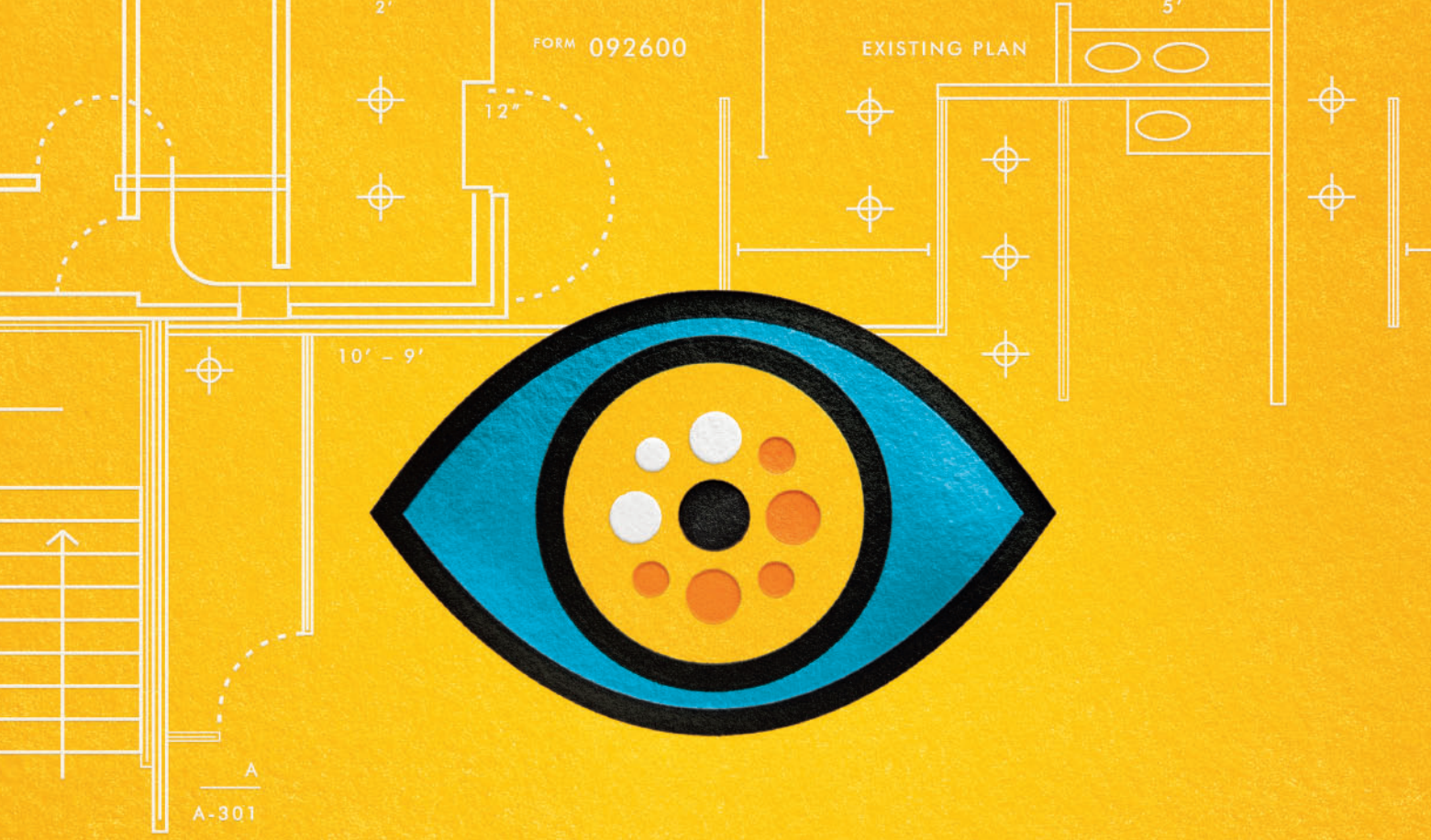
✓ NEW! Test Your 64-bit Apps

✓ NEW! Test Your Web Services

Try TestComplete™ free for 30 days and enter to win an **Apple iPod Touch**
www.testcomplete.com/try

FREE TRIAL - DOWNLOAD NOW
www.testcomplete.com/try

AutomatedQA
test, debug, deliver!
(978) 236-7900



ALTERNATIVE THINKING ABOUT SERVICE-ORIENTED ARCHITECTURE:

Rethink The Architecture, Rethink The Business. (Rethink Your Bottom Line.)

Alternative thinking is shifting your SOA strategy to focus on business initiatives rather than technology initiatives. (Head in the game, not on bits and bytes.)

It's using governance to speed the adoption and reuse of services—keeping everything from getting out of hand.

It's building quality and management into every step, from idea through execution, to help ensure you're always up and running smoothly.

It's capitalizing on software that works across platforms, so your business operates efficiently through mergers and acquisitions and alliances and massive growth.

It's choosing the right SOA partner so you can get on with the business of the business. (Hello, tomorrow.)

Technology for better business outcomes. hp.com/go/soa

©2008 Hewlett-Packard Development Company, L.P.

BETTER SOFTWARE CONFERENCE & EXPO

JUNE 9–12, 2008
LAS VEGAS, NEVADA
THE VENETIAN

SOFTWARE QUALITY ENGINEERING CONFERENCES
www.sqe.com

Conference Sponsor: RALLY SOFTWARE

KEYNOTES BY INTERNATIONAL EXPERTS

Ken Schwaber
Advanced Development Methods, Inc.

Michael Mah
QSM Associates

Jean Tabaka
Rally Software Development

Johanna Rothman
Rothman Consulting Group, Inc.

TUTORIALS WORKSHOPS CLASSES

CELEBRATING 5 FIVE YEARS

Alan Shalloway
Andy Glover
Andy Hunt
Andy Kaufman
Ari Takanen
Beth Layman
Bob Hartman
Chris Ronak
Chuck Allison
Dan North
David Garmus
David Herron
David Spann
Ed Weller
Elle Ringham
Guruprasad Gopalakrishnan
Herbert (Hugh) Thompson
Hubert Smits
James McCaffrey
James Newkirk
Jared Richardson
Jean Tabaka
Jeff Patton
Jeff Payne
Jerry Smith
Jimmy Xu
Johanna Rothman
John Janakiraman
Julie Gardiner
Ken Pugh
Ken Schwaber
Kent McDonald
Kevin Bodie
Lee Copeland
Lee Devin
Linda Rising
Linda Westfall
Lisa Crispin
Michael Mah
Michele Sliger
Mike Seavers
Mike Tholfsen
Mitch Lacey
Nelson Perez
Paco Hope
Payson Hall
Pollyanna Pixton
Rebecca Wirfs-Brock
Richard Bender
Rob Myers
Robert Galen
Stacia Broderick
Tim Korson
Todd Little
Vladimir Pavlov
Will McKnight

www.sqe.com/bsce

BETTER SOFTWARE CONFERENCE & EXPO

JUNE 9–12, 2008
LAS VEGAS, NEVADA
THE VENETIAN
www.sqe.com/bsce

CONTENTS

4 Conference at-a-Glance
Build Your Own Conference!

6 Special Events

7 Conference Speakers

8 49 In-Depth Pre-Conference Tutorials

16 4 Keynote Presentations

18 42 Concurrent Classes Covering:
- Managing Projects and Teams
- Agile Development
- Plan-Driven Development
- Process Improvement and Measurement
- Testing and Quality Assurance
- Security and Special Topics

24 Event Location and Las Vegas Highlights

25 What Our Delegates are Saying

26 Sponsors and Exhibitors

27 Registration Information

LETTER FROM THE PROGRAM CHAIR

Dear Colleague,

I'm Lee Copeland, Program Chair of the Better Software Conference & EXPO. On behalf of Software Quality Engineering, I'm pleased to invite you to attend this exciting conference, now in its fifth year. Software Quality Engineering has a long history of producing great conferences. The STAR*EAST* and STAR*WEST* testing conferences are in their sixteenth year. The Agile Development Practices conference made its successful debut last December.

The four-day Better Software Conference & EXPO kicks off on Monday and Tuesday with two days of pre-conference tutorials and workshops (both full- and half-day) led by industry experts. Then, during the two-day conference on Wednesday and Thursday, we combine keynotes, concurrent classes, and the EXPO for a fantastic program. Four world-class experts present keynote addresses to enlighten and inspire you. Forty-two concurrent classes allow you to create your own conference specific to your personal and organizational needs.

Choose from a wide variety of presentations on agile development, project management, leading people and teams, testing and QA, requirements, architecture, design, metrics, security, and other vital topics.

At the EXPO you can meet one-on-one with representatives of today's most progressive and innovative organizations. Join us in Las Vegas where there's never a dull moment. Whether you're attending the conference by yourself or you've brought the family along for the excitement, there's something for everyone. The Venetian is one of Las Vegas' finest conference venues. I look forward to seeing you at the Better Software Conference & EXPO.

"What happens in Vegas stays in Vegas" is their advertising slogan, but what you'll learn in Vegas at the Better Software Conference & EXPO will benefit your organization for years to come.

Sincerely,

Lee Copeland
Program Chair

CELEBRATING 5 FIVE YEARS

WHO'S BEHIND THE CONFERENCE?

SOFTWARE QUALITY ENGINEERING
www.sqe.com

Software Quality Engineering assists professionals interested in improving software practices. Four conferences are hosted annually—the STAR conference series, the Better Software Conference & EXPO, and Agile Development Practices. Software Quality Engineering also delivers software training, publications, and research. **www.sqe.com**

BETTER SOFTWARE MAGAZINE

Better Software magazine brings you the hands-on facts you need to run smarter projects and to deliver better products that win in the marketplace. **www.BetterSoftware.com**

StickyMinds.com

StickyMinds.com is a complete online resource to help you produce better software. It offers original articles from industry experts, technical papers, industry news, a tools guide, forums, and much more. **www.StickyMinds.com**

TOP 10 REASONS TO ATTEND THE BETTER SOFTWARE CONFERENCE & EXPO

1. Discover the latest in software development technologies, trends, and practices
2. Find the right balance between the plan-driven and agile approaches
3. Learn new ways to lead and inspire your software teams
4. Discover effective measurement approaches to optimize your software investment
5. Address your software security issues at the source of the problems
6. Speak with leading industry experts regarding the latest software development technologies
7. Benefit from real-world experiences of leading software development organizations
8. Network and receive strategic insight and tactical advice from peers and experts
9. Attend the EXPO for the latest tools and services to help you build and deliver Better Software
10. Finally ... It's Vegas, Baby!

BRING A BUDDY DISCOUNT

Bring a colleague and each of you saves an additional $200. Register any two people at the same time for the full conference and save $200 off each registration. **Please call the Client Support Group at 888-268-8770 or 904-278-0524 to register.** Use promotion code **BBBS** when registering to receive your discount.

WHO SHOULD ATTEND?

Software professionals seeking the latest practices in software development today—from the highly structured plan-driven approaches to the highly creative, customer-intimate agile ones.

- Software managers, directors, CTOs, and CIOs
- Project managers and leads
- Measurement and process improvement specialists
- Requirements and business analysts
- Software architects
- Lead developers and software engineers
- Security engineers
- Test and QA managers

NETWORKING WITH COLLEAGUES

- **The EXPO**—Look for answers to your development needs and meet other attendees with the same challenges
- **Open Space**—Discuss a topic in greater depth or examine a topic not on the program
- **Meet the Speakers at Lunch**—Pose your toughest questions to industry experts
- **Bookstore and Speaker Book Signings**—Meet the author of your favorite software book
- **Welcome Reception**—Kick off the Better Software Conference & EXPO with hors d'oeuvres and refreshments, and network with peers and experts in the industry
- **EXPO Reception**—Socialize with others and enjoy complimentary food and beverages
- **Breakfasts, Breaks, Lunches, and More!**

CONFERENCE AT-A-GLANCE

SUNDAY, JUNE 8

8:30	**Software Testing Certification—Certified Tester – Foundation Level Training** (8:30 a.m. - 12:00 p.m.)
12:00	Lunch
1:00	**Software Testing Certification—Certified Tester – Foundation Level Training** (1:00 p.m. - 5:00 p.m.)

MONDAY, JUNE 9

8:30 Tutorial Sessions (8:30 a.m. - 12:00 p.m.)

MONDAY FULL DAY TUTORIALS

MA The Leadership Tutorial: Improving Your Ability to Stand and Deliver
Andy Kaufman, Institute for Leadership Excellence & Development Inc.
MB Principles and Practices of Lean-Agile Development
Alan Shalloway, Net Objectives
MC Essential Software Requirements – *Lee Copeland, Software Quality Engineering*
MD Software Security Fundamentals – *Paco Hope, Cigital*
ME Leading Change through Collaboration
Pollyanna Pixton, Accelinnova; Todd Little, Landmark Graphics Corporation
MF Practical Software Measurement: Objective Information for Decision Makers
Beth Layman, Layman and Layman
MG Test-Driven Development – *Rob Myers, Net Objectives*

MONDAY MORNING TUTORIALS

MH Influence Strategies for Practitioners – *Linda Rising, Independent Consultant*
MI User Story Mapping – *Jeff Patton, Independent Consultant*
MJ Risk-Based Testing: A Systematic Approach – *Julie Gardiner, Grove Consultants*
MK One Iteration at a Time: The Life of an Agile Tester
Lisa Crispin, ePlan Services, Inc.
ML Functional Programming Makes a Comeback – *Chuck Allison, Utah Valley University*
MM Ending Deadly Meetings: The Keys to Success
David Spann, Agile Adaptive Management, Inc.
MN Artful Making for Agile Teams
Stacia Broderick, AgileEvolution, Inc.; Lee Devin, Swarthmore College
MO Software Inspections in the 21st Century
Ed Weller, Integrated Productivity Solutions, LLC
MP Project Risk Management: A Systematic Approach – *Jeff Payne, Independent Consultant*

12:00 Lunch

1:00 Tutorial Sessions (1:00 p.m. - 4:30 p.m.)

MONDAY FULL DAY TUTORIALS (CONTINUED)

MA The Leadership Tutorial: Improving Your Ability to Stand and Deliver
Andy Kaufman, Institute for Leadership Excellence & Development Inc.
MB Principles and Practices of Lean-Agile Development
Alan Shalloway, Net Objectives
MC Essential Software Requirements – *Lee Copeland, Software Quality Engineering*
MD Software Security Fundamentals – *Paco Hope, Cigital*
ME Leading Change through Collaboration
Pollyanna Pixton, Accelinnova; Todd Little, Landmark Graphics Corporation
MF Practical Software Measurement: Objective Information for Decision Makers
Beth Layman, Layman and Layman
MG Test-Driven Development – *Rob Myers, Net Objectives*

MONDAY AFTERNOON TUTORIALS

MQ Fundamentals of Writing Secure Code – *Herbert (Hugh) Thompson, People Security*
MR Test Estimation for Development and Test Managers
Julie Gardiner, Grove Consultants
MS From User Story to User Interface – *Jeff Patton, Independent Consultant*
MT Value Stream Mapping for Software Development – *Bob Hartman, Net Objectives*
MU Design Principles Behind Design Patterns – *Chuck Allison, Utah Valley University*
MV Measuring and Monitoring Process Improvement
David Herron and David Garmus, The David Consulting Group
MW Metrics in Agile Development – *Michael Mah, QSM Associates, Inc.*
MX Understanding the CMMI® – *Will McKnight, Next Level Consultants*
MY Groovy Agile Development—With Groovy – *Andy Glover, Stelligent, Inc.*

5:00 ASTQB Testing Certification Information Session

TUESDAY, JUNE 10

8:30 Tutorial Sessions (8:30 a.m. - 12:00 p.m.)

TUESDAY FULL DAY TUTORIALS

TA Fearless Change: Introducing New Ideas – *Linda Rising, Independent Consultant*
TB Facilitation Skills for Project Leaders – *Jean Tabaka, Rally Software Development*
TC Managing Imposed Deadlines: Risk Management in the Real World
Michael Mah, QSM Associates, Inc.
TD Test Case Development in Agile Development – *Timothy Korson, Korson Consulting*
TE Hands-On Responsibility-Driven Design – *Rebecca Wirfs-Brock, Wirfs-Brock Associates*
TF Finding Ambiguities in Requirements – *Richard Bender, Bender RBT*
TG Twelve Steps to a Successful Metrics Program – *Linda Westfall, The Westfall Team*
TH Agile Requirements Interactive – *Ken Pugh, Net Objectives*

TUESDAY MORNING TUTORIALS

TI Scrum Basics from the Master – *Ken Schwaber, Advanced Development Methods, Inc.*
TJ Finding and Developing Agile Leaders – *David Spann, Agile Adaptive Management, Inc.*
TK Behavior-Driven Development: A Tester's Dream – *Dan North, ThoughtWorks*
TL Fitnesse: Executable Documentation for Agile Testing – *Andy Glover, Stelligent, Inc.*
TM Refactoring Your Wetware: Thinking About Thinking
Andy Hunt, The Pragmatic Programmers
TN Gambling Your Future: Effective Portfolio Management
Todd Little, Landmark Graphics Corporation; Kent McDonald, Knowledge Bridge Partners
TO Leading Successful Projects in Changing Environments – *Pollyanna Pixton, Accelinnova*
TP Software Endgames: Learning to Finish What You've Started
Robert Galen, Robert Galen Consulting Group

12:00 Lunch

1:00 Tutorial Sessions (1:00 p.m. - 4:30 p.m.)

TUESDAY FULL DAY TUTORIALS (CONTINUED)

TA Fearless Change: Introducing New Ideas – *Linda Rising, Independent Consultant*
TB Facilitation Skills for Project Leaders – *Jean Tabaka, Rally Software Development*
TC Managing Imposed Deadlines: Risk Management in the Real World
Michael Mah, QSM Associates, Inc.
TD Test Case Development in Agile Development – *Timothy Korson, Korson Consulting*
TE Hands-On Responsibility-Driven Design – *Rebecca Wirfs-Brock, Wirfs-Brock Associates*
TF Finding Ambiguities in Requirements – *Richard Bender, Bender RBT*
TG Twelve Steps to a Successful Metrics Program – *Linda Westfall, The Westfall Team*
TH Agile Requirements Interactive – *Ken Pugh, Net Objectives*

TUESDAY AFTERNOON TUTORIALS

TQ All Out Scrum: Experiencing a Product Release
Hubert Smits, Rally Software Development
TR Measuring and Using Your Team's Velocity – *Rob Myers, Net Objectives*
TS Releasing Agile Products in the Enterprise
Robert Galen, Robert Galen Consulting Group
TT Writing Good Software Security Requirements – *Paco Hope, Cigital*
TU Quantitative Techniques for Software Management
James McCaffrey, Volt Information Sciences, Inc.
TV Agile Leadership: Inside the Project and From Above – *Pollyanna Pixton, Accelinnova*
TW Expanding Your Discussion Toolkit for Better Communication
Michele Sliger, Sliger Consulting
TX Discovering the Agile Project Manager Inside You
Johanna Rothman, Rothman Consulting Group

4:30 Welcome Reception (4:30 p.m. - 6:00 p.m.)

CONFERENCE AT-A-GLANCE

WEDNESDAY, JUNE 11

8:30	**Opening Remarks —** *Lee Copeland, Program Chair*						
8:45	**That's No Way to Run a Business —** *Ken Schwaber, Advanced Development Methods, Inc.*						
9:45	Morning Break						
	Managing Projects and Teams	Agile Management	Agile Development	Process Improvement	Testing	Requirements	Special Topics
10:00	W1 **What's the Deal with "Best Practices"—Revisiting the Idea** Dan North, Thoughtworks	W2 **Flow, Pull, Innovate: The Secrets to Agile Adoption** Jean Tabaka, Rally Software Development	W3 **Agile in the Non-Agile Enterprise: Hurdling Obstacles** Michele Sliger, Sliger Consulting	W4 **More than the Process Police: CMMI® Process and Product Quality Assurance** Will McKnight, Next Level Consultants	W5 **Lessons Learned in Programmer Testing** James Newkirk, Microsoft	W6 **Beyond User Stories: Managing Requirements by Business Need** Alan Shalloway, Net Objectives	W7 **Agile Death Marches: Fast, Expensive, Deadly** Ken Schwaber, Advanced Development Methods, Inc.
11:30	Lunch in the EXPO *(EXPO open 11:00 a.m. - 3:00 p.m.)*						
12:45	W8 **Bandages or Tombstones? Distinguishing Between Minor Setbacks and Impending Doom** Payson Hall, Catalysis Group, Inc.	W9 **Pragmatic Agility: Principles, Not Dogma** Andy Hunt, The Pragmatic Programmers	W10 **Collaborative Card Play: A Fun Way to Learn Agile Modeling** Jeff Patton, Independent Consultant	W11 **Deception and Estimation: How We Fool Ourselves** Linda Rising, Independent Consultant	W12 **Early Defect Detection for Software Analysis and Design** Vladimir Pavlov, International Software and Productivity Engineering Institute	W13 **Answer the Call: Help Product Owners Define and Prioritize Requirements** Kent McDonald, Knowledge Bridge Partners	W14 **The Give and Take of Design Criticism** Rebecca Wirfs-Brock, Wirfs-Brock Associates
2:15	Networking Break in the EXPO						
2:45	W15 **The Psychology of Software Engineers** James McCaffrey, Volt Information Sciences, Inc.	W16 **Agile Leadership: Coaching Great Teams** Robert Galen, Robert Galen Consulting Group	W17 **Agile Software Testing Strategies** Jared Richardson, 6th Sense Analytics	W18 **Successful Process Improvement—The Agile Way** Nelson Perez, Sierra's Edge, Inc.	W19 **Ten Principles of an Agile Tester** Lisa Crispin, ePlan Services, Inc.	W20 **Who Are Your Project Stakeholders?** Linda Westfall, The Westfall Team	W21 **Eight Steps to a Virtualized Test Environment** John Janakiraman, illumita
4:15	Networking Break in the EXPO *(EXPO open 4:00 p.m. - 6:30 p.m.)*						
4:30	**The Good, the Better, and the Rather Puzzling: The Agile Experience at Five Companies —** *Michael Mah, QSM Associates*						
5:30	Reception in the EXPO, 5:30 p.m. – 6:30 p.m.						

THURSDAY, JUNE 12

8:30	**Opening Remarks —** *Lee Copeland, Program Chair*						
8:35	**Attacking Waste in Software: Three Practices We Must Embrace Now —** *Jean Tabaka, Rally Software Development*						
9:45	Networking Break in the EXPO *(EXPO open 9:30 a.m. - 3:00 p.m.)*						
	Managing Projects and Teams	Personal Development	Agile Development	Metrics	SOA	Security	Special Topics
10:15	T1 **Fifteen Tips for Speeding Up Your Project** Johanna Rothman, Rothman Consulting Group, Inc.	T2 **Your Attention Please: Concentration is a Learnable Skill** Lee Devin, Swarthmore College	T3 **Continuous Integration: The Cornerstone of a Great Shop** Jared Richardson, 6th Sense Analytics	T4 **The Uncertainty Surrounding the Cone of Uncertainty** Todd Little, Landmark Graphics Corporation	T5 **A Toolkit for Assessing SOA Readiness** Jerry Smith, Symphony Services	T6 **Software Security Assessment: The Naked Truth** Herbert (Hugh) Thompson, People Security	T7 **SOX and HIPPA and RESPA, Oh My! Mastering Software Compliance** Elle Ringham, Cognizant Technology Solutions
11:45	Lunch in the EXPO • Meet the Speakers						
1:00	T8 **A Recipe for Self-Induced Project Failure** Mike Seavers, BJC HealthCare	T9 **The Leadership Imperative: Creating a Culture of Trust** Pollyanna Pixton, Accelinnova	T10 **The Impact of Poor Estimating and How to Fix It** Mitch Lacey, Ascentium	T11 **Function Point Analysis: A Quick and Easy Primer** David Garmus & David Herron, The David Consulting Group	T12 **SOA Testing Challenges and Proven Practices** Guruprasad Gopalakrishnan, Wipro Technologies	T13 **Automating Security Testing with cUrl and Perl** Paco Hope, Cigital	T14 **Real Software QA** Linda Westfall, The Westfall Team
2:30	Networking Break in the EXPO *(EXPO open 9:30 a.m. - 3:00 p.m.)*						
3:00	T15 **Maximizing ROI on New Technology Acquisition** Chris Ronak, Divestco, Inc.	T16 **An Alternative to Consensus: Accelerating Effective Decisions** Michele Sliger, Sliger Consulting	T17 **Test-Driven Analysis: Focus on the User, Not the Code** Ken Pugh, Net Objectives	T18 **You Just Don't Understand Me: Interdisciplinary Awareness to the Rescue** Mike Tholfsen, Microsoft	T19 **The Challenges of SOA Security** Jimmy Xu, CGI, Inc.	T20 **Fuzzing: New Tests for Robustness and Security** Ari Takanen, Codenomicon	T21 **Scenario-Based Architecture Reviews: A Quality Process** Kevin Bodie, Pitney Bowes
4:30	Networking Break						
4:45	**Lessons Learned in Project Management —** *Johanna Rothman, Rothman Consulting Group, Inc.*						
5:30	ASTQB Certified Tester Certification Exam						

More than 99.7% of 2007 attendees recommend the Better Software Conference & EXPO to others in the industry.

Meet the Speakers at Lunch

Thursday, June 12, 2008 During Lunch

Meet with industry experts for an open discussion in key areas of software development technology. Pose your toughest questions, address specific project needs, and gather details on the latest research and cutting-edge practices...or just come by to chat.

Bookstore and Speaker Book Signings

During EXPO hours, purchase popular industry books—many authored by Better Software Conference & EXPO speakers—from BreakPoint Books. Authors are available for questions and book signings during session breaks and EXPO hours.

Certification Exam and Information Session

Experienced software testing professionals at the Better Software Conference & EXPO have the opportunity to take the ISTQB™ Certified Tester—Foundation Level exam facilitated by the American Software Testing Qualifications Board (ASTQB). To pre-register for the exam or to download the syllabus, visit www.astqb.org. The public examination will be held on site Thursday, June 12, at 5:30 p.m. Not sure if you are ready? On Monday, June 9, at 5:00 p.m. there will be a free information session to introduce you to the certification program and examination.

Welcome Reception

Tuesday, June 10, 2008, 4:30 p.m.-6:00 p.m.

Kick off the Better Software Conference & EXPO with hors d'oeuvres and beverages. Take this opportunity to network with your peers and experts in the industry.

EXPO Reception

Wednesday, June 11, 2008, 5:30-6:30 p.m.

All attendees are invited to the EXPO reception for complimentary food and beverages.

OPEN SPACE

Wednesday and Thursday, June 11–12 During the EXPO

Want to discuss a topic in greater depth? Examine a topic that is not on the program? Meet with others who are also interested? Great! You're looking for Open Space.

We supply an open space—a room, tables, chairs, and flipcharts... you supply the ideas and the leadership. Choose a topic you'd like to discuss, pick a timeslot, promote your topic at the conference, enroll others, and have an Open Space discussion.

That's what Open Space is all about—you are in charge of your own learning. Everyone who comes to an Open Space session should be passionate about the topic and willing to take some responsibility for creating learning out of that passion.

Key principles of Open Space meetings are:

1. Whoever attends is the right person.
2. Whatever happens is the only thing that could have happened.
3. Whenever it starts is the right time.
4. When it is over, it is over.
5. The Law of Two Feet—If you find yourself in a situation where you aren't learning or contributing, go somewhere else. Don't be passive—create your own conference session on a topic of interest to you. If you've never tried Open Space, do it now. You'll teach and you'll learn.

Software Testing Certification Training

Certified Tester—Foundation Level

Sunday, June 8 – Tuesday, June 10
8:30 a.m. – 5:00 p.m.

Are you looking for internationally recognized certification in software testing?

Delivered by top experts in the testing industry, Software Testing Certification is an accredited training course, designed to help prepare you for the ISTQB™ Certified Tester-Foundation Level exam. This certification program, accredited by the ISTQB™ through its network of National Boards, is the only internationally accepted certification for software testing. The ISTQB™, a non-proprietary and non-profit organization, has granted more than 60,000 certifications in more than twenty countries around the world.

REGISTER EARLY—SPACE IS LIMITED!

At 3:30 p.m., on the third day of the course you will have the opportunity to take the ISTQB™ Certified Tester—Foundation Level exam. The ISTQB™ Certified Tester—Foundation Level certification exam is independently administered by the American Software Testing Qualifications Board. A $250 fee for the exam is included in your course registration. For more information on ISTQB™ certification or to download the syllabus, please visit www.astqb.org.

You save an additional $200 by attending both the Certification Training and the Conference!

To register for the Certification Course and the Conference, contact our Client Support Group at 888-268-8770 or 904-278-0524 or sqeinfo@sqe.com.

THE EXPO June 11–12, 2008

Visit Top Industry Providers Offering the Latest in Software Development Solutions

Looking for answers? Take time to explore this one-of-a-kind EXPO, designed to bring you the latest solutions in technologies, software, and tools covering all aspects of software development. Throughout the EXPO, participate in technical presentations and demonstrations to help you find the tools and services you need to support and improve your software projects. Meet one-on-one with representatives from some of today's most progressive and innovative organizations.

EXPO Hours

Wednesday, June 11
11:00 a.m. – 3:00 p.m.
4:00 p.m. – 6:30 p.m.

Reception: 5:30 p.m. – 6:30 p.m.
All attendees are invited to the EXPO reception for complimentary food and beverages.

Thursday, June 12
9:30 a.m. – 3:00 p.m.

For Sponsor/Exhibitor news and updates, visit **www.sqe.com/bsce**

See page 26 for a preview of sponsors and exhibitors.

CONFERENCE SPEAKERS

K Keynote Speaker | T Tutorial Speaker | C Class Speaker

T
Conference Chair
Lee Copeland
Software Quality Engineering

T C
Alan Shalloway
Net Objectives

T C
David Herron
The David Consulting Group

T C
Jeff Patton
Independent Consultant

T C
Lee Devin
Swarthmore College

C
Payson Hall
Catalysis Group, Inc.

T
Andy Glover
Stelligent, Inc.

T
David Spann
Agile Adaptive Management, Inc.

T
Jeff Payne
Independent Consultant

T C
Linda Rising
Independent Consultant

T C
Pollyanna Pixton
Accelinnova

T C
Andy Hunt
The Pragmatic Programmers

T
Ed Weller
Integrated Productivity Solutions, LLC

C
Jerry Smith
Symphony Services

T C
Linda Westfall
The Westfall Team

T C
Rebecca Wirfs-Brock
Wirfs-Brock Associates

T
Andy Kaufman
Institute for Leadership Excellence & Development Inc.

C
Elle Ringham
Cognizant

C
Jimmy Xu
CGI, Inc.

T C
Lisa Crispin
ePlan Services, Inc.

T
Richard Bender
Bender RBT

C
Ari Takanen
Codenomicon

C
Guruprasad Gopalakrishnan
Wipro Technologies

K T C
Johanna Rothman
Rothman Consulting Group, Inc.

K T
Michael Mah
QSM Associates

T
Rob Myers
Net Objectives

T
Beth Layman
Layman and Layman

T C
Herbert (Hugh) Thompson
People Security

C
John Janakiraman
illumita

T C
Michele Sliger
Sliger Consulting

T C
Robert Galen
Robert Galen Consulting Group

T
Bob Hartman
Net Objectives

T
Hubert Smits
Rally Software Development

T
Julie Gardiner
Grove Consultants

C
Mike Seavers
BJC HealthCare

T
Stacia Broderick
AgileEvolution, Inc.

C
Chris Ronak
Divestco, Inc.

T C
James McCaffrey
Volt Information Sciences, Inc.

T C
Ken Pugh
Net Objectives

C
Mike Tholfsen
Microsoft

T
Tim Korson
Korson Consulting

T
Chuck Allison
Utah Valley University

C
James Newkirk
Microsoft

K T C
Ken Schwaber
Advanced Development Methods, Inc.

C
Mitch Lacey
Ascentium

T C
Todd Little
Landmark Graphics Corporation

T C
Dan North
Thoughtworks

C
Jared Richardson
6th Sense Analytics

T C
Kent McDonald
Knowledge Bridge Partners

C
Nelson Perez
Sierra's Edge, Inc.

C
Vladimir Pavlov
International Software and Productivity Engineering Institute

T C
David Garmus
The David Consulting Group

K T C
Jean Tabaka
Rally Software Development

C
Kevin Bodie
Pitney Bowes

T C
Paco Hope
Cigital

T C
Will McKnight
Next Level Consultants

MONDAY PRE-CONFERENCE TUTORIALS

MONDAY, JUNE 9, 8:30–4:30 (FULL DAY)

FULL DAY MA The Leadership Tutorial: Improving Your Ability to Stand and Deliver

Andy Kaufman, Institute for Leadership Excellence & Development Inc.

In this highly interactive session, Andy Kaufman helps you wrestle with real world leadership issues we all face—influencing without authority, motivating your team, and dealing with conflict. Explore the difference between leadership and management—and why it matters—and get a clear picture of a leader's responsibilities, including the balance between short-term and long-term focus and the need to deliver results while developing organizational capability. Discuss the importance of developing the leadership skills of your team members, including practical ways to do so even with a limited training budget. Andy delves into the importance of one-on-one relationships and delivers proven insights on managing upward, dealing with peers, and developing stronger bonds both inside and outside your organization. Accelerate your ability to influence your organization, your projects, and your career to become the leader your team needs and demands. Walk away with practical tools to help you lead your team, including a template for formalizing a team charter, and a reproducible survey to solicit leadership feedback from bosses, peers, stakeholders, and team members.

Andy Kaufman helps people around the world become better leaders so they can more reliably deliver results while having a life. He is an international speaker and executive coach, and president of the Institute for Leadership Excellence & Development Inc. Andy is a certified Project Management Professional (PMP®) and is the author of *Navigating the Winds of Change: Staying on Course in Business & in Life, How to Organize Your Inbox & Get Rid of E-Mail Clutter,* and *Shining the Light on The Secret.*

FULL DAY MB Principles and Practices of Lean-Agile Development

Alan Shalloway, Net Objectives

As the popularity of agile development spreads, more and more companies are discovering that simply breaking down projects into small iterations is not sufficient. Agile methods require changes in management, analysis, architecture, design, testing, and quality assurance, as well as project management. Given the substantial adjustments required, where can a team or enterprise look for guidance in its transition? Learning the required skill sets individually is fraught with problems—analysis, design, code, and test are not independent; they must be integrated. Join Alan Shalloway as he describes the landscape of skills that a development team needs to become effective agile developers. He discusses a set of principles and practices that integrate the guidance provided by lean, agile methods, design patterns, and more. In particular, Alan details how agile analysis and design patterns support agile methods, and how core "lean" principles support all agile methods, including design and test-driven development.

Alan Shalloway is the founder and CEO of Net Objectives. With more than thirty-five years of experience, Alan is an industry thought leader, trainer, and coach in the areas of lean software development, the lean-agile connection, Scrum, agile architecture and using design patterns in agile environments. He is a popular speaker at prestigious conferences worldwide. Alan is the primary author of *Design Patterns Explained: A New Perspective on Object-Oriented Design* and is currently writing a book on Lean Anti-Patterns.

FULL DAY MC Essential Software Requirements

Lee Copeland, Software Quality Engineering

You deal with software requirements all the time. Whether you are a developer in an agile environment, an analyst who gathers and documents requirements for plan-driven development, a software designer who studies requirements as the basis of your work, a tester who employs or often must discover requirements as the foundation of test cases, or a technical user who describes your needs to development, you need the right approaches and skills to develop and interpret software requirements. Join Lee Copeland to learn how to identify all the important stakeholders of a system and better ways to elicit and capture requirements in different settings: one-on-one interviews, meetings, brainstorming and Joint Application Development (JAD) sessions, buddy checks, inspections, ambiguity reviews, and retrospectives. Discover ways to ferret out the big risks, unknowns, and unresolved conflicts that often doom projects from the start.

With more than thirty years of experience as an information systems professional at commercial and nonprofit organizations, **Lee Copeland** has worked in applications development, software testing, and software process improvement. Lee has developed and taught numerous training courses on software development and testing issues and is a well-known speaker with Software Quality Engineering. The author of the popular reference book, *A Practitioner's Guide to Software Test Design,* Lee presents at software conferences around the world. He is a frequent contributor to StickyMinds.com and managing technical editor for *Better Software* magazine.

FULL DAY MD Software Security Fundamentals

Paco Hope, Cigital

The key to proactive, effective computer system security is getting a risk management handle on the problem of security inside the software. Created by the experts who literally wrote the book on software security, this interactive session encompasses the software security awareness and best practices you need to achieve a secure and trustworthy environment. Everyone involved in software development requires baseline knowledge of software security problems and risks, along with an overall understanding of approaches for producing secure software. Join Paco Hope in this interactive session as he defines the software security problem and then describes a set of software security principles, touch points, and key concepts that can be integrated into any software development lifecycle. Paco describes how and why software is exploited and presents an overview of architectural risk analysis, security testing, and advanced tools for code review. Learn why software security is everyone's job, and take back an overview of your next steps for adopting a comprehensive software security program.

A managing consultant at Cigital, **Paco Hope** has more than twelve years of experience in software and operating system security with areas of expertise in software security policy, code analysis, host security, and PKI. Paco has worked significantly with embedded systems in the gaming and mobile communications industries, and has also served as a subject matter expert on issues of network security standards in the financial industry. Prior to joining Cigital, he served as director of product development for Tovaris, Inc. and head systems administrator in the Department of Computer Science at the University of Virginia. Paco is co-author of *Mastering FreeBSD* and *OpenBSD Security.*

FULL DAY ME Leading Change through Collaboration

Pollyanna Pixton, Accelinnova; Todd Little, Landmark Graphics Corporation

Leaders today are faced with an incredible challenge—delivering the right results to changing marketplaces, doing more with limited resources, improving processes to reduce costs, opening new markets, and keeping the company from falling into chaos. Amazingly though, the solutions to many challenges are already held within your organization and team. How do you unleash the talent within and foster the flow of innovative ideas? In this hands-on and highly interactive session, Pollyanna Pixton and Todd Little introduce the principles of collaboration and the tools you need to create collaborative cultures in your team and organization. Combining principles with practice, you will learn how to use a proven collaboration process to generate new ideas and embrace change, identify barriers to innovation and agility, and discover novel ways to implement solutions. Practice these techniques and tools to become a more collaborative leader while learning the process for leading upwards and outwards.

An international collaborative leadership expert, **Pollyanna Pixton** developed the models for collaboration and collaborative leadership through her thirty-five years of working inside and consulting with corporations and organizations. She helps companies create workplaces where talent and innovation are unleashed—making them more productive, efficient, and profitable. Pollyanna is a founding partner of Accelinnova, president of Evolutionary Systems, director of the Institute for Collaborative Leadership, and co-author of forthcoming book, *Stand Back and Deliver, A Leader's Guide to the Agile Enterprise* due out in November 2008. She co-founded the Agile Project Leadership Network (APLN) and chaired the Agile 2006 Leadership Summits in London and Minneapolis. Contact her at ppixton@accelinnova.com.

Todd Little is a senior development manager for Landmark Graphics Corporation. For more than twenty-five years he has been involved in almost all aspects of software development with a focus on commercial software applications. Todd is on the Board of Directors for the Agile Alliance, a co-author of the Declaration of Interdependence for Agile Project Leadership, and a founding member and past president of the Agile Project Leadership Network (APLN). Todd is a well-known speaker and writer on software engineering topics including business value, uncertainty, complexity, and leadership.

FULL DAY MF Practical Software Measurement: Objective Information for Decision Makers

Beth Layman, Layman and Layman

Today's fast-paced business environments require just-in-time decisions based on the best information available. What initiatives should we fund? Are we getting value from our efforts and investments? Are we getting better over time? Project teams are concerned with their ability to meet budgets and schedules, whether they will be ready to release as planned, and whether customer requirements have been satisfied. Beth Layman explains the concepts of Practical Software Measurement (PSM) to define measurement programs that can improve your decision-making. Beth discusses the role of measurement at all levels of the enterprise and how history, culture, and maturity influence the measurement footprint. She describes how to use an issue-driven measurement approach by defining what to measure, how to collect the data, how to analyze the information, and how to use the results. Beth illustrates this approach through real-world case studies. Take away a practical approach for measuring what's important to your organization and learn ways to avoid the typical measurement roadblocks that plague many organizations.

A successful process improvement consultant, facilitator, teacher, and coach with more than twenty-five years of experience in the high tech sector, **Beth Layman** is an authority on measurement and process improvement. Her wide-ranging experience includes commercial, government, aerospace, and product software organizations. Beth provides training and interactive workshops, assessments, management consulting, and coaching in areas such as process definition, management, and improvement, software and performance measurement, project and portfolio management, and software quality assurance. Beth is an SEI Authorized CMMI® Lead Appraiser and is co-author of *Practical Software Measurement: Objective Information for Decision Makers.*

LAPTOP REQUIRED

FULL DAY MG Test-Driven Development

Rob Myers, Net Objectives

Test-Driven Development (TDD) is a powerful technique for combining software design, testing, and coding to increase reliability and productivity. Rob Myers demonstrates the basic and essential TDD techniques, including unit testing with the common xUnit family of open source development frameworks, refactoring code, and using mock/fake objects in development. Use exercises to practice the techniques. With many years of product development experience using TDD, Rob will address the questions that arise during your own relaxed exploration of the techniques.

Attendees should have strong programming skills and be familiar with an object-oriented language and programming techniques. Each delegate should bring a laptop installed with your favorite programming language and IDE—and come prepared to write code. Rob can provide JUnit for Java and NUnit for any .NET language. For any other language choice (e.g., C++ or Ruby), you will need to install (and verify) your chosen xUnit framework prior to the tutorial.

Rob Myers has more than twenty years of professional experience in software development, including projects for industry leaders in medical, aerospace, and financial services. In the late 1990s, Rob became an eXtreme Programming coach and traveled throughout the country assisting teams with agile software development practices and object-oriented design techniques. Rob brings to the classroom his passion for Lean software development, team development, and sane work environments. He currently teaches Test-Driven Development and Refactoring, Effective .NET, and a new Test-Driven ASP.NET course.

MONDAY PRE-CONFERENCE TUTORIALS

MONDAY, JUNE 9, 8:30–12:00 (MORNING HALF-DAY)

HALF DAY MH Influence Strategies for Practitioners

Linda Rising, Independent Consultant

You've tried and tried to convince people of your position. You've laid out your logical arguments on impressive PowerPoint slides—but you are still not able to sway them. Cognitive scientists understand that the approach you are taking is rarely successful. Often you must speak to others' subconscious motivators rather than their rational, analytic side. Linda Rising shares influence strategies that you can use to more effectively convince others to see things your way. These strategies take advantage of a number of hardwired traits: "liking"—we like people who are like us; "reciprocity"—we repay in kind; "social proof"—we follow the lead of others similar to us; "consistency"—we align ourselves with our previous commitments; "authority"—we defer to authority figures; and "scarcity"—we want more of something when there is less to be had. Learn how to build on these traits as a way of bringing others to your side. Use this valuable toolkit in addition to the logical left-brain techniques on which we depend.

Linda Rising has a Ph.D. from Arizona State University in the field of object-based design metrics and a background that includes university teaching and industry work in telecommunications, avionics, and strategic weapons systems. An internationally known presenter on topics related to patterns, retrospectives, and the change process, Linda is the author of *Design Patterns in Communications, The Pattern Almanac 2000, A Patterns Handbook,* and co-author with Mary Lynn Manns of *Fearless Change: Patterns for Introducing New Ideas*. Find more information about Linda at www.lindarising.org.

HALF DAY MI User Story Mapping

Jeff Patton, Independent Consultant

Is your agile project buried under a mountain of user stories? As you add stories, does your vision of the product you're building grow hazier? As story count increases, do business stakeholders become more frustrated with prioritization? Do you find it difficult to communicate the big picture of what your system does? User story mapping is a simple approach to gathering and organizing user stories. A story map will help you prioritize stories into sensible releases that maximize value by placing emphasis on the users of the software and what they can accomplish when the software is released. In a fun and fast-paced tutorial, Jeff Patton reviews the basics of good agile stories and describes approaches for gathering and combining user stories into a story map. Leverage story maps for planning incremental releases and for breaking down large stories into smaller pieces of work.

For the past twelve years, **Jeff Patton** has designed and developed software on a wide variety of projects from on-line aircraft parts ordering to electronic medical records. A winner of the Agile Alliance's 2007 Gordon Pask Award for contributions to agile development, Jeff has focused on agile approaches since working on an early Extreme Programming team in 2000. He specializes in the application of user centered design techniques to improve agile requirements, planning, and products. Some of Jeff's recent writing on the subject can be found at www.agileproductdesign.com. Jeff's forthcoming book gives tactical advice to those seeking to deliver useful, usable, and valuable software.

HALF DAY MJ Risk-Based Testing: A Systematic Approach

Julie Gardiner, Grove Consultants

Risks are endemic in every phase of every project. One key to project success is to identify, understand, and manage these risks effectively. However, risk management is not the sole domain of the project manager, particularly with regard to product quality. It is here that the effective tester can significantly influence the project outcome. Shortened time scales, particularly in the latter stages of projects, are a frustration with which most of us are familiar. Julie Gardiner explains how risk-based testing can shape the quality of the delivered product in spite of such time constraints. Join Julie as she reveals how you can apply product risk management to a variety of organizational, technology, project, and skills challenges. Receive practical advice—gained through interactive exercises—on how to apply risk management techniques throughout the testing lifecycle, from planning through execution and reporting. Take back a practical process and the tools you need to apply risk analysis to testing in your organization.

With more than eighteen years of experience in the IT industry **Julie Gardiner** has spent time as an analyst programmer, Oracle DBA, and project manager. She has first hand experience as a test analyst, test team leader, test consultant, and test manager. At Grove Consultants Julie provides consultancy and training in all aspects of testing, specializing in risk-based testing, agile testing, test management, and people issues. She is a certified SCRUM master. Julie won best presentation at STAR*EAST* 2007 and 2005; best presentation at BCS SIGiST 2005; and best tutorial at EuroSTAR 2006.

HALF DAY MK One Iteration at a Time: The Life of an Agile Tester

Lisa Crispin, ePlan Services, Inc.

Has your organization recently implemented agile development practices? Or are they considering doing so? As test manager, tester, or someone involved in testing on a daily basis, you may have questions. What do testers do during the first part of an iteration—before anything's ready to test? Where does user acceptance testing fit into an agile release cycle? How can testing possibly keep up with two-week development cycles? During eight years of working on and with a variety of agile teams, Lisa Crispin has determined which practices and skills help agile testers succeed. Learn what testers do during release or "theme" planning when the team determines the work it will do for several upcoming iterations. Follow a tester's activities through the start, middle, and end of one two-week development iteration. Discover the new roles testers must embrace to help ensure a successful release, including the end-game, user acceptance testing, packaging, documentation, and training. In this interactive session, hands-on exercises, real-life examples, and group discussions give you the practical testing-related skills necessary to succeed with agile development.

A tester on agile teams since 2000, **Lisa Crispin** currently works as a tester at ePlan Services Inc., developing Web-based financial applications using XP and Scrum. She leads tutorials and workshops on agile testing at conferences in the US and Europe. Lisa regularly contributes articles about agile testing to publications such as *Better Software* magazine, *IEEE Software*, and *Methods and Tools*. Lisa co-authored *Testing Extreme Programming* with Tip House, and is co-writing *Agile Testing: The Tester Role in Agile Development* with Janet Gregory. For more about Lisa's work, visit her Web sites: http://lisa.crispin.home.att.net and http://www.agiletester.ca.

LAPTOP REQUIRED

HALF DAY ML Functional Programming Makes a Comeback

Chuck Allison, Utah Valley University

Functional Programming (FP), which treats computational algorithms as mathematical functions, is arguably the oldest programming paradigm—it was developed well before computers were invented. With the fast pace of our industry today, you'd think FP would be old news. Ironically, popular programming languages are now rediscovering its power and simplicity. C++ has function objects and adapters; C# has delegates and lambda expressions; Java is adding closures; ML and the new D language have all of these. A few newer, dynamically-typed languages including Python and Ruby have always had FP capabilities. What exactly is functional programming? What is its timeless appeal? How can you use FP to improve your designs and code? In this hands-on workshop, Chuck Allison helps you examine functional programming's constructs and idioms, how they work in today's languages, and how they can increase your programming effectiveness while making your code more expressive and easier to read and understand.

To take full advantage of this hands-on session, bring a laptop installed with the Python (www.python.org) and ML (www.smlnj.org/) languages. Visual Studio (C# and C++) and D (www.digitalmars.com) are optional.

Before becoming a professor of computer science at Utah Valley University, **Chuck Allison** developed software for more than twenty years. He is a contributing editor for *Better Software* magazine and editor of The C++ Source, an online journal. He spent most of the 1990s as an active member of the C++ Standards Committee and is author of *Thinking In C++, Volume 2*, with Bruce Eckel. Chuck offers onsite training in C++, Python, and Design Patterns. Whenever he finds a little down time, Chuck plays classical guitar or bikes the country roads of central Utah. Contact him at chuck@freshsources.com.

HALF DAY MM Ending Deadly Meetings: The Keys to Success

David Spann, Agile Adaptive Management, Inc.

Do you hate meetings? Have you attended a release planning meeting when most of the "right" people were not in attendance? Or when the meeting ends, no one really understands what the next steps are and who is responsible for them? Have you ever been in a meeting where everyone is pointing fingers at everyone else so you have to have yet another meeting to sort out who is right and who is wrong? Are some meetings simply a waste of your time—with everyone leaving more confused than when they arrived? If any of these descriptions fit your situation or if you just want your meetings to be more focused and productive, this tutorial is for you. David Spann presents key practices to help groups define and then focus on their purpose for meeting, debate the merits of possible solutions, and leave with specific actions. Make sure that your next meeting is productive—issues are resolved quickly and participants understand what needs to be done once the meeting is over. Because many software people spend 50%-75% of their working lives in business meetings, it's about time that we actually got something accomplished in them.

A senior management consultant in Park City, Utah, **David Spann** focuses on strategic planning, team building, executive coaching, and training to help organizations become more agile and adaptive. David helped host the first Agile Software Development conference in 2002 and co-hosted the Agile Executive Summit (2003-2005). He teaches the only MBA course on adaptive project management in the US and is a Certified Professional Facilitator (CPF) and an Assessor for the CPF exam. In his spare time David enjoys life—teaching, hiking, singing, and skiing in Park City.

MONDAY PRE-CONFERENCE TUTORIALS

MONDAY, JUNE 9, 8:30–12:00 (MORNING HALF-DAY)

HALF DAY MN — Artful Making for Agile Teams

Stacia Broderick, AgileEvolution, Inc.; Lee Devin, Swarthmore College

The phrase "working together" is based on a team collaboration metaphor. However, Stacia Broderick and Lee Devin found that most teams don't usually collaborate—rather, they consist of modular parts that are steeped in competition and oriented to reward the "stars." Stacia and Lee use a metaphor drawn from theatre art, a form of group work that requires collaboration, encourages interdependency, eschews competition, and emphasizes the project rather than any particular member of the group. Going from simply "working together" to "innovating collaboratively" requires a quantum shift in our thinking about teamwork. Stacia and Lee introduce you to the experience of artful collaboration—an experience that encourages an innovative mindset, which, when practiced in the workplace, results in innovations. This is not a "techniques" workshop—in the kind of work they advocate, there are no quick fixes. However, Stacia and Lee introduce a frame of mind that's necessary if a person or group wants to break out of the box of conventional teamwork.

In 2006 **Stacia Broderick** founded AgileEvolution, Inc., based on the belief that agile practices present a humane, logical way for teams and companies to deliver products. A project manager for fourteen years, the last seven in software development, Stacia was trained and mentored as a ScrumMaster by Ken Schwaber. She is a Certified ScrumMaster Trainer as well as a PMP, a mix that proves invaluable when assisting organizations as they embrace the principles of agile and transition from traditional to modern practices. With Michele Sliger, Stacia is co-authoring *A Software Project Manager's Bridge to Agility.*

Lee Devin taught theatre at the University of Virginia (1962-66), Vassar College (1966-70), and Swarthmore College (1970-2002). In 1975 be became a member of the artistic staff of the People's Light and Theatre, acting, teaching acting, and doing dramaturgy, currently Senior Dramaturg. With Rob Austin of the Harvard Business School, Lee wrote *Artful Making; What Managers Need to Know about How Artists Work*, published in 2003. In 2005 it won LMDA's Elliott Hayes Award for dramaturgy. Lee is at work on writing projects that not only interfere with his trout fishing but also cause him to neglect his grandchildren.

HALF DAY MO — Software Inspections in the 21st Century

Ed Weller, Integrated Productivity Solutions, LLC

Formally developed at IBM in the 1970s, software inspections are still one of the top three items listed as "good things to do" in software development. In today's distributed, global development environment, inspections remain relevant and, more importantly, both cost-effective and feasible. Ed Weller shares his insights into the economics of inspections and how they can positively affect the bottom line. He explains the roles in an inspection and why they are important to success. Learn the steps in the inspections process and the measurements you need to quantify the value of inspections and find areas for improvement. Ed discusses the impact of the global workforce on inspections, and the tools you need to adapt inspections to multiple locations in different time zones. Take back the six critical factors you must consider when implementing inspections or starting an improvement project.

With more forty years of experience in software systems, test, and software process engineering, **Ed Weller** is internationally recognized as an expert in inspections, having successfully initiated inspection programs that have stood the test of time. His primary interest has been in software process and metrics with a focus on improving quality and productivity. Ed is an SEI-Certified SCAMPI High Maturity Lead Appraiser and instructor for the Introduction to the CMMI®. He has delivered numerous presentations and tutorials at conferences around the world. Ed can be contacted via ed.weller@integratedproductivitysolutions.com.

HALF DAY MP — Project Risk Management: A Systematic Approach

Jeff Payne, Independent Consultant

Successfully delivering software projects continues to be a struggle for many software organizations. Studies continue to show that nearly 25% of large-scale software projects are never delivered and that a majority of the projects that are delivered do not meet time, budget, or quality objectives. Jeff Payne lays out the most common causes of software project failure and explains what you can do to identify and mitigate these risks as early as possible in the software lifecycle. The sometimes fatal risks associated with immature technologies, tool introduction, poor software testing, ambiguous development artifacts, inadequate project staff, and failed project management are discussed and examined. Attendees will leave this tutorial with a structured and proven framework for performing project risk analysis that ties risks to specific business consequences. In a case study of a real-world project, you will practice risk mitigation concepts and reinforce your new skills.

Jeff Payne is an independent consultant who helps organizations improve the efficiency and quality of their software development processes. Jeff co-founded Cigital, Inc. and was their CEO between 1992 and 2008 when it became the leader in software security and quality solutions. He is a recognized software expert and speaks to companies nationwide about the business risks of software failure. Jeff is a frequent conference speaker and has testified before Congress on intellectual property rights, cyber-terrorism, and software quality.

MONDAY, JUNE 9, 1:00–4:30 (AFTERNOON HALF-DAY)

HALF DAY MQ — Fundamentals of Writing Secure Code

Herbert (Hugh) Thompson, People Security

Warning! This tutorial contains graphic examples of software failure ... not for the faint of heart. This "no holds barred" session arms you with information you need to create secure software applications. Hugh Thompson begins by examining why and how software fails with respect to security. He then describes the economics of security and why new legislation and standards are increasing the pressure on organizations to produce more secure code. Hugh provides an example-rich tour through the most severe classes of software vulnerabilities and presents techniques for you to avoid and fix these vulnerabilities. Through live exploits he illustrates vulnerabilities followed by a look at the offending code and remediation strategies. Learn the latest trends in attacks against standalone applications, server software, and Web applications. Take back new defensive coding techniques to battle the most common and costly vulnerabilities in software including SQL injection, 2nd order vulnerabilities, buffer overflows, XSS and XSRF weaknesses, common AJAX flaws, SOA implementation blunders, and more. Leave with the knowledge and insight to significantly improve the security of your system's code.

An expert on application security and testing, **Herbert (Hugh) Thompson** is Chief Security Strategist at People Security (www.peoplesecurity.com). He has co-authored several books and more than eighty academic and industrial publications on security. In 2006 he was named one of the "Top 5 Most Influential Thinkers in IT Security" by SC Magazine and was featured (along with Harri Hursti) in "Hacking Democracy," the Emmy-nominated HBO documentary on e-voting vulnerabilities. On AT&T's tech channel (techchannel.att.com), he currently hosts "The Hugh Thompson Show," which features industry luminaries in IT security. Hugh earned his Ph.D. in Applied Mathematics from Florida Institute of Technology where he remains on the graduate faculty.

HALF DAY MR — Test Estimation for Development and Test Managers

Julie Gardiner, Grove Consultants

Test estimation is one of the most difficult software development activities to do well. The primary reason is that testing is not an independent activity and is often plagued by destabilizing dependencies. Julie Gardiner describes common problems in test estimation, how to overcome them, and reveals six powerful ways to estimate test effort. Some estimation techniques are quick but can be challenged easily; others are more detailed and time consuming to use. The estimation methods are FIA (Finger in the Air), Formula or Percentage, Historical, Consensus of Experts, Work Breakdown Structures, and Estimation Models. Julie looks at how we can approach the "set-in-stone deadlines" that are often presented to us and effectively communicate estimates for testing to senior management. Through the use of exercises, gain experience using these techniques. Spreadsheets and utilities will be given out during this session to help testers, test managers, and development manager.

With more than eighteen years of experience in the IT industry **Julie Gardiner** has spent time as an analyst programmer, Oracle DBA, and project manager. She has first hand experience as a test analyst, test team leader, test consultant, and test manager. At Grove Consultants Julie provides consultancy and training in all aspects of testing, specializing in risk-based testing, agile testing, test management, and people issues. She is a certified SCRUM master. Julie won best presentation at STAR*EAST* 2007 and 2005; best presentation at BCS SIGiST 2005; and best tutorial at EuroSTAR 2006.

HALF DAY MS — From User Story to User Interface

Jeff Patton, Independent Consultant

You've chosen to take an agile approach to development. You've written down as a set of user stories what users want for their system. Now, the developers have questions regarding the look and feel of the user interface. How can you quickly, predictably, and with confidence move from user stories to a user interface? Jeff Patton introduces a practical approach for translating user goals and tasks into user interface designs that effectively support users' work. Discover how a user-centered design practitioner moves quickly from user tasks to user interface. Practice taking a set of user stories and transforming them into more tangible actions that users might take in the user interface; then, collaboratively build and test paper prototypes of your proposed user interface. In addition to paper prototyping skills and basic usability testing skills, learn the essential visual design skills that can help improve the appeal of your new user interface.

For the past twelve years, **Jeff Patton** has designed and developed software on a wide variety of projects from on-line aircraft parts ordering to electronic medical records. A winner of the Agile Alliance's 2007 Gordon Pask Award for contributions to agile development, Jeff has focused on agile approaches since working on an early Extreme Programming team in 2000. He specializes in the application of user centered design techniques to improve agile requirements, planning, and products. Some of Jeff's recent writing on the subject can be found at www.agileproductdesign.com. Jeff's forthcoming book gives tactical advice to those seeking to deliver useful, usable, and valuable software.

MONDAY PRE-CONFERENCE TUTORIALS

MONDAY, JUNE 9, 1:00–4:30 (AFTERNOON HALF-DAY)

HALF DAY MT

Value Stream Mapping for Software Development

Bob Hartman, Net Objectives

You can measure efficiency of a process by calculating the "value-add" time and dividing it by the total time to come up with a percentage. When software development organizations are measured end-to-end in this way, their overall efficiency is almost always lower than 20%. Value stream mapping is a way to identify the impediments in the end-to-end process and improve overall efficiency. In the software world, it is not sufficient to simply apply a methodology such as SCRUM to teams and assume everything will work more efficiently. To achieve the best results, you need to create, analyze, and improve the value stream maps of each sub-process. Only then can you ensure that the practices your teams use will generate maximum business value in the most efficient way possible. Join Bob Hartman to find out how to employ value stream maps and take away the information you need to rapidly improve the efficiency of your software development process.

Bob Hartman is a senior trainer/coach for Net Objectives in the areas of lean-agile processes and testing. He has more than thirty years of experience developing software and is frequently invited to speak about project management practices and agile development. Since starting with agile processes in 2000, Bob's passion has been to help software development companies change in ways that allow them to quickly deliver products that have extremely high quality and exceed customer expectations.

HALF DAY MU

Design Principles Behind Design Patterns

Chuck Allison, Utah Valley University

The introduction of Design Patterns has revolutionized software development. Sadly, most developers are only familiar with a selection of the twenty-three patterns found in the groundbreaking book, *Design Patterns: Elements of Reusable Object-Oriented Software*. These particular patterns are not sacrosanct—many patterns are used in software development, and new patterns are continually identified. While some developers may seek to employ patterns "just because," whether they need them or not, what really matters is mastering the principles behind the patterns. For decades, industry and academia alike have sought an effective vehicle for teaching sound software design principles, and nothing has rivaled design patterns in getting the job done. Join Chuck Allison as he examines important design patterns and shows how they resolve design problems by appealing to enduring principles. Chuck also examines patterns other than design patterns to better understand the pattern concept in general.

Before becoming a professor of computer science at Utah Valley University, **Chuck Allison** developed software for more than twenty years. He is a contributing editor for *Better Software* magazine and editor of The C++ Source, an online journal. He spent most of the 1990s as an active member of the C++ Standards Committee and is author of *Thinking In C++, Volume 2*, with Bruce Eckel. Chuck offers onsite training in C++, Python, and Design Patterns. Whenever he finds a little down time, Chuck plays classical guitar or bikes the country roads of central Utah. Contact him at chuck@freshsources.com.

HALF DAY MV

Measuring and Monitoring Process Improvement

David Herron and David Garmus, The David Consulting Group

Software organizations know that it can take months or even years of investment to achieve significant process improvement results. The goal, of course, is to realize a positive return on that investment and attain a development organization that is more productive and delivers higher quality software. How can the organizational leadership be sure that process improvement is paying off? David Herron and David Garmus outline the quantitative and qualitative measures necessary for an organization to determine its improvement progress. They describe a practical and effective measurement process that permits an organization to protect its investment and ensure that it is on the path to improved productivity and quality. Learn valuable performance modeling techniques that you can use to forecast performance improvement and ways to dynamically monitor progress against your goals for improvement.

David Garmus is a founder of The David Consulting Group (an SEI CMMI® Approved Transition Partner), and supports software development organizations in achieving software excellence with a metric-centered approach. David is an acknowledged authority in the sizing, measurement, and estimation of software application development. He is a past president of the International Function Point Users Group (IFPUG) and a member of their Counting Practices Committee. David has spoken at numerous conferences and written many articles and several books.

David Herron is an acknowledged authority in the use of metrics to monitor the impact of Information Technology (IT) on the business and on the advancement of IT organizations to higher levels of software process maturity. He is a noted author and lecturer and has addressed audiences throughout the US and Europe on performance measurement, software process improvement, and outsourcing governance. With David Garmus, David Herron has co-authored two books on functional measurement. David Herron's current engagements include senior level consulting and coaching on matters relating to organizational change management, team and individual mentoring.

LAPTOP REQUIRED

HALF DAY MW

Metrics in Agile Development

Michael Mah, QSM Associates, Inc.

When implementing agile methods in your organization, how do you compare the productivity and quality you achieve versus traditional waterfall projects? Join Michael Mah to learn about both agile and waterfall metrics and how the metrics behave. Use your own data to move from guesses on a project whiteboard to realistic agile project trends on productivity, time-to-market, and defects. With real-world case studies, you will get an inside look at agile measurement by seeing metrics in action. In hands-on exercises, learn how to replicate these techniques to make your own comparisons on time, cost, and quality. Working in pairs, you will use templates to calculate productivity metrics. Leverage these new methods to make the case for changing to more agile practices at your company. Take back new ways for communicating to key decision makers the value of implementing agile development practices.

To take full advantage of this session, each participant should bring a laptop computer for metrics capture and productivity calculations.

Michael Mah is director of the Benchmarking Practice, an author with the Cutter Consortium, and managing partner of QSM Associates Inc., specializing in software measurement and project estimation. Michael has written extensively and consulted with the world's leading software organizations while collecting data on thousands of projects worldwide. Michael's book-in-progress, *Optimal Friction*, examines the dynamics of teams under time pressure and its role in contributing to success and failure. He lives in the mountains of western Massachusetts with his two young children. Michael can be reached at www.qsma.com.

HALF DAY MX

Understanding the CMMI®

Will McKnight, Next Level Consultants

A common misconception is that if you have run a development project, then you can run a process improvement project—not true. Understanding the requirements of the CMMI® model is critical, and there is more to understanding CMMI® than simply reading a book. You must be able to interpret what the model identifies as "required" and translate that into how your organization will define a process that makes sense for the way you develop your products. Will McKnight introduces the Capability Maturity Model Integration (CMMI®) for Development (version 1.2) fundamental concepts. He discusses not only the Process Areas defined in CMMI but also the Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) to help you be prepared when your organization is ready to obtain that coveted maturity or capability level designation. Although CMMI® does not directly cover the important process of organizational change management, Will describes how you can leverage practices defined in the model to foster positive organizational change.

CMMI® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

Will McKnight is an experienced process improvement specialist, who has worked on CMM®/CMMI®-based improvement programs in multinational settings with a wide range of organization sizes, styles, and types of software. He has more than twenty years of experience in all phases of the software development life cycle. Will's specialization in product development and management provides him with a deep, "hands-on" understanding of what it takes to provide practical guidance to organizations working to improve their processes. As an SEI-authorized Lead Assessor for CMMI he has performed numerous appraisals.

LAPTOP REQUIRED

HALF DAY MY

Groovy Agile Development—With Groovy

Andy Glover, Stelligent, Inc.

Groovy is a dynamic language that builds upon the strengths of Java and includes additional features inspired by languages such as Python, Ruby, and Smalltalk. With an extremely fast learning curve, Groovy makes available to Java developers the most modern programming features including closures and meta-programming. Andy Glover presents the basics of the Groovy language—how to define objects, use Groovy's native constructs, interface with normal Java objects, and how Groovy adds functionality to normal Java objects through the GDK. Then, he describes Groovy Builders to help you rapidly create XML documents and GUIs. Andy explains GroovySQL to permit easy database access and Groovlets, slick servlets for quickly building Web applications. Learn the unit testing techniques that will facilitate incorporating Groovy into your daily regimen and get you on your way to groovy agile development.

To take full advantage of this hands-on session, each delegate should bring a laptop with the Java 1.5/IDE installed.

An established expert in automated testing frameworks and tools, **Andy Glover** is a published author for multiple online publications including IBM's DeveloperWorks, InfoQ, and O'Reilly's ONJava, Dev2Dev, and ONLamp portals. A frequent speaker at conferences around the country, Andy is the co-author of *Java Testing Patterns, Groovy in Action*, and *Continuous Integration: Improving Software Quality and Reducing Risk*. Andy is passionate about accelerating software development and blogs frequently about agile practices and techniques at thediscoblog.com, testearly.com, and integratebutton.com.

TUESDAY PRE-CONFERENCE TUTORIALS

TUESDAY, JUNE 10, 8:30–4:30 (FULL DAY)

FULL DAY TA Fearless Change: Introducing New Ideas

Linda Rising, Independent Consultant

Those who attend conferences or read books and articles discover new ideas they want to bring into their organizations—but they often struggle when trying to implement those changes. Unfortunately, those introducing change are not always welcomed with open arms. Linda Rising offers proven change management strategies to help you become a more successful agent of change in your organization. Learn how to plant effective seeds of change, and what forces in your organization drive or block change. In addition to using these approaches to change your organization, you can use them to become a more effective person. Come and discuss your organizational and personal change challenges. Linda shows how the lessons from her book, *Fearless Change: Patterns for Introducing New Ideas,* can help you succeed. Learn how to overcome adversity to change and to celebrate your improvement successes along with your organization's new found practices.

Linda Rising has a Ph.D. from Arizona State University in the field of object-based design metrics and a background that includes university teaching and industry work in telecommunications, avionics, and strategic weapons systems. An internationally known presenter on topics related to patterns, retrospectives, and the change process, Linda is the author of *Design Patterns in Communications, The Pattern Almanac 2000, A Patterns Handbook,* and co-author with Mary Lynn Manns of *Fearless Change: Patterns for Introducing New Ideas*. Find more information about Linda at www.lindarising.org.

FULL DAY TB Facilitation Skills for Project Leaders

Jean Tabaka, Rally Software Development

This "on-your-feet" tutorial guides project managers, agile coaches, and Scrum Masters in how to apply facilitation techniques and tools to support collaborative decision making. These practices are critical for agile planning, daily interaction, and reviews of agile software development projects and teams. Jean Tabaka shows why agile teams require a collaborative style of decision making rather than classic command-and-control approaches. Practice planning for agile meetings and kicking off those meetings to ensure that the attendees are truly engaged and results-oriented. Find out about tools to help teams gather the important insights and wisdom necessary to attain the sustainable agreements in their agile projects. Learn ways to deal with conflict that occurs when many opinions and recommendations arise and help teams inspect and adapt their agile processes collaboratively. Along the way, you will discover what must change within your organization to successfully apply collaboration, especially with large and distributed agile teams.

Jean Tabaka is an agile mentor and coach with Rally Software Development. In addition to being a Certified Scrum Trainer and Practitioner, she is also a Certified Professional Facilitator. Her unique blend of passions and skills has been applied in a variety of organizations—large and small, co-located and distributed—eager to adopt the best of agile and bring out the best in their teams. Author of the Agile Software Development Series book *Collaboration Explained*, Jean holds a Masters in Computer Science from Johns Hopkins University. When not sharing her agile passion with clients, she resides in beautiful Boulder, Colorado.

LAPTOP REQUIRED

FULL DAY TC Managing Imposed Deadlines: Risk Management in the Real World

Michael Mah, QSM Associates, Inc.

Frequently, schedules and deadlines all too often are dictated to software development teams. When this happens, what is a manager to do? Michael Mah addresses the key issues in deadline-driven projects—estimation and risk management. Employing industry data from more than 7,000 completed projects worldwide, Michael describes how different software projects—agile development, waterfall development, and package implementations—behave in unique and interesting ways when a deadline is imposed. Using case studies from leading companies, Michael illustrates how to estimate and commit to a reasonable project scope in the face of aggressive deadlines. Find out how to "triage" the amount of functionality you can deliver within an imposed deadline and deal methodically with the inevitable project trade-offs. Develop a core set of estimation metrics that will help you avoid common scheduling traps.

To take full advantage of this tutorial, each participant should bring a laptop computer for data capture and estimation calculations.

Michael Mah is director of the Benchmarking Practice, an author with the Cutter Consortium, and managing partner of QSM Associates Inc., specializing in software measurement and project estimation. Michael has written extensively and consulted with the world's leading software organizations while collecting data on thousands of projects worldwide. Michael's book-in-progress, *Optimal Friction,* examines the dynamics of teams under time pressure and its role in contributing to success and failure. He lives in the mountains of western Massachusetts with his two young children. Michael can be reached at www.qsma.com.

FULL DAY TD Test Case Development in Agile Development

Timothy Korson, Korson Consulting

"Pure" agile development uses story cards to scope and organize customer needs. Each story is described in a sentence or two with details filled in through conversations. Because there are no written requirements that contain enough information for independent test teams to create comprehensive test suites, testers find themselves in a difficult position. In some agile philosophies, testers must create test cases directly from discussions with clients. In effect, the test cases become the only detailed requirements. Eliciting test requirements directly from stakeholders requires that testers learn a new set of skills and practices. In addition to explaining how to effectively create system test cases from stories and stakeholders, Tim Korson examines unit, component, increment, and regression test development as parts of a comprehensive testing process within an agile development environment. Tim presents test automation strategies and tools that agile testers are successfully using today.

Tim Korson has a decade of experience working on a large variety of systems developed using modern software engineering techniques. This experience includes distributed, real time, and embedded systems as well as business information systems in an n-tier, client-server environment. Tim's typical involvement on a project is as a senior management consultant with additional technical responsibilities to ensure high quality, robust test and quality assurance processes and practices. He has authored numerous articles and co-authored the book *Object Technology Centers of Excellence.*

FULL DAY TE Hands-On Responsibility-Driven Design

Rebecca Wirfs-Brock, Wirfs-Brock Associates

Objects are more than simple bundles of logic and data—they are service-providers, information-holders, coordinators, controllers, and interfacers to other systems. Rebecca Wirfs-Brock discusses how objects play specific roles and occupy well-known positions in an application's architecture. Each object must know and do its part! Role stereotypes—think of them as purposeful oversimplifications—are a fundamental way of seeing objects' responsibilities. Learn and practice practical responsibility-driven design techniques to enhance your design process and design thinking. Experience the latest in Class Responsibility Collaborator (CRC) modeling, object identification and naming, object role stereotypes, control style design, collaboration trust regions, and contracts. Find out how responsibility-driven design thinking can enhance your design and development practices.

Delegates should be familiar with object-oriented technology and object concepts. Some experience with object design and programming languages is a plus.

Rebecca Wirfs-Brock, design columnist for *IEEE Software*, is a well-known object practitioner who invented the way of thinking about objects known as responsibility-driven design. Through her writing, teaching, consulting, and speaking, Rebecca popularizes the use of informal techniques and practical thinking tools for designers, architects, and analysts. She teaches courses on responsibility-driven design, practical UML, developing and communicating software architecture, and agile design skills. Rebecca regularly mentors teams on use case writing, design, architecture, and managing incremental, iterative object-technology projects. Rebecca is the author of *Object Design: Roles, Responsibilities, and Collaboration.*

FULL DAY TF Finding Ambiguities in Requirements

Richard Bender, Bender RBT

In this process-oriented class—geared to business analysts, designers, programmers, testers, technical writers, and users—Richard Bender teaches a powerful and practical method for ensuring that requirements specifications are clear, concise, and unambiguous. Learn how to verify that requirements are written at the correct level of detail needed by designers, developers, and testers. Because this level of detail must be discovered one way or another, this process does not add any additional overhead to the effort and costs of developing requirements specifications. In fact, by eliminating ambiguous requirements early in development, you can save time, reduce confusion, and avoid unnecessary re-work. In this hands-on workshop, learn the ambiguity review process and how to quickly identify ambiguities in specifications in any format. Eliminate unnecessary complexity from your requirements documents and help your team develop and test applications more quickly and more effectively.

Richard Bender has more than thirty-five years of experience in software with a primary focus on quality assurance and testing. He has consulted internationally for large and small corporations, government agencies, and the military on applications that run the gamut from finance, billing, and manufacturing to medical, transportation, and communications—to prison management and weather forecasting. Richard teaches a series of courses on the techniques for practical, rigorous requirements-based testing, code-based testing, and writing testable requirements.

TUESDAY PRE-CONFERENCE TUTORIALS

TUESDAY, JUNE 10, 8:30–4:30 (FULL DAY)

FULL DAY TG

Twelve Steps to a Successful Metrics Program

Linda Westfall, The Westfall Team

Linda Westfall offers a practical process for establishing and tailoring a software metrics program that focuses on business goals and information needs. Learn a practical, start-to-finish method of selecting, designing, and implementing software metrics. Linda outlines a "cookbook method" you can use to simplify the journey from conceptual software measurement and metrics to valuable information summarized and delivered to management. Utilize the Goal/Question/Metric paradigm to select metrics that align with the organizational, project, and process goals. Walk through the steps for designing important metrics—standardizing entity and attribute definitions, choosing measurement functions, establishing measurement methods, defining decision criteria, designing reporting mechanisms, and determining additional qualifiers. Find out who should collect the data, what data to collect, and how to collect it. Learn to consider the human issues of implementing a measurement system and the metric do's and don'ts that Linda has discovered over many years of helping people with their metrics programs.

Linda Westfall is the president of The Westfall Team, which provides software engineering, quality and project management consulting, and training services. Prior to starting her own company, Linda was senior manager of quality metrics and analysis at DSC Communications, where her team designed and implemented a corporate-wide metrics program. An ASQ Certified Software Quality Engineer, Linda has more than thirty years of experience in real-time software engineering, quality, and metrics. A past chair of the ASQ Software Division, Linda Westfall has served as the Software Division's Program Chair and Certification Chair and on the ASQ National Certification Board.

FULL DAY TH

Agile Requirements Interactive

Ken Pugh, Net Objectives

All projects, whether agile or traditional, need requirements. Ken Pugh explores the differences between agile and traditional requirements by interactively creating a set of agile-style requirements. These requirements are developed through progressive elaboration—rather than the big-bang, big-document approach. Ken first examines how stakeholders and requirements gatherers interact and communicate in an agile environment. Students will create a charter for a project that defines the overall scope and participate in a story-gathering workshop to create an initial set of stories. Learn when and how to revise stories by chunking and de-chunking to ensure that the requirements fulfill the characteristics of good stories. Explore user roles, personas, and narratives to determine additional stories. Practice prioritizing the requirements and estimating their business value to help in that prioritization. At the end of the session students will begin constructing use cases and acceptance tests to add details to the requirements.

A fellow consultant with Net Objectives, **Ken Pugh** has more than one-third of a century of experience in software development—from gathering requirements for stock market analysis to testing real-time radar systems. Ken consults, trains, mentors, and testifies on agile processes and technology topics ranging from object-oriented design and test-driven development to Linux/Unix and the system development process. He has written several programming books, including the Jolt Award winner, *Prefactoring*, and has served clients from London to Sydney. When not computing, Ken enjoys snowboarding, windsurfing, biking, and hiking the Appalachian Trail. Ken can be reached at ken.pugh@netobjectives.com.

TUESDAY, JUNE 10, 8:30–12:00 (MORNING HALF-DAY)

HALF DAY TI

Scrum Basics from the Master

Ken Schwaber, Advanced Development Methods, Inc.

Scrum is a project and product management process targeted at complex software product development. Scrum is characterized by a product backlog of prioritized work that is completed in a series of short iterations or "sprints." It includes a brief, daily meeting or "scrum," at which the team discusses progress, upcoming work, and any impediments. Before an iteration, Scrum teams have a sprint planning session to discuss the backlog items for the sprint, and after the iteration Scrum teams have a sprint retrospective to reflect about the past sprint. Using Scrum, many organizations and programs have gained significantly higher development productivity, higher product return on investment, and increased product quality. Ken Schwaber discusses the underlying concepts on which Scrum is built and why these concepts facilitate productivity and quality improvements. Ken describes how to use "time boxes" for delivering working software and explains the roles and rules of Scrum that instantiate the theory. Join Ken to look at what your project and organization must do to take advantage of the capabilities Scrum offers.

The co-developer of the Scrum process (with Jeff Sutherland), **Ken Schwaber** is a signatory of the Agile Manifesto and founder of the Agile Alliance and Scrum Alliance. He spends his time helping others figure out how to get the benefits of agile development and determine if agile practices are appropriate for them. Ken has been in software and product development for more than thirty years—in positions ranging from bottle-washer to boss. He lives in Lexington, Massachusetts.

HALF DAY TJ

Finding and Developing Agile Leaders

David Spann, Agile Adaptive Management, Inc.

You've been asked to write a job description and announcement for a lead position on your agile team. Although you know that the person must be familiar with the technology and agile methodologies, you are struggling to define the behavioral characteristics of the "right" person for the job. How do you find and develop leaders for your agile development team? What behaviors make one person a success and another an out-and-out failure? David Spann presents the eight key attributes—the top three: strategic, consensual, and empathetic—he discovered in a role expectations survey of agile practitioners and consultants. Help yourself, your team, and your organization understand these behaviors and enhance the search for people to fill agile leader roles. Use these same behavioral traits and proven staff development techniques to help grow your existing team.

A senior management consultant in Park City, Utah, **David Spann** focuses on strategic planning, team building, executive coaching, and training to help organizations become more agile and adaptive. David helped host the first Agile Software Development conference in 2002 and co-hosted the Agile Executive Summit (2003-2005). He teaches the only MBA course on adaptive project management in the US and is a Certified Professional Facilitator (CPF) and an Assessor for the CPF exam. In his spare time David enjoys life—teaching, hiking, singing, and skiing in Park City.

HALF DAY TK

Behavior-Driven Development: A Tester's Dream

Dan North, ThoughtWorks

Behavior-driven development (BDD) is a new evolution in agile software delivery. With its roots in test-driven development, domain-driven design, and automated acceptance testing, BDD enables teams to deliver valuable software more quickly. It puts the tester at the heart of the delivery process rather than in the usual position at the end, squashed against the deadline. Dan North introduces the principles behind BDD and demonstrates how it works in practice by looking at the roles and interactions within a development team. Learn to capture requirements and deliver working, tested software that will wow the customer. See a live demonstration of BDD showing how the various roles in a BDD team work together to deliver the right software. Anyone involved in getting software delivered—analysts, developers, and testers—will benefit from this session.

Dan North has been writing software for more than fifteen years and is a principal consultant with ThoughtWorks. He spends his time helping teams become more effective at delivering software and presents at conferences such as JAOO, Agile, and OOPSLA on topics ranging from learning theory to development methodologies. He has published articles in the *Java Developers' Journal, Better Software* magazine, *CIO* newsletters, and the *DSDM consortium.*

LAPTOP REQUIRED

HALF DAY TL

Fitnesse: Executable Documentation for Agile Testing

Andy Glover, Stelligent, Inc.

Although agile development practices push for reducing repetitive documentation and embracing change, achieving these goals is by no means easy. Fitnesse is an open source tool that facilitates collaboration by bridging the gap between those who define requirements—customers—and those who turn requirements into code—developers. With Fitnesse, collaborative teams develop stories in a specific format that is then implemented as tests through a framework that marries with the underlying application. A Fitnesse test suite enables rapid change and helps produce agreement among agile team members in short order. Join Andy Glover to learn how to embrace collaboration and change by defining Fitnesse tests that exercise an application end-to-end. Practice defining specific Fitnesse structures, plugging them into real code, and running them in an automated fashion. Experience first-hand how non-coders can define tests easily and how this collaboration yields working software faster.

To take full advantage of this hands-on session, each delegate should bring a laptop with the Java 1.5/IDE installed.

An established expert in automated testing frameworks and tools, **Andy Glover** is a published author for multiple online publications including IBM's DeveloperWorks, InfoQ, and O'Reilly's ONJava, Dev2Dev, and ONLamp portals. A frequent speaker at conferences around the country, Andy is the co-author of *Java Testing Patterns, Groovy in Action*, and *Continuous Integration: Improving Software Quality and Reducing Risk*. Andy is passionate about accelerating software development and blogs frequently about agile practices and techniques at thediscoblog.com, testearly.com, and integratebutton.com.

TUESDAY PRE-CONFERENCE TUTORIALS

TUESDAY, JUNE 10, 8:30–12:00 (MORNING HALF-DAY)

HALF DAY TM Refactoring Your Wetware: Thinking About Thinking

Andy Hunt, The Pragmatic Programmers

Software development happens in your head—not in an editor, IDE, or design tool. We're well educated on how to work with software and hardware, but what about wetware—our brains? Join Andy Hunt for a look at how the brain really works (hint: it's a dual processor, shared bus design) and how to use the best tool for the job by learning to think differently about thinking. Andy looks at the importance of context and the role of expert intuition in software development. Learn to take advantage of pole-bridging and integration thinking. Compare different laterally-specialized functions, including synthesis vs. analysis and sequential processing vs. pattern-matching. Discover the one simple habit that separates the geniuses from the "wanna-bes." Andy helps you discover how to learn more deliberately by managing your knowledge portfolio. Explore practical learning techniques including mind maps, reading techniques, situational feedback, and how best to cope with the torrent of new information that assaults each of us.

In the industry since the early 1980s, **Andy Hunt** is one of the seventeen founders of the Agile Alliance, which launched the Agile Manifesto and the agile movement. Andy is a programmer, consultant, author, publisher, and co-founder of the Pragmatic Bookshelf. He co-authored the best-selling book *The Pragmatic Programmer* and five others, including the recent award-winning *Practices of an Agile Developer*. At conferences and private corporations throughout the US and Europe, Andy is a frequent speaker on topics ranging from software development to management and cognition. When not working, Andy is an active musician composing, recording, and playing trumpet, flugelhorn, and piano.

HALF DAY TN Gambling Your Future: Effective Portfolio Management

Todd Little, Landmark Graphics Corporation; Kent McDonald, Knowledge Bridge Partners

Yogi Berra is quoted as saying, "It is difficult to make predictions, especially about the future." Effective management of a software portfolio is a challenge that many companies ignore, avoid, or fail to follow through with because it is too difficult. In this hands-on session, Todd Little and Kent McDonald run a simulation of an online gambling company's software portfolio. Todd and Kent provide an overview of some basic product and portfolio management guidelines and then introduce the simulation game in which participants make decisions about what investments the company should make in its software. Through the instruction and the simulation, learn about product, project, and portfolio management issues including business strategy, investment return, constraint management, technical and market uncertainty, and project complexity. Find out what it takes to optimize overall return on your software investments.

Todd Little is a senior development manager for Landmark Graphics Corporation. For more than twenty-five years he has been involved in almost all aspects of software development with a focus on commercial software applications. Todd is on the Board of Directors for the Agile Alliance, a co-author of the *Declaration of Interdependence for Agile Project Leadership*, and a founding member and past president of the Agile Project Leadership Network (APLN). Todd is a well-known speaker and writer on software engineering topics including business value, uncertainty, complexity, and leadership.

A business systems coach with more than a decade of experience, **Kent McDonald** has successfully guided projects and designed business solutions in the financial services, health insurance, performance marketing, human services, non-profit, and automotive industries. His background includes delivering data-intensive and Web-enabled application development projects that provide outstanding business value. He has coached client staff to help teams reach project goals more productively and effectively. Kent is a sought after speaker, writer, and coach on project leadership, business analysis, and delivering business value through projects. He is the current president of the APLN.

HALF DAY TO Leading Successful Projects in Changing Environments

Pollyanna Pixton, Accelinnova

There's no doubt about it—agile has gone mainstream. Short delivery iterations give organizations the means to incorporate change safely, reach go/no-go decisions early, and discover realistic team velocities. Managers can better determine if market windows can be reached—thus placing successful products in customers' hands. What if the ground beneath the project team is changing rapidly even as it is trying to make progress? Pollyanna Pixton shares a collaboration model and iterative delivery process that will help you succeed, even in unstable conditions. She shares her ideas on creating an open environment, identifying the talent the team needs, managing risks, and creating team ownership to ensure great results. Among the skills you need are a collaborative, transparent leadership style; an approach to positively influence outcomes; collaborative communication—and then the knowledge of when to stand back and let things happen. Leave this session with some keys to successfully lead agile project teams—even in the midst of chaos.

An international collaborative leadership expert, **Pollyanna Pixton** developed the models for collaboration and collaborative leadership through her thirty-five years of working inside and consulting with corporations and organizations. She helps companies create workplaces where talent and innovation are unleashed—making them more productive, efficient, and profitable. Pollyanna is a founding partner of Accelinnova, president of Evolutionary Systems, director of the Institute for Collaborative Leadership, and co-author of forthcoming book, *Stand Back and Deliver, A Leader's Guide to the Agile Enterprise* due out in November 2008. She co-founded the Agile Project Leadership Network (APLN) and chaired the Agile 2006 Leadership Summits in London and Minneapolis. Contact her at ppixton@accelinnova.com.

HALF DAY TP Software Endgames: Learning to Finish What You've Started

Robert Galen, Robert Galen Consulting Group

Nothing feels worse than when your team works their hearts out on a project only to have it fail to meet the customer's needs and quality targets at the end of the project. So much focus is typically put on the beginning of a project that we fail to realize how important ending well can be. Bob Galen shares tools and techniques he's used to successfully deliver on the promises of his projects. There's no magic involved. Instead, Bob explores how to plan an iterative model for testing in your endgame; create dynamic release criteria and connect them to your requirements and to the reality of the project; manage change control in agile and non-agile environments; handle defects; winnow down change via several code freeze models; and finally, define core metrics for guiding your project towards release. Software endgames are also focused toward your team. Bob wraps-up the session with a set of powerful patterns that help you engage your teams within the endgame scenario.

The director of Product Development and Agile Architect for ChannelAdvisor, **Bob Galen** has held director, manager, and contributor level positions in both software development and quality assurance organizations. He is a Certified Scrum Master Practicing (CSP), Certified Scrum Product Owner (CSPO), and an active member of the Agile Alliance and Scrum Alliance. Bob authored *Software Endgames—Eliminating Defects, Controlling Change, and the Countdown to On-Time Delivery*. Bob may be reached at bob@rgalen.com or at www.rgalen.com.

TUESDAY, JUNE 10, 1:00–4:30 (AFTERNOON HALF-DAY)

HALF DAY TQ All Out Scrum: Experiencing a Product Release

Hubert Smits, Rally Software Development

Hubert Smits has created a large-scale project for this session in which students will use agile methods to create a plan. By working together in small teams, you learn the planning process for large agile projects, experience real-life examples, and apply your new knowledge immediately. As a starting point for the exercise, Hubert provides a description of a product that you are to develop. Work in your group to develop a program strategy for the new product. Create the product vision, the product roadmap, the backlog of product features, and the release plan. This exercise simulates the experience of working in a low-tech, high-collaboration style and allows you to experience the impact of agile practices on you and your team.

Hubert Smits is an agile coach, working for Rally Software Development in Boulder, Colorado. In this role he travels the world to support organizations in the implementation of agile methods. He works with teams to train them during the implementation cycle, facilitates planning meetings, and coaches executive teams in the management of the new approach to software development. A Certified ScrumMaster and Scrum Trainer, Hubert has authored papers on Scrum implementations ("The CIO Playbook of Implementing Scrum" with Ken Schwaber) and planning in agile projects ("Five Levels of Agile Planning").

HALF DAY TR Measuring and Using Your Team's Velocity

Rob Myers, Net Objectives

"Velocity" is an oft-misunderstood agile term. Developers worry they're being evaluated based on this number. Managers want to know how it can be increased. The team's definition of velocity—explicit or implicit—affects the way the team estimates stories, plans iterations, and tracks progress. The definition of velocity must be consistent and agreed upon; otherwise, planning efforts quickly unravel. Using a monetary metaphor, Rob Myers illustrates how to use velocity in iteration planning. In this simulation you'll experience a non-technical agile iteration planning session that concretely demonstrates how velocity works. Learn about estimation techniques such as "planning poker" and try out this valuable, rapid-estimation technique based on story-complexity. See how Big Visible Charts reveal the team's progress through an iteration or release and discuss what to do about vacations, meetings, sick days, and surprises. Rob discusses the "Four Variables" of software development, and what to do when the answer to "Are we on schedule?" is "No."

Rob Myers has more than twenty years of professional experience in software development, including projects for industry leaders in medical, aerospace, and financial services. In the late 1990s, Rob became an eXtreme Programming coach and traveled throughout the country assisting teams with agile software development practices and object-oriented design techniques. Rob brings to the classroom his passion for Lean software development, team development, and sane work environments. He currently teaches Test-Driven Development and Refactoring, Effective .NET, and a new Test-Driven ASP.NET course.

TUESDAY PRE-CONFERENCE TUTORIALS

TUESDAY, JUNE 10, 1:00–4:30 (AFTERNOON HALF-DAY)

HALF DAY TS Releasing Agile Products in the Enterprise

Robert Galen, Robert Galen Consulting Group

Agile methods bring wonderful dynamics to software projects—focus on the team, quality-driven development, business value connected by customer engagement, and embracing change—leading toward vastly improved project performance. However, most agile projects are developed within a wider enterprise context that is still waterfall-bound. For the product to be released successfully, you must deal with many other factors. Bob Galen shares his "enterprise extensions" for agile releases including methods for integrating agile teams within a more traditional PMO structure. He discusses iteration models for extending agile testing across the enterprise in regulated and other heavyweight testing environments. See examples of "agile release train" planning dynamics when integrating releases across multiple agile teams. Learn how to develop iteration release criteria and metrics that drive improved quality and visibility throughout your enterprise. Take away new tools and techniques for making agility work within your enterprise and ensuring that your agile products' successful release.

The director of Product Development and Agile Architect for ChannelAdvisor, **Bob Galen** has held director, manager, and contributor level positions in both software development and quality assurance organizations. He is a Certified Scrum Master Practicing (CSP), Certified Scrum Product Owner (CSPO), and an active member of the Agile Alliance and Scrum Alliance. Bob authored *Software Endgames—Eliminating Defects, Controlling Change, and the Countdown to On-Time Delivery*. Bob may be reached at bob@rgalen.com or at www.rgalen.com.

HALF DAY TT Writing Good Software Security Requirements

Paco Hope, Cigital

Unfortunately, security is an afterthought for most software development projects. As with any aspect of high quality software, you achieve the best results in security when you consider it early in the lifecycle—when you establish the customer and business requirements. What are good security requirements and how do you write them clearly and in a way that is testable? Paco Hope explains the differences between standard functional requirements and security requirements and describes what to look for when developing security requirements for your application systems. Find out the different techniques you can use to generate and capture robust security requirements in mission-critical applications—abuse cases, misuse cases, and anti-requirements. See examples of written security requirements to learn the characteristics that make them good—or bad. In a mocked-up system you will practice writing security requirements to augment existing functional requirements.

A managing consultant at Cigital, **Paco Hope** has more than twelve years of experience in software and operating system security with areas of expertise in software security policy, code analysis, host security, and PKI. Paco has worked significantly with embedded systems in the gaming and mobile communications industries, and has also served as a subject matter expert on issues of network security standards in the financial industry. Prior to joining Cigital, he served as director of product development for Tovaris, Inc. and head systems administrator in the Department of Computer Science at the University of Virginia. Paco is co-author of *Mastering FreeBSD* and *OpenBSD Security.*

HALF DAY TU Quantitative Techniques for Software Management

James McCaffrey, Volt Information Sciences, Inc.

A growing trend in managing software development projects is the use of lightweight metric-based techniques. These techniques are easy to compute (using only a calculator), easy to understand, and often associated with agile software development. In this hands-on session, James McCaffrey describes how and when to use these simple but powerful quantitative methods to improve software projects of any type, size, or complexity. James discusses the Minimax Regret and Expected Value criteria for decision making in the face of uncertainty. He helps you understand and interpret critical path metrics and related concepts including Early Start, Fast Tracking, Crashing, and Float Time. Learn to use a Risk Analysis Matrix to prioritize software quality assurance activities and improve your estimating using the Beta distribution statistic. Practice calculating and interpreting some basic financial and scheduling metrics you can put to use immediately.

James McCaffrey manages technical training for software engineers working at Microsoft's campus in Redmond, Washington. He has worked on several Microsoft products including Internet Explorer and MSN Search. James is the author of *.NET Test Automation Recipes* and is a contributing editor of Microsoft's MSDN Magazine. He holds a doctorate in Research Methodology from the University of Southern California and an MS in Information Systems from Hawaii Pacific University. James can be reached at jmccaffrey@volt.com.

HALF DAY TV Agile Leadership: Inside the Project and From Above

Pollyanna Pixton, Accelinnova

Tired of your senior leaders not "getting" what you, as the project manager, are doing on your agile projects? As a PM, how can you convince your organization's senior leaders that you need help or more resources? As a senior leader, do you sometimes wonder what your PMs are really doing on their projects? And, if they need your help, how would you know? Pollyanna Pixton offers practical techniques that PM and senior leaders can use to eliminate these disconnects. Take away new strategies that leaders and PMs can employ to support and help one another successfully—without rescuing each other. Among other vexing issues, Pollyanna addresses what to communicate, what to expect from each other, how to read progress, and how to get stalled projects moving. Put your new skills to work immediately when you get back to the office.

An international collaborative leadership expert, **Pollyanna Pixton** developed the models for collaboration and collaborative leadership through her thirty-five years of working inside and consulting with corporations and organizations. She helps companies create workplaces where talent and innovation are unleashed—making them more productive, efficient, and profitable. Pollyanna is a founding partner of Accelinnova, president of Evolutionary Systems, director of the Institute for Collaborative Leadership, and co-author of forthcoming book, *Stand Back and Deliver, A Leader's Guide to the Agile Enterprise* due out in November 2008. She co-founded the Agile Project Leadership Network (APLN) and chaired the Agile 2006 Leadership Summits in London and Minneapolis. Contact her at ppixton@accelinnova.com.

HALF DAY TW Expanding Your Discussion Toolkit for Better Communication

Michele Sliger, Sliger Consulting

Saying the right thing at the right time can be difficult. Many of us find ourselves wishing we had some magic phrases that would make difficult conversations easier and more productive. Build your toolkit full of just the right phrases, the perfect questions, and the best ways to start, guide, and end discussions. Whether you're a manager, an engineer, or a tester, you can benefit from learning new and more open ways to exchange ideas—after all, to get better software we have to have better communication. Join Michelle Sliger for this interactive session, where you'll have ample opportunity to try these phrases in simulations with your colleagues in an environment that's safe for learning and experimentation. Find out how to stop a pontificator, keep meetings on track, disagree without shutting down the discussion, deliver feedback, say "no" politely yet firmly, and encourage further dialog. Discover how to pack your discussion toolkit with what you need to keep conversations going in the right direction.

For the past eight years—of her more than twenty years in software development—**Michele Sliger** has been embracing change with agile methodologies. Co-author of the forthcoming book *The Software Project Manager's Bridge to Agility* and a self-described "bridge builder," her passion lies in helping those in traditional software development environments cross the bridge to agility. Michele consults to businesses ranging from small start-ups to Fortune 500 companies, helping teams with their agile adoption and organizations with the changes that agile adoption brings. A regular contributor to StickyMinds.com, Michele is a certified Project Management Professional (PMP) and a Certified Scrum Trainer (CST). She can be reached at michele@sligerconsulting.com.

HALF DAY TX Discovering the Agile Project Manager Inside You

Johanna Rothman, Rothman Consulting Group

If you've been managing projects for a while, you may not understand how an agile project works or your role in it. If you're accustomed to predicting the schedule, assigning the work, and tracking a Gantt chart, you may be puzzled by how to use empirical data to know the project's progress, having people self-assign work, and your new role as coach and facilitator. Without a Gantt chart, you may be perplexed and not know how to answer your management's question, "When will you be done?" Agile projects provide the project manager (and any other manager) more useful information than a serial-lifecycle project. Yet, it's difficult for many project managers to make the transition to agile because they don't know what they can or should do. In this experiential tutorial, Johanna Rothman uses a small problem to practice working on an agile project. Practice collecting the data—both quantitative and qualitative—that tells you how the project and the team are progressing. Learn how to assess the project's true state and be able to tell management when you will be done.

Johanna Rothman consults, speaks, and writes on managing high-technology product development. She assists managers, teams, and organizations to become more effective. Johanna has helped engineering organizations, IT organizations, and startups hire technical people, manage projects, and release successful products faster. Johanna is the author of *Manage It! Your Guide to Modern Pragmatic Project Management* and *Hiring the Best Knowledge Workers, Techies & Nerds: The Secrets and Science of Hiring Technical People*, and coauthor with Esther Derby of the pragmatic *Behind Closed Doors, Secrets of Great Management*. Johanna is a host and session leader at the Amplifying Your Effectiveness (AYE) conference.

KEYNOTES

WEDNESDAY, JUNE 11, 8:45 a.m.

That's No Way to Run a Business

Ken Schwaber, Advanced Development Methods, Inc.

Every enterprise sets a baseline plan and reports against it frequently—budget vs. actual financial reports, for example. All parts of the enterprise—sales, manufacturing, service, and finance—provide transparent reports against a plan and then adjust their plans based on actual results and changing circumstances. In software development, project and product managers haven't had credible answers for senior management to these BIG questions: How is this project going? When can we starting migrating to the new system? How do you know? Do you have the capacity to take a new business opportunity? What can you slow down or de-commit? What will be the cost? Now there is an answer that is working in hundreds of development organizations around the world—Scrum. Scrum's iterative, incremental development of just-in-time orders of functionality, coupled with rigorous quality standards, gives those of us in software development answers to these questions. Ken Schwaber shows how Scrum enables you to manage software development projects, programs, and development organizations at least as precisely and accurately as every other part of the business.

Ken Schwaber

The co-developer of the Scrum process (with Jeff Sutherland), Ken Schwaber is a signatory of the Agile Manifesto and founder of the Agile Alliance and Scrum Alliance. He spends his time helping others figure out how to get the benefits of agile development and determine if agile practices are appropriate for them. Ken has been in software and product development for more than thirty years—in positions ranging from bottle-washer to boss. He lives in Lexington, Massachusetts.

WEDNESDAY, JUNE 11, 4:30 p.m.

The Good, the Better, and the Rather Puzzling: The Agile Experience at Five Companies

Michael Mah, QSM Associates

Strategic software development is happening every day—and failures continue to plague us. Unquestionably, a major paradigm shift is underway with the movement to agile methods. But are they really working? With results drawn from industry statistics, Michael Mah answers vital questions about the effectiveness of agile methodologies—XP, Scrum, TDD, pair programming, etc. One discovery underway is that agile methods could be turning the "law of software physics" upside down. For decades, there have been predictable relationships among schedule pressure, staff ramp-up, and bug rates; now, industry data tells us that all this could be changing with agile. Join Michael Mah for a revealing discussion of productivity findings at five—all ostensibly agile—companies, and how they produced a range of results for time-to-market, productivity, and quality. Michael addresses questions such as: Which agile practices are right for your environment? What are the characteristics of a successful agile deployment? How do you measure success or failure of agile process change? Find out how the five case study companies "did" agile in their own ways and how their metrics reveal insights into new agile approaches that are fast becoming mainstream.

Michael Mah

Michael Mah is director of the Benchmarking Practice, an author with the Cutter Consortium, and managing partner of QSM Associates Inc., specializing in software measurement and project estimation. Michael has written extensively and consulted with the world's leading software organizations while collecting data on thousands of projects worldwide. Michael's book-in-progress, *Optimal Friction*, examines the dynamics of teams under time pressure and its role in contributing to success and failure. He lives in the mountains of western Massachusetts with his two young children. Michael can be reached at www.qsma.com.

FUN VEGAS FACTS

- **When Paul Anka first played Vegas, he was too young to enter the casino.**
- **Bugsy Siegel named his casino the Flamingo after the long legs of his showgirl girlfriend.**
- **In Nevada it is mandatory that video slot machines pay a minimum of 75 percent on average.**
- **Vegas Vic, the enormous neon cowboy that towers over Fremont Street, is the world's largest mechanical neon sign.**
- **Howard Hughes stayed at the Desert Inn for so long that he was asked to leave. He bought the hotel.**
- **Camels were used as pack animals in Nevada as late as 1870.**
- **More than 38.9 million people visit Las Vegas each year.**
- **Seventeen of the twenty largest hotels in the U.S. are in Las Vegas.**

THURSDAY, JUNE 12, 8:35 a.m.

Attacking Waste in Software: Three Practices We Must Embrace Now

Jean Tabaka, Rally Software Development

One of the seven principles of Lean Thinking is "eliminate waste." Eliminating waste means minimizing the cost of the resources we use to deliver software to our stakeholders. Jean Tabaka proposes three pivotal practices that we must embrace to aggressively attack waste in software delivery—Software as a Service (SaaS), Community, and Fast Feature Throughput. SaaS eliminates waste by deploying software-based services without the cost inherent in traditional software delivery—materials, shipping, time delay, and more. Community involves stakeholders working together to create products rather than competing among themselves for limited resources. Community eliminates waste by democratizing software development to obviate the need for multiple systems with the same functionality. Fast Feature Throughput refers to development methods that embrace change and quickly deliver value to customers. It eliminates waste by responding to market pull with short, incremental delivery cycles. When IT and all software organizations embrace these practices, they will eliminate waste within their organizations, reduce the waste that consumes our entire industry, and ultimately support the broad 21st century global mandate to manage our scarce resources.

Jean Tabaka

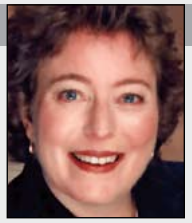
Jean Tabaka is an agile mentor and coach with Rally Software Development. In addition to being a Certified Scrum Trainer and Practitioner, she is also a Certified Professional Facilitator. Her unique blend of passions and skills has been applied in a variety of organizations—large and small, co-located and distributed—eager to adopt the best of agile and bring out the best in their teams. Author of the Agile Software Development Series book *Collaboration Explained*, Jean holds a Masters in Computer Science from Johns Hopkins University. When not sharing her agile passion with clients, she resides in beautiful Boulder, Colorado.

THURSDAY, JUNE 21, 4:45 p.m.

Lessons Learned in Project Management

Johanna Rothman, Rothman Consulting Group, Inc.

You've managed projects but they're never easy. They don't fit into the nice definitions found in project management books. Your schedules are generally off. There are always unkind surprises. Although you're not failing, you feel you could be more successful. There is a solution. Based on her many years of consulting with large and small software teams, Johanna Rothman coaches leaders to take a more pragmatic approach. Employ mini-projects and iterations to explore alternatives technologies. Use incremental steps to finish features one-at-a-time when you don't know how far along you are. Make sure stakeholders agree on what "done" really means. Learn how to escape the dreaded trap of "multi-tasking," a management style that drains energy from everyone whenever there is a task switch. One final secret every project manager must discover: There is no "one right way" to manage a project. Everything depends on context—the company and its products, the technology employed, the people on your team, and you. If you can learn to keep everything in balance, you will have a successful project. Let something get out of whack and you can kiss all your hard work goodbye.



Johanna Rothman

Johanna Rothman helps managers define and solve problems. She assists managers, teams, and organizations to become more effective. Johanna has helped engineering organizations, IT organizations, and startups hire technical people, manage projects, and release successful products faster. Johanna is the author of *Manage It! Your Guide to Modern Pragmatic Project Management* and *Hiring the Best Knowledge Workers, Techies & Nerds: The Secrets and Science of Hiring Technical People*, and coauthor with Esther Derby of the pragmatic *Behind Closed Doors, Secrets of Great Management*. Johanna is a host and session leader at the Amplifying Your Effectiveness (AYE) conference.

CONFERENCE BONUS! One-Year Digital Subscription to *Better Software* magazine!

Better Software Conference & EXPO attendees receive a one-year digital subscription (ten issues) to *Better Software* magazine—the only magazine delivering relevant, timely information so you can tackle the challenges of building better quality software, regardless of your role in the software development lifecycle. www.BetterSoftware.com

If you are a current subscriber, your subscription will be extended for an additional ten digital issues.

WEDNESDAY, JUNE 11, 10:00

W1 MANAGING PROJECTS AND TEAMS

What's the Deal with "Best Practices"—Revisiting the Idea

Dan North, Thoughtworks

We talk about "best practices" as though they exist—an ideal way to manage a team, develop software, and test applications. All we have to do is discover what best practices are. At best, this is naïve, and at worst it's an irresponsible way to approach anything, especially software development. Learning theory—specifically the Dreyfus model of skills acquisition—provides the missing context for practices in general and best practices in particular. Dan North describes how people really learn and acquire skills and helps you discover where and how to use the ideas offered by best practices. See how the arbitrary imposition of best practices is inherently risky and can have a detrimental effect on productivity and morale. Dan explains why the term "best practices" is flawed and suggests more useful ways of sharing experience and evolving what we do.

W2 AGILE MANAGEMENT

Flow, Pull, Innovate: The Secrets to Agile Adoption

Jean Tabaka, Rally Software Development

Jean Tabaka provides straightforward guidance on how teams can begin their agile journey and learn to mature and scale into more and more discipline. The five-step approach emphasizes a path based on the principles of Lean Thinking—Flow, Pull, and Innovate. Each of the five steps outlines specific practices for growth as well as pitfalls and roadblocks to navigate and avoid. Step 1: The team learns to work in a continuous flow. Step 2: The team matures by pulling ready items from the backlog. Step 3: A group of teams adopts and scales up the individual team practices. Step 4: The scaling continues to cover multiple projects. Step 5: The practices are adopted throughout the entire organization. You can apply the disciplines discussed in this class to a single co-located team, a team of teams, or an entire organization eager to take advantage of both agile and lean approaches. Join Jean and learn to achieve the greatest innovations with a much lower risk of failure.

W3 AGILE DEVELOPMENT

Agile in the Non-Agile Enterprise: Hurdling Obstacles

Michele Sliger, Sliger Consulting

Agile is entering the mainstream as a software development practice and leading wider organizational change in many companies. However, in large organizations, it's not practical just to "flip a switch" and have your entire software department "go agile" all at once. In that situation, agile and non-agile teams must work together during the transition. Agile teams must continue to interface with their company's business processes, while management must streamline traditional processes and activities. Agile teams face many obstacles in their quest for cooperative development—resistance to change; differing culture and value systems; changes to measurement, evaluation, and reward systems; and new contracting terms. Join Michele Sliger as she explains how to clear these and other common hurdles facing agile teams working in a traditional organization. Michele discusses the organizational issues that you must address as part of an enterprise-wide agile rollout.

W4 PROCESS IMPROVEMENT

More than the Process Police: CMMI® Process and Product Quality Assurance

Will McKnight, Next Level Consultants

For organizations to succeed in process improvement efforts, they must determine whether newly introduced processes are, in fact, being adopted by managers and practitioners. The Capability Maturity Model Integrated (CMMI®) identifies this verification activity as Process and Product Quality Assurance (PPQA). If you think PPQA is simply "process police," you're not getting all that you should out of your CMMI® practices. Done right, PPQA can be a driving agent for change in your organization. Unfortunately, all too often PPQA ends up little more than a post-mortem review of what was done wrong. That approach, which offers little opportunity to change behavior, not only lowers the value of the process but also hampers change management efforts. Will McKnight demonstrates the potential of an efficient PPQA process. Take back a full functional PPQA process to help transform your process police into valuable, pro-active change agents.

W5 TESTING

Lessons Learned in Programmer Testing

James Newkirk, Microsoft

It has been more than six years since the first release of NUnit 2.0, an open source unit testing tool. In that time literally millions of tests have been written using the tool. Many of these tests have become and continue to be invaluable resources for their teams. Unfortunately, many other NUnit-based tests have not been maintained and are now viewed as having been a waste of effort from the beginning. What separates tests that are used, maintained, and highly valued from tests that are quickly discarded? James Newkirk describes seven key ideas that are proven to increase the readability of NUnit tests and make them much easier to maintain. Learn about the impact of test fixture size and dependency injection on unit testing. James demonstrates how to use the attributes [ExpectedException], [Setup], and [TearDown] to make tests more readable. Incorporating these and the other lessons can make the difference between tests that become a burden to the team and tests that become practical, growing resources.

W6 REQUIREMENTS

Beyond User Stories: Managing Requirements by Business Need

Alan Shalloway, Net Objectives

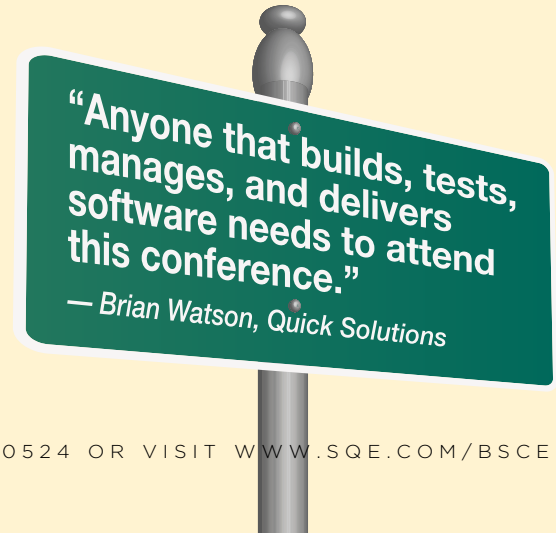
The use of stories in agile projects is commonplace. However, teams in many organizations have discovered limitations in the user story's narrow view in complex projects. Attempts to coordinate related stories through "epics" and "themes" may help the details of managing the problem but generally leave the enterprise view unaddressed—particularly when multiple teams are working together. From his experiences on large agile projects, Alan Shalloway found that combining small pieces together to get a bigger view does not work as well as starting with the bigger view and segmenting it. With agile methods, you must go beyond stories and start with what is known as the "Minimally Releasable Feature" (MRF). The MRF creates the bigger picture of what constitutes business value and enables the management of small stories within this bigger picture. Thus you get the best of both worlds—the efficiency of agile methods aligned with the needs of the enterprise. Alan helps you expand the typical use of stories to keep the bigger business needs in mind while building the smaller pieces that the stories describe.

W7 SPECIAL TOPICS

Agile Death Marches: Fast, Expensive, Deadly

Ken Schwaber, Advanced Development Methods, Inc.

Traditional projects often culminate in death marches—when the developers ruin their lives while cutting quality in a vain attempt to hit dates. With the advent of agile processes, some teams have adopted this deadly practice in almost every software release iteration in which they participate, turning what was a periodic professional hazard into a complete disaster. The consequences of death marches can be seen throughout the industry as the ability of an organization to compete lessens and the consequences come home to roost. Join Ken Schwaber to look at the cause of the iterative and incremental death marches that agile practices "gone wrong" can engender. Ken considers the consequences for agile teams and the industries they serve. Learn to identity and defeat death marches before they start and prevent a death spiral in your current or next project.



W8 MANAGING PROJECTS AND TEAMS

Bandages or Tombstones? Distinguishing Between Minor Setbacks and Impending Doom

Payson Hall, Catalysis Group, Inc.

Are the challenges confronting your project normal and treatable setbacks or signs of something more serious? Can we treat them with a Band-Aid® and a kiss? Should we call the ambulance? The undertaker? Payson Hall shares patterns he's observed while consulting on dozens of large software development and systems integration projects—executive sponsors distancing themselves from your project, ebbing morale, aggressive schedules, and more. Although good project teams react to adversity and try to get the job done in spite of troubles, their adaptive behavior can lead to a loss of perspective. Sometimes, teams become desensitized to the warning signs of degrading project health and are slow to respond to significant issues. Learn the symptoms of project problems and regain perspective as you identify the causes and find the remedies.

W9 AGILE MANAGEMENT

Pragmatic Agility: Principles, Not Dogma

Andy Hunt, The Pragmatic Programmers

You've got questions. Andy Hunt has answers. What is agile software development all about? Why the sudden popularity of agile? Why is it fundamentally different from other approaches? Will it work for my organization and me? Join Andy, one of the seventeen original signatories of the Agile Manifesto and a founder of the Agile Alliance, for his pragmatic take on the answers to these and your other questions. Look at the foundations of agile software development and see what problems agility seeks to address. Don't be distracted by dogma as you take some time to explore the core aspects of agile development. Andy presents a brief overview of the major agile methodologies and walks you through a typical day in the life of an agile developer. Find out what's really important about the agile approach and take back new ideas to help you transition to agile while avoiding common stumbling blocks. Join Andy to find out how to make agility work for *you.*

W10 AGILE DEVELOPMENT

Collaborative Card Play: A Fun Way to Learn Agile Modeling

Jeff Patton, Independent Consultant

While sophisticated modeling tools may produce high quality drawings, it takes a different set of tools and skills to produce high quality discussions that result in a useful model of your software. Using index cards, sticky notes, and markers, it's relatively easy to create simple, agile models that improve communication and foster common understanding of what you are trying to represent. Join Jeff Patton to learn the essentials of card modeling and avoid those monotonous, unproductive meetings. In hands-on exercises, you will work with others to elicit, distill, prioritize, and communicate new information. Learn to employ a variety of simple models and walk away with ideas for using collaborative card play within your project team and with cross-functional teams.

W11 PROCESS IMPROVEMENT

Deception and Estimation: How We Fool Ourselves

Linda Rising, Independent Consultant

Cognitive scientists tell us that we are hardwired for deception. It seems we are overly optimistic, and, in fact, we wouldn't have survived without this trait. With this built-in bias as a starting point, it's almost impossible for us to estimate accurately. That doesn't mean all is lost. We must simply accept that our estimates are best guesses and continually re-evaluate as we go, which is, of course, the agile approach to managing change. Linda Rising has been part of many plan-driven development projects where sincere, honest people with integrity wanted to make the best estimates possible and used many "scientific" approaches to make it happen—all for naught. Re-estimation was regarded as an admission of failure to do the best up-front estimate and resulted in a lot of overhead and meetings to try to "get it right." Offering examples from ordinary life—especially from the way people eat and drink—Linda demonstrates how hard it is for us to see our poor estimating skills and helps us learn to avoid the self-deception that is hardwired in all of us.

W12 TESTING

Early Defect Detection for Software Analysis and Design

Vladimir Pavlov, International Software and Productivity Engineering Institute

For large software development projects, the most important decisions—and the most expensive mistakes—are made at the beginning of the project. At the same time, the initial quality assurance activity is minimal but grows as development moves forward. This results in costly rework (often hidden) in the later stages of the project. Vladimir Pavlov explains how to reduce delays between bug insertions and bug fixes by allocating quality activities over the entire project in proportion to the importance of potential errors. Vladimir describes practical techniques to discover and fix critical analysis and design mistakes almost immediately after their introduction—not in the late phases where they are the most expensive to resolve. He also explains how to integrate these techniques into software development lifecycles including Rational Unified Process, Open Unified Process, and Microsoft Solutions Framework. To increase quality and lower total project costs, establish early defect detection procedures for your projects.

W13 REQUIREMENTS

Answer the Call: Help Product Owners Define and Prioritize Requirements

Kent McDonald, Knowledge Bridge Partners

Numerous software development methodologies are available to provide project teams excellent guidance on how to build systems right. But how do we know that we are building the right systems? We often ask product owners to define and prioritize their requirements—without offering them a great deal of guidance on how to do so. Understanding what the software needs to do and the value that it will add to the organization will help them decide the importance of each requirement. Kent McDonald explains how you can employ a value model based on the project's purpose, costs, benefits, considerations, and its relation to the organization's overall strategies to help product owners define and quantify the value delivered by a project. He will also show how you can use a regular reevaluation of this value model to decide what requirements should be completed and in what order. More importantly, you can empower product owners to determine what requirements should be changed or dropped as the project proceeds.

W14 SPECIAL TOPICS

The Give and Take of Design Criticism

Rebecca Wirfs-Brock, Wirfs-Brock Associates

Have you ever engaged in a design discussion where people didn't play fair? Do you have trouble giving advice that sticks or accepting criticism of your own work? Do you know when you should take up an argument and when is it better to let things slide? Every software engineer needs skills at giving, absorbing, and reacting appropriately to criticism. We should know when to pick our battles and how to spot and counteract faulty reasoning. We should be able to give advice so that others get it, and if they don't, determine why. Join Rebecca Wirfs-Brock to explore how design teams can engage in more effective conversations while eliciting and exchanging constructive criticism. Rebecca surveys the biases that underlie reactions people commonly have to new information and how to overcome those biases. Practice techniques for organizing and presenting constructive criticism as you learn to recognize different types of criticism and the appropriate responses.

Red Rocks Canyon, Nevada

WEDNESDAY, JUNE 11, 2:45

W15 MANAGING PROJECTS AND TEAMS

The Psychology of Software Engineers

James McCaffrey, Volt Information Sciences, Inc.

The personality traits of software engineers tend to be quite different from those of the general population. In recent years, psychologists have come to a nearly unanimous consensus on the number and nature of human personality dimensions. A recent large-scale study involving several hundred software engineers and "regular" people (non-engineers) revealed that the personalities of developers, testers, and managers tend to be different from each other and from the personalities of the general population as a whole. So, how can you use this information? Although administering a personality assessment as part of a hiring process may be legal, it is problematic at best. A much better use of a personality assessment is to gauge the profile of your existing team members to maximize their productivity. Join James McCaffrey as he describes how you can quickly and easily create, administer, and interpret a personality profile of your team. At the conclusion of the session, you will have the opportunity to take the personality assessment used in the study and see how your personality compares with other software professionals.

W16 AGILE MANAGEMENT

Agile Leadership: Coaching Great Teams

Robert Galen, Robert Galen Consulting Group

When adopting agile methods, many project managers find it difficult to move from a traditional, control-based model to a servant leader-based model. This paradigm challenges managers to their core because agility demands a coaching-driven mindset rather than the classic "I'm-in-charge" view. Explore the core aspects of agile leadership and team coaching with Bob Galen as you look at leadership from an agile perspective. Bob discusses "coaching up" the team as part of an agile adoption strategy and offers a conversation framework you can immediately use at work—and at home. Learn the fundamental coaching patterns and anti-patterns as you find out when to step in to help and when to be patient. You'll have the opportunity to practice a conversation or two and hone your new coaching skills.

W17 AGILE DEVELOPMENT

Agile Software Testing Strategies

Jared Richardson, 6th Sense Analytics

Test automation is like exercise. We know both are great ideas, but most of us don't do enough of either. Although we know that creating a solid automated test suite is critical to any agile testing strategy, we are often just told to "Do it" without much support—money or people. Jared Richardson examines the infrastructure and tools needed for your automated testing to succeed and prosper. Jared examines three strategies—test-driven development, defect-driven testing, and blitzkrieg testing—you can use to ensure great test coverage on your projects. You'll gain an understanding of how to leverage your testing investments by employing continuous integration practices in your development projects. With real life scenarios as a backdrop, Jared discusses appropriate testing strategies for your current project or the next one down the road. Jared will get you moving toward automated testing, whether you're starting fresh or trying to clean up an existing project.

W18 PROCESS IMPROVEMENT

Successful Process Improvement—The Agile Way

Nelson Perez, Sierra's Edge, Inc.

Using agile techniques to develop and implement new processes—whether for use in agile environments or not—will increase stakeholder involvement and buy-in, lower cultural resistance, reduce process development cycle time, and encourage continuous process improvement. Join Nelson Perez as he explains how to translate the core principles of the Agile Manifesto into a context that you can apply to any process development and improvement program. Use the Agile Manifesto values and principles to speed up your process improvement initiatives and ensure its success. Based on his experience in a company with a highly resistive culture, Nelson realized that process improvement approaches must be tailored to each situation—what works consistently in one organizational culture may not be useful in another culture down the street, across town, or in another country. The agile paradigm works in process improvement programs because it is a universal approach for encouraging adaptive change. Learn new ways to encourage continuous process improvement and build stronger teams within your development group and throughout the enterprise.

W19 TESTING

Ten Principles of an Agile Tester

Lisa Crispin, ePlan Services, Inc.

Everyone on an agile team does testing. If that's true, what's so special about an agile tester? If I define myself as a tester on an agile team, what does that really mean? Do agile testers need skill sets different from testers on traditional teams? What guides agile testers in their daily activities? Lisa Crispin believes that when it comes to agile testers, skills are important—but attitude is everything. The best agile testers have a results-oriented, customer-focused, collaborative, and creative mindset that makes them successful in an agile development environment. Lisa explains how you can apply ten agile principles to add value on agile teams, or on any software development team for that matter. The ten principles of an agile tester include areas such as feedback, communication, simplicity, continuous improvement, and responding to change. At the end of this session, you'll have gained some practical advice for your own self-improvement process.

W20 REQUIREMENTS

Who Are Your Project Stakeholders?

Linda Westfall, The Westfall Team

It's easy to list all the stakeholders and identify different types of users for your software project—WRONG! Although it may be obvious who holds the checkbook for your project and who the "average" users will be, many other people and user roles are not so obvious. Unaccounted for stakeholders and users result in missed requirements and often leave important conflicts unresolved. Even worse, you can lose support—and the whole project can fail—if important people are left out of the process. As Linda Westfall demonstrates unique "brain writing" techniques in a facilitated, interactive requirements workshop, you will learn ways to identify a complete list of the important project stakeholders and user roles. After pruning the stakeholder list to eliminate duplicates, Linda demonstrates how to define a requirements elicitation strategy to select appropriate techniques for each stakeholder. Practice techniques for resolving stakeholder conflicts and take back a stakeholder identification checklist to ensure that you consider a broad range of stakeholder categories for your projects.

W21 SPECIAL TOPICS

Eight Steps to a Virtualized Test Environment

John Janakiraman, illumita

Virtualized software test environments deliver quantifiable benefits—lower lab costs, faster test cycles, and lower IT support overhead. New capabilities in virtualization and virtual test lab solutions are being brought to market by vendors such as VMWare, Surgient, VM Logix, and illumita. These tools promise compelling productivity improvements: richer test scope, tighter lab integration with test tools and processes, and on-demand test infrastructure. John Janakiraman describes capabilities and benefits of virtual test lab environments, offers guidance in adopting a virtual test lab, and shares lessons learned from real world implementations. John walks you through eight important steps to adopting a virtualized environment in your test lab. As John shares the lessons he's learned implementing virtual test labs, find out if a virtualized lab environment is right for your organization.

T1 MANAGING PROJECTS AND TEAMS

Fifteen Tips for Speeding Up Your Project

Johanna Rothman, Rothman Consulting Group, Inc.

Faster is better for software projects—if and only if all the right elements are in place and ready to go. Sometimes your organization is in a sweet spot—that period of time when your project should start immediately. Other times, it's better to wait. Join Johanna Rothman to discover how to decide whether your project is ready to go, including how to help your managers define the project portfolio to see where your project fits in and how it supports your organization's goals. Johanna discusses fifteen ways to measure and steer projects to help you get to the end faster. Learn about rolling wave scheduling, continuous integration, time-boxing, and much more. In this interactive session, you'll discuss the meaning of "done" so you can help the team finish a project sooner and avoid having it drag on. Although you don't have to use all of the tips, the more you use, the faster your project will run.

T2 PERSONAL DEVELOPMENT

Your Attention Please: Concentration is a Learnable Skill

Lee Devin, Swarthmore College

With the possible exception of the fakir walking barefoot on a bed of nails, no one can focus attention on a single object for more than about fifteen seconds. There's a practice, though, that anyone can learn to accommodate this fact and go on to solve vexing problems quickly and creatively. Lee Devin borrows from the skills that actors develop to direct their attention so their mind and body behave as if the imaginary world they've created is real. Similarly, when you watch a good movie or read a great novel, you direct your attention with single-minded intensity. Using theatre exercises, Lee introduces you to the techniques of warm-up and the skills of concentration. Although simple, they're by no means easy. Learn and practice these mind-bending exercises and take away a powerful tool that can increase your concentration both at work and in your personal life.

T3 AGILE DEVELOPMENT

Continuous Integration: The Cornerstone of a Great Shop

Jared Richardson, 6th Sense Analytics

Jared Richardson believes that of all the development practices being promoted today the best single practice is continuous integration. It's a simple concept—you run a software program that monitors your source code in an automated version control system. When anything changes, your code is automatically checked out, re-built, and all the automated tests are re-run. Continuous integration gives you an early warning if anything in the most recent changes broke the software. Continuous integration forces you to use 100 percent source code management and demands a solid, automated build script. It provides a framework for your automated tests to grow, live, and thrive. Continuous integration becomes a new "team member" who keeps a constant eye on your code and provides the reminders you need to keep the product solid and your team on track. Join Jared to learn the steps to introduce continuous integration into your shop and how to set it up yourself.

T4 METRICS

The Uncertainty Surrounding the Cone of Uncertainty

Todd Little, Landmark Graphics Corporation

Barry Boehm first defined the "Cone of Uncertainty" of software estimation more than twenty-five years ago. The fundamental aspect of the cone is quite intuitive—that project uncertainty decreases as you discover more during the project. Todd Little takes an in-depth look into some of the dynamics of software estimation and questions some of the more common interpretations of the meaning of the "cone." Todd presents surprising data from more than one hundred "for market" software projects developed by a market-leading software company. He compares their data with other published industry data. Discover the patterns of software estimation accuracy Todd found, some of which go against common industry beliefs. Understanding the bounds of uncertainty and patterns from past projects help us plan for and manage the uncertainties we are sure to encounter. Take back a collection of measures and metrics you can use to track and analyze uncertainty in your current and next project.

T5 SOA

A Toolkit for Assessing SOA Readiness

Jerry Smith, Symphony Services

Before charging "full speed ahead" into the land of service-oriented architecture (SOA), you need help to ensure success and mitigate the risks inherent in such major systems changes. Jerry Smith provides proven tools for assessing SOA readiness and outlines the essential steps to implementing SOA. Jerry presents reference SOA architectures that demonstrate solid standards and specifications to compare with your implementation plans. He introduces an SOA Maturity Model to help you understand your current organizational and technological state. The SOA Maturity Model is a communications tool that outlines how the organization's SOA implementation will evolve along both business and technical lines. Jerry outlines the various stages the model entails and how to apply it so that technical and organizational changes are easily coordinated across the enterprise. With this new toolkit in hand you can deliver a clear action plan to drive the improvements only SOA makes possible.

T6 SECURITY

Software Security Assessment: The Naked Truth

Herbert (Hugh) Thompson, People Security

With software running our most critical business processes, we need to think about both its utility and the risk it can add to those processes. Hugh Thompson describes some of the best current techniques to efficiently assess software security risk. Hugh identifies the biggest risks to your software systems, presents the major categories of security vulnerabilities with their business consequences, and how you can begin an effective software risk assessment process. Specifically, Hugh discusses the 17 critical questions to ask vendors, software component suppliers, and software-as-a-service (SaaS) providers about their product before you commit to using it. He describes how to benchmark your own software security practices, the top application security flaws that put your business at risk and their symptoms. You'll also learn to make more security-savvy software acquisition, development, and outsourcing decisions.

T7 SPECIAL TOPICS

SOX and HIPPA and RESPA, Oh My! Mastering Software Compliance

Elle Ringham, Cognizant Technology Solutions

Determining whether legal and contractual issues apply to your development efforts isn't always simple. There may be some obvious factors—industry regulations, service level agreements (SLAs), and state or federal agency oversight. However, other factors may not be so obvious. Even today, the Sarbanes-Oxley Act is largely legally untested, subjecting your company to unknown legal risks. Examining legal, compliance, and audit issues throughout the QA process lifecycle, Elle Ringham discusses the right questions to ask and what to do with the answers. She provides guidelines you need to work with stakeholders, developers, attorneys, and auditors. Incorporate legal requirements and compliance issues as part of the architecture, development process, and for all strategic business initiatives. Take away audit templates, metrics to help you know where you stand, and sample reports you may need to produce in your current or next project.

THURSDAY, JUNE 12, 12:45

T8 MANAGING PROJECTS AND TEAMS

A Recipe for Self-Induced Project Failure

Mike Seavers, BJC HealthCare

Even with the progress of development practices in the last few years, have we significantly increased the percentage of successful projects? Are we meeting the needs of our customers at a cost they can afford? Are we adding business value with every project? Mike Seavers believes that, in many cases, our development practices are no better than before. The core of the problem is a fundamental misunderstanding between customers and software organizations about the difference between needs, driving forces, and features. Customers are needs-driven, but they speak in terms of features without understanding why. Similarly, developers are feature-driven and rarely address customers' actual needs. By uncovering these misconceptions, Mike explains how to better understand your customers—what they really want and value, how to identify and remove feature-driven behaviors in our organizations that prevent us from being successful, and how, by mobilizing your organization around real value, to turn your customers into advocates rather than adversaries.

T9 PERSONAL DEVELOPMENT

The Leadership Imperative: Creating a Culture of Trust

Pollyanna Pixton, Accelinnova

In our personal and business lives, many of us know leaders who successfully foster environments of incredible creativity, innovation, and ideas—while other leaders try but fail. So, how do the top leaders get it right? Going beyond the basics, Pollyanna Pixton explores with you the ways that the best leaders create "safety nets" that allow people to discover and try new possibilities, fail early, and correct faster. Removing fear and engendering trust make the team and organization more creative and productive as they spend less energy protecting themselves and the status quo. Pollyanna shares the tools you, as a leader, need to develop open environments based on trust—the first step in collaboration across the enterprise. Learn to step forward and do the right thing without breaking trust. Find out when and how to acknowledge and reward trust in your team and organization.

T10 AGILE DEVELOPMENT

The Impact of Poor Estimating and How to Fix It

Mitch Lacey, Ascentium

Running Scrum by-the-book for three months, the team was continually failing to meet its delivery commitments. As a result, trust between the business managers and the team degraded almost to a point-of-no-return. Even holding bi-weekly retrospectives, the team could not pinpoint the problems causing their inability to release iterations on time. Mitch Lacey was asked to assist the team in finding the root causes of the problems. He analyzed multiple aspects of the project—from individual work items to planning meetings. Multiple issues were identified, but one thing stood out above all others—the estimation process they used had caused the team to miss its deadlines again and again. Mitch discusses the estimation problems the team was having, how they were fixed, and the resulting improvements in customer satisfaction value delivered. He presents new concepts, ideas, and techniques to help you improve your estimating practices and outlines the benefits of employing collaboration tools in the process.

T11 METRICS

Function Point Analysis: A Quick and Easy Primer

David Garmus and David Herron, The David Consulting Group

The function point metric is used by many organizations worldwide to more accurately size systems. Knowing the size of a system allows developers to better meet customer demands of functionality within time and budget and communicate about these issues with the system "owners." Based on the latest version of the International Function Point Users Group (IFPUG) Counting Practices Manual, David Garmus and David Herron provide a detailed explanation of the rules engineers must follow to accurately count function points. Join them to learn the value and use of function points within an overall software measurement program and the basics of how and when to use function point analysis (FPA). Examine real-world examples of software to see how to identify the different functional components according the IFPUG's FPA standards. To assist them in applying IFPUG rules for counting function points, all participants will receive a Certified Reference Card.

T12 SOA

SOA Testing Challenges and Proven Practices

Guruprasad Gopalakrishnan, Wipro Technologies

The best thing about Service Oriented Architecture (SOA) is its flexibility—a heterogeneous computing environment in which different services and service providers can use different technologies; loose coupling of components to allow any application to make use of service capabilities; and ad-hoc integration of applications within and across organizations. However, from a tester's perspective, these very advantages make the testing of Web services and SOA-based applications highly complex. Testing Web services through the front-end of applications is usually ineffective. Tracking defects to their source is difficult because of the layered application designs. Instead, you must design and execute mostly non-functional tests for compliance to standards, interoperability, security, reliability, and performance. Join Guruprasad Gopalakrishnan to find out about the challenges inherent in SOA testing, discover the tools available to help you in your quest, and go home with a "tested" approach you can use to optimally confirm SOA operations.

T13 SECURITY

Automating Security Testing with cUrl and Perl

Paco Hope, Cigital

Although all teams want to test their applications for security, our plates are already full with functional tests. What If we could automate those security tests? Fortunately, most Web-based and desktop applications submit readily to automated testing. Paco Hope explores two flexible, powerful, and totally free tools that can help to automate security tests. cUrl is a free program that issues automatic basic Web requests; Perl is a well-known programming language ideally suited for writing test scripts. Paco demonstrates the basics of automating tests using both tools and then explores some of the more complicated concerns that arise during automation—authentication, session state, and parsing responses. He then illustrates simulated malicious inputs and the resulting outputs that show whether the software has embedded security problems. The techniques demonstrated in this session apply equally well to all Web platforms and all desktop operating systems. You'll leave with an understanding of the basics and a long list of resources you can reference to learn more about Web security test automation.

T14 SPECIAL TOPICS

Real Software QA

Linda Westfall, The Westfall Team

With the ever-increasing demand for software products, it is imperative for organizations to move away from trying to "test in" software quality and move toward adopting a comprehensive, total-life cycle software quality management approach. Such an approach is the basis of real software quality assurance (QA)—all the planned and systematic actions necessary to provide confidence that a system will perform satisfactorily in production. In contrast to software quality control (QC), which consists of detection activities such as testing that are *product* focused, software QA is *process* focused. Software QA includes defining, establishing, and monitoring the control practices that ensure policies, systems, and processes are effective and efficient across the entire software development life cycle. Linda Westfall focuses on defining the elements and techniques of a comprehensive software QA program. Her goal is to help you identify gaps and process improvement opportunities in your organization's software QA activities.

T15 MANAGING PROJECTS AND TEAMS

Maximizing ROI on New Technology Acquisition

Chris Ronak, Divestco, Inc.

IT departments and software technologists must invariably navigate many challenges when planning to acquire new tools, invest in new technology, fund new technology projects, and introduce process changes. How do you get the most out of these investments without upsetting existing mission-critical processes or projects? Subsequently, how do you rapidly turn your new technology into a successful release that augments your product suite? Chris Ronak shares his experiences and offers his recommendations on how to best integrate newly acquired technology into mainstream development processes and projects. A strategic acquisition must provide missing functionality that enhances your existing product suite or technical framework—and it must be implemented without hindering or stopping progress on other business-critical projects. Join Chris to explore ways to build an extensible core technology framework that allows you to successfully integrate new technology and innovative practices into your existing development organization.

T16 PERSONAL DEVELOPMENT

An Alternative to Consensus: Accelerating Effective Decisions

Michele Sliger, Sliger Consulting

Software development teams don't always need, want, or have time to make decisions via group consensus. And project leaders often already feel over-burdened with the multiple decisions they have to make on their own. But there is a middle path—an alternative to consensus—in which shared responsibility for decision-making provides for input from many and one voice to represent the team and make the final choice. In this decision-making process, a team member volunteers to be the decision-maker on a particular issue with only one mandatory rule—seek guidance. The greater the impact the decision will have on the organization, the wider the quest for advice must be—all the way to the board of directors, if appropriate. Join Michele Sliger to learn how this approach to decision-making might be right for your organization.

T17 AGILE DEVELOPMENT

Test-Driven Analysis: Focus on the User, Not the Code

Ken Pugh, Net Objectives

Test-driven analysis (TDA) focuses on the system at the user interface level. By concentrating on scenarios from the users' point of view, TDA examines the inputs and outputs of a system and how the state of that system changes as a result of actions performed on the system. Further, TDA concentrates on creating tests that help developers and business analysts understand the problem domain and the system requirements. Ken Pugh explores how tests created during TDA efforts relate to user acceptance tests and how the TDA tests can propagate to those created in a test-driven development framework. Together with Ken, you'll create analysis tests for a sample system and differentiate between various types of tests—acceptance, analysis, and design. This class is about the process of inquiry we need during analysis to answer the question "I know you're ready for testing, but what are you testing for?"

T18 METRICS

You Just Don't Understand Me: Interdisciplinary Awareness to the Rescue

Mike Tholfsen, Microsoft

Different disciplines and departments in an organization still have the same goals, but often misunderstand one another. We have all heard someone say about another group, "Those people are clueless." The irony is that "those people" are saying the same thing back under their breath. Within the software disciplines, poor understanding, lack of communication, and unfortunate stereotyping are often commonplace. Presenting a new concept and team exercise called Interdisciplinary Awareness, Mike Tholfsen helps us focus on the importance of team dynamics in building good software. With case studies from both Microsoft Office and Windows teams, Mike shows how they built stronger trust within and between teams. Incorporate this exercise in your group and discover how interdisciplinary awareness can lead to greater understanding and appreciation, a stronger sense of team, and a higher degree of trust.

T19 SOA

The Challenges of SOA Security

Jimmy Xu, CGI, Inc.

Service Oriented Architecture (SOA) has many security challenges. To address these challenges, it is not enough to set up a secure operational infrastructure. SOA security must be implemented in all key areas of software development—architecture, design, platform, governance, requirements, development, and testing. Jimmy Xu discusses today's SOA security challenges and explains why it is important to address these challenges inside software development. He presents the latest security practices: standards compliance; review of architectural blueprints and SOA platforms; secure SDLC process; threat modeling; secure coding; and security testing. This session not only prepares you to delve into the details of SOA security methodology, process, and techniques but also gives you the background information you need to plan and scope security assurance activities in your SOA development projects.

T20 SECURITY

Fuzzing: New Tests for Robustness and Security

Ari Takanen, Codenomicon

Traditional security measures are doomed to fail because they are focused only on defending against known attacks—and studies show that more than 80 percent of software will likely crash when extensive negative testing is employed. Fuzzing is a new, proactive technique for discovering security vulnerabilities and robustness issues in software. Although fuzz testing is most often based on some form of syntax checking, random input testing also can be appropriate. Fuzzing is valuable during development when application testers use the technique to surface issues and in production when security testers use it for audits. Any type of system can be fuzz tested—from enterprise solutions to consumer products such as mobile phones and set-top TV cable boxes. Ari Takanen discusses the origins of fuzzing, explains the different technologies used by fuzzers, and identifies current fuzzing tools, their uses and limitations. Ari describes various metrics related to fuzzing that allow you to measure effectiveness and compare the efficiency of various tools.

T21 SPECIAL TOPICS

Scenario-Based Architecture Reviews: A Quality Process

Kevin Bodie, Pitney Bowes

Quality attributes—performance, scalability, availability, maintainability, etc.—are often referred to as the non-functional requirements. Unfortunately, these critical factors often are specified through vague platitudes rather than explicit statements. Participatory, scenario-based architecture reviews are essential to determine if the architecture meets the system's quality attributes as well as the functional requirements. Scenario-based architecture reviews rigorously examine the characteristics of the system to determine strengths, limitations, operating bounds, and overall requirements satisfaction. Results include the impact on and risk of architectural choices for the system. Kevin Bodie details the architecture review planning process, illustrates its execution, and shows you the results with specific deliverables—the Architecture Readout Presentation and Executive Architecture Readout. Kevin presents real-world examples of these deliverables and offers you a review process model to implement in your organization.

EVENT LOCATION

Experience the romance of Italy in the heart of the Las Vegas Strip. Beauty and grace surround you at The Venetian Resort Hotel Casino, where every detail has been carefully designed to provide you with a memorable stay—from the world's largest standard suites to the city's most elegant casino. Immerse yourself in the enchanting atmosphere of this all-suite Las Vegas hotel resort. Choose from nineteen remarkable restaurants—featuring an unprecedented six James Beard Award-winning chefs, including Wolfgang Puck, Emeril Lagasse, and Thomas Keller. Browse the eighty international boutiques of The Grand Canal Shoppes, offering collectibles and couture from such renowned names as Jimmy Choo, Lladro, Dooney & Bourke, and Burberry. Rejuvenate at Canyon Ranch SpaClub—the very picture of rest and relaxation—where a full-service salon, fitness center, rockclimbing wall, and indulgent packages await. Discover the ideal Las Vegas resort vacation destination and business retreat where exquisite all-suite accommodations, attentive service, and endless attractions converge in a magnificent setting.

VEGAS HIGHLIGHTS

So what's a person to do in the middle of the desert? The sky's the limit in Las Vegas! Here are some neat things to do in the city that never sleeps...

- Take a roller coaster ride around New York skyscrapers on the Manhattan Express
- Snap your picture with your favorite celebrity at Madame Tussaud's Wax Museum
- Get whisked away by the dancing fountains on the ten-acre glittering lake at the Bellagio
- Take a guided tour of the Eiffel Tower, or at least at half the size
- Enjoy an interactive adventure at Star Trek: The Experience™
- Visit the world's tallest Coke® bottle
- Dine from the finest selection of buffets the world has to offer
- Play a little golf...choose from the fifty-seven neighboring clubs
- Lose yourself in the 250 video games at GameWorks
- Visit the Fremont Street Experience for a light and sound show staged ninety feet above a downtown mall
- Shop 'til you drop—there's something for everyone in the hundreds of retail stores just minutes from the Strip
- Drop the kids off at Camp Las Vegas for a fun-filled day of roller coasters, movies, and animal visits

What Past Delegates are Saying...

"I extremely enjoyed the tutorials and regular conference keynote speakers and breakout sessions. All presenters were knowledgeable and open to one-on-one discussions outside of the sessions."
— Roger Blecha, Torex Hospitality

"The tutorials were very informative; provided practical information that I can take home and use right away."
— Susan Hennessey, TU Group

"GREAT CONFERENCE. This is the right conference for people at all levels. If you are in a quality profession—this is the right conference."
— Mukesh Jain, Microsoft Corporation

"Keynote speakers were both informative and entertaining."
— Frank Agrusa, ACL Corporate Systems

"What a great conference! The tutorials and the concurrent sessions were very informative. The speakers were very knowledgeable too! Thank you!"
— Oriana Falco, Thales e-Security

"The tutorials offered were extremely informative. There is take back value I can use immediately. Networking opportunities were everywhere with people, companies and opportunities from coast to coast."
— Chris Riley, Nationwide Insurance Company

"It's hard to find a multi-dimensional conference that applies to the full development lifecycle, but the Better Software conference achieves just that! Keep it up & thanks for making it available to the industry."
— Wendy Haroldson, Thompson

"The information delivered at this conference was valuable & immediately actionable. I have enough process improvement, methodology improvement and team leadership ideas to keep our team moving forward for a long time. Great conference!"
— Michael Nathan, Wolters Kluwer

2006 + 2007 Jolt Product Excellence Award
WINNER!

Sign up now for our FREE Community Edition.
Rally's award-winning Agile life cycle management tools for a single team!

www.rallydev.com/bsw

RALLY SOFTWARE
Scaling Software Agility

CONFERENCE SPONSOR

Rally Software Development helps companies successfully adopt and scale Agile software development practices, from an initial pilot to enterprise rollout. Rally's family of Agile life cycle management products, which received back-to-back Jolt Product Excellence awards in 2006 and 2007, map an incremental road to Agile adoption. The company's world-renowned coaching services help mentor teams to create internal Agile experts. **www.rallydev.com**

INDUSTRY SPONSORS

Empirix helps organizations implement new Web technologies with confidence. We provide award-winning automated testing and monitoring solutions for QA and network operations professionals who deploy and manage business critical Web applications. Because Empirix spans the evolution of advanced technologies—from testing in R&D labs through monitoring the end-user experience—our clients gain unrivaled insight into technology performance, enabling them to accelerate application rollouts and ensure quality in production. **www.empirix.com**

McCabe Software provides Software Quality, Release, and Configuration Management solutions worldwide. McCabe IQ uses a comprehensive collection of software metrics to analyze and visualize the quality and testing of critical applications. McCabe CM utilizes exclusive "Integrated Difference" technology to manage software changes across versions, platforms, and the enterprise. **www.mccabe.com**

Net Objectives provides targeted Lean-Agile Training Solutions with coordinated technical offerings in design patterns and test-driven development to help you foster sustainability, and deliver product and business value faster. We offer the most comprehensive Lean-Agile curriculum in the industry, Professional Scrum Certifications, Assessments, on-site Training, custom curricula, Coaching, and Enterprise Agility Consulting services for teams, management and IT. **www.netobjectives.com**

Telelogic® is a leading global provider of solutions for automating and supporting best practices across the enterprise, from powerful modeling of business processes and enterprise architectures to requirements-driven development of advanced systems and software. Telelogic's solutions enable organizations to align product, systems, and software development lifecycles with business objectives and customer needs to dramatically improve quality and predictability, while significantly reducing time-to-market and overall costs. **www.telelogic.com**

VersionOne, recognized by customers as the number one Agile project planning and management tool, helps development teams deliver working software faster by simplifying the process of planning, tracking and managing agile software projects. Since 2002 we've helped more than 4,000 teams in over 40 countries deliver more value to their customers by simplifying and standardizing the adoption of iterative software development. **www.versionone.com**

IN COOPERATION WITH

The International Function Point Users Group (IFPUG) is the recognized leader in promoting and encouraging effective management of application software development and maintenance activities through the use of Function Point Analysis and other software measurement techniques. It is the steward of the international standard of software size measurement – ISO/IEC 20926:2003 Software Engineering. **www.ifpug.org**

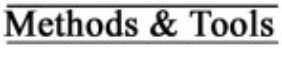
PLUS See these Exhibitors and Sponsors at the EXPO (June 11–12, 2008)

ACM *Queue*	***IEEE Computer Society***	**Rally Software Development**
APLN	**IFPUG**	Software Quality Engineering
ASTQB (American Software Testing Qualifications Board)	**McCabe Software**	SQE Training
AutomatedQA	***Methods & Tools***	**StickyMinds.com**
***Better Software* magazine**	**Net Objectives**	TechExcel
***CoDe* Magazine**	PleaseTech Ltd.	**Telelogic**
Empirix	The Process Group	**VersionOne**
	Q/P Management Group	VMC
	Quantitative Software Management	

Better Software Conference & EXPO sponsors are listed in bold.

For Sponsor/Exhibitor news and updates, visit www.sqe.com/bsce

MEDIA SPONSORS



BETTER SOFTWARE CONFERENCE & EXPO

June 9–12, 2008 • Las Vegas, NV

Easy to Register

ONLINE:
www.sqe.com/bscereg

PHONE:
888-268-8770
904-278-0524

EMAIL:
sqeinfo@sqe.com

CONFERENCE PRICING

Registration Fees:*	On or Before May 9	After May 9	
❑ **Conference + 2 Tutorials**	**$2,195**	**$2,395**	◀ ***Best Value!***
❑ Conference + 1 Tutorial	$1,995	$2,195	
❑ Conference Only (Wed.-Thurs.)	$1,695	$1,895	
❑ Tutorials (2 Days)	$1,390	$1,590	
❑ Tutorial (1 Day)	$795	$895	
❑ Certification Training Course + Conference**	$3,740	$3,940	◀ ***A Savings of $200!***
❑ Certification Training Course**	$2,245	$2,245	

Special Early Bird Offer!

Receive $200 off the regular conference registration fee if payment is received on or before May 9, 2008. See discounted pricing information above.

PAYMENT INFORMATION

The following forms of payment are accepted: Visa, MasterCard, American Express, check, or company purchase order. Payment must be received before the registration is confirmed. Make all checks payable to Software Quality Engineering. You will receive a confirmation package upon payment by check, credit card, or company purchase order. Payment must be received at Software Quality Engineering by May 9, 2008, to take advantage of the Early Bird conference rates listed above.

HOTEL RESERVATIONS

Take advantage of the special conference hotel rate. Call 866-996-6338 to make your reservation, or visit **www.sqe.com/go?bscehotel8**. Space is limited, and early booking is recommended.

CANCELLATION POLICY

Registrations cancelled after May 23, 2008, are subject to a 20% cancellation fee. No cancellations or refunds may be made after May 30, 2008. Substitutions may be made at any time before the first day of the program. Call our Client Support Group at 888-268-8770 or 904-278-0524 to obtain a cancellation code. All valid cancellations require a cancellation code.

SATISFACTION GUARANTEE

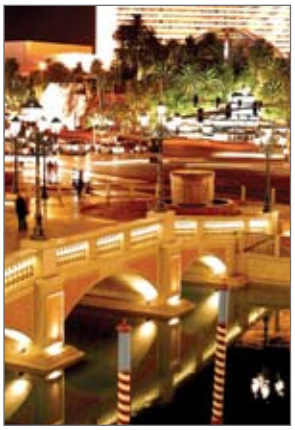
Software Quality Engineering is proud to offer a 100% satisfaction guarantee. If we are unable to satisfy you, we will gladly refund your registration fee in full.

PHOTO RELEASE

From time to time we use photographs of conference participants in our promotional materials. By virtue of your attendance at the Better Software Conference & EXPO, you acknowledge that Software Quality Engineering, Inc. reserves the right to use your likeness in such materials.

* Your registration fee includes $39 for a one-year digital subscription (ten issues) to *Better Software* magazine. If you are a current subscriber, your subscription will be extended for an additional ten digital issues.

** A $250 exam fee is included in the cost of the course registration.



SPECIAL HOTEL RATES FOR BETTER SOFTWARE CONFERENCE & EXPO ATTENDEES!

Rooms are being held at a special conference rate for the Better Software Conference & EXPO attendees. To reserve a room, call the hotel at 888-283-6423 or visit www.sqe.com/go?bscehotel8. To receive the discounted rate, please mention you are attending the Better Software Conference & EXPO. If you need special facilities or services, please notify the agent at the time of reservation. Cancellations on a guaranteed reservation must occur more than 72 hours prior to the specified arrival time to ensure a refund.

ONLINE ACCESS AT THE CONFERENCE

There will be a designated WiFi area at the conference.

WAYS TO SAVE ON YOUR CONFERENCE REGISTRATION

Special Early Bird Offer!

Receive $200 off your registration fee if payment is received on or before May 9, 2008.

PowerPass Discount

PowerPass holders receive an additional $100 off their registration fee. Not a PowerPass member? Visit www.stickyMinds.com/PowerPass to learn more.

Bring a Buddy!

Bring a colleague and each of you saves an additional $200. Any two people registering at the same time save an additional $200 off each registration. Please call the Client Support Group at 888-268-8770 or 904-278-0524 to register. Use promotion code **BBBS** when registering to receive your discount.

Win a Free Conference Registration

Request a brochure or register by May 9, 2008, and be entered to win 1 of 10 free conference registrations!

Please note: This prize is limited to conference and registration fees only. Hotel stay and travel costs are not included in this offer.

Alumni Discount

Better Software Conference & EXPO alumni receive up to an additional $200 off their registration fee.

For **Group Discounts** or more details on our discount policy, please contact the Software Quality Engineering Client Support Group at sqeinfo@sqe.com or 888-268-8770 or 904-278-0524.

BETTER SOFTWARE CONFERENCE & EXPO

JUNE 9-12, 2008
LAS VEGAS, NEVADA
THE VENETIAN

TUTORIALS WORKSHOPS CLASSES

CELEBRATING 5 FIVE YEARS

Alan Shalloway
Andy Glover
Andy Hunt
Andy Kaufman
Ari Takanen
Beth Layman
Bob Hartman
Chris Ronak
Chuck Allison
Dan North
David Garmus
David Herron
David Spann
Ed Weller
Elle Ringham
Guruprasad Gopalakrishnan
Herbert (Hugh) Thompson
Hubert Smits
James McCaffrey
James Newkirk
Jared Richardson
Jean Tabaka
Jeff Patton
Jeff Payne
Jerry Smith
Jimmy Xu
Johanna Rothman
John Janakiraman
Julie Gardiner
Ken Pugh
Ken Schwaber
Kent McDonald
Kevin Bodie
Lee Copeland
Lee Devin
Linda Rising
Linda Westfall
Lisa Crispin
Michael Mah
Michele Sliger
Mike Seavers
Mike Tholfsen
Mitch Lacey
Nelson Perez
Paco Hope
Payson Hall
Pollyanna Pixton
Rebecca Wirfs-Brock
Richard Bender
Rob Myers
Robert Galen
Stacia Broderick
Tim Korson
Todd Little
Vladimir Pavlov
Will McKnight

Conference Sponsor:
RALLY SOFTWARE

Industry Sponsors:
empirix Be Confident.
McCabe SOFTWARE
NetObjectives
Telelogic
VERSIONONE Simplifying Software Delivery

In Cooperation With:

Media Sponsors:
acm queue tomorrow's computing today
APLN
BETTER SOFTWARE MAGAZINE
CoDe www.code-magazine.com
IEEE COMPUTER SOCIETY
IEEE
Methods & Tools
StickyMinds.com

PLUS!
The EXPO *June 11-12, 2008*
Find Solutions to Your Software Development Challenges!
TOOLS • TECHNIQUES • SERVICES • DEMOS • SOLUTIONS

GROUP DISCOUNTS AVAILABLE

www.sqe.com/bsce REGISTER EARLY AND SAVE $200!

SOFTWARE QUALITY ENGINEERING CONFERENCES

330 Corporate Way, Suite 300
Orange Park, FL 32073

PRESORTED
STANDARD
U.S. POSTAGE
PAID
GAINESVILLE, FL
PERMIT NO. 726

IF ADDRESSEE IS NO LONGER EMPLOYED:
Re-route to Director of Software Development