

BETTERTM SOFTWARE

A TECHWELL PUBLICATION

HOW CAN I USE THIS?

Use mnemonics to remember how to test your app's usability

PRODUCT OWNERSHIP

Four distinct roles to lead an agile team

REQUIREMENTS REUSE

*Fantasy
or Feasible?*

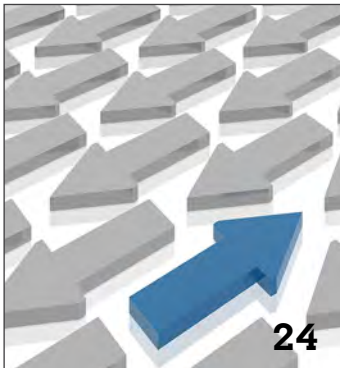




CONTENTS



16



24

in every issue

Mark Your Calendar	4
Editor's Note	5
Contributors	6
Interview with an Expert	9
TechWell Spotlight	10
Product Announcements	28
FAQ	31
Ad Index	32

Better Software magazine—The companion to TechWell.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 at <http://www.BetterSoftware.com> or call 800.450.7854.

features

12 COVER STORY REQUIREMENTS REUSE: FANTASY OR FEASIBLE?

Software development teams think nothing about reusing code, but what about requirements? The benefits include faster delivery, lower development costs, consistency across and within applications, fewer defects, and reduced rework.
by Karl Wiegers and Joy Beatty

16 MOVING BEYOND THE BACKLOG: THE FOUR QUADRANTS OF PRODUCT OWNERSHIP

What do you consider the role of product owner in an agile development project to be? Bob presents a compelling perspective that a product owner has four distinct critical roles that can prove impactful to a team's success.
by Bob Galen

20 USABILITY TESTING WITH "CAN I USE THIS?"

David introduces a unique approach for application usability testing. Each letter of the memorable "CAN I USE THIS?" will assist with test idea generation.
by David Greenlees

24 ACHIEVING A PROJECT'S BENEFITS AND GOALS: HOW CHANGE MANAGEMENT CAN HELP

Studies have shown that leading organizations frequently have less than successful results when incorporating significant changes. Kris presents useful advice dealing with staff, communication, and what to do for the long term after the change takes place.
by Kris Irwin

columns

7 TECHNICALLY SPEAKING USING DEVOPS TO DEVELOP RELIABLE SOFTWARE

by Bob Aiello and Leslie Sachs
How do we build more reliable, complex systems in a way that is both pragmatic and economically feasible? Many of the DevOps practices provide the key to building better software that can be maintained, upgraded, and supported from its first installation to its eventual retirement when the system is no longer required.

32 THE LAST WORD PMO 2.0: REBOOTING THE IT PROJECT MANAGEMENT OFFICE

by Tony McClain
More companies are adopting project management offices, and with that additional oversight and structure, their impact can often miss the mark. Tony explains that a "kick in the pants" might be necessary for a PMO to get staff buy-in that delivers successful project results.

MARK YOUR CALENDAR

**BETTER
SOFTWARE**
A TECHWELL PUBLICATION

training weeks

<http://sqetraining.com/trainingweek>

Testing Training Week

March 24–28, 2014

Boston, MA

May 12–16, 2014

San Diego, CA

June 9–13, 2014

Chicago, IL

Agile Software Development Training

November 10–12, 2013

Boston, MA

June 1–3, 2014

Las Vegas, NV

software tester certification

<http://sqetraining.com/certification>

Foundation Certification Training

November 10–12, 2013

Boston, MA

December 3–5, 2013

Phoenix, AZ

February 25–27, 2014

Baltimore, MD

March 4–6, 2014

San Francisco, CA

Philadelphia, PA

March 11–13, 2014

Cincinnati, OH

Denver, CO

March 18–20, 2014

Raleigh-Durham, NC

LA area

Advanced Tester Certification

March 3–7, 2014

San Francisco, CA

conferences

Agile Development Conference East

<http://adceast.techwell.com>

November 10–15, 2013

Sheraton Boston Hotel

Boston, MA

Better Software Conference East

<http://bsceast.techwell.com>

November 10–15, 2013

Sheraton Boston Hotel

Boston, MA

STARCANADA 2014

<http://starcanada.techwell.com>

April 5–9, 2014

Hilton Toronto

Toronto, ON

STAREAST 2014

<http://stareast.techwell.com>

May 4–9, 2014

Rosen Centre Hotel

Orlando, FL

Agile Development Conference West 2014

<http://adcwest.techwell.com>

June 1–6, 2014

Caesars Palace

Las Vegas, NV

Better Software Conference West 2014

<http://bscwest.techwell.com>

June 1–6, 2014

Caesars Palace

Las Vegas, NV

Publisher
Software Quality Engineering Inc.

President/CEO
Wayne Middleton

Director of Publishing
Heather Shanholtzer

Editorial

Better Software Editor
Ken Whitaker

Online Editors
Beth Romanik
Jonathan Vanian

Production Coordinator
Donna Handforth

Design

Creative Director
Catherine J. Clinger

Advertising

Sales Consultants
Daryll Paiva
Kim Trott

Sales Coordinator
Minda Crosby

Marketing

Manager
Sean Burgess



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.
340 Corporate Way, Suite 300
Orange Park, FL 32073

GREETINGS FROM THE NEW EDITOR



I'm pleased to be the new editor starting with this edition of *Better Software* magazine. Being in the software/IT world for well over twenty-five years and the author of books and articles, I've always wanted to be a part of spreading best practices for software development and IT practitioners. In this role, I have the opportunity to work with inspiring authors and an incredible magazine staff to put out an amazing periodical.

I've known about *Better Software* magazine for some time as a result of attending Software Quality Engineering conferences that take place throughout the year. What I really like about Software Quality Engineering—and, by extension, *Better Software* magazine, TechWell.com, and the community websites—is the entire organization's focus on education and communication for the software development and IT community. In my opinion, it is rare to find a company that really “gets it.”

While I've been a *Better Software* subscriber for some time, I've mentioned the magazine to fellow technologists, and most have never heard of it. So, we have some work to do in order to get the word out!

In order for *Better Software* magazine to grow and sustain its mission to educate software and IT professionals, we need to spread the word. Please take a moment and use your influence in social media to let people know about *Better Software*. You can mention it on LinkedIn, Facebook, or Twitter—or just forward this issue, to two or three other folks. A BSM subscription is complimentary, so put your credit card away!

If you're interested in writing an article for *Better Software* magazine, please send me an email at kwhitaker@sqe.com. I'm looking for article proposals for agile, testing, project and people management, configuration management, ALM, development, and any other topic you think is relevant to today's software professionals. You are out there doing the work, so you tell me what you want to read more about in *Better Software* magazine.

Finally, I'd like to thank Joey McAllister, who was the *Better Software* magazine editor for a long time and who has been so great at helping me through this transition. I met the Software Quality Engineering staff in Orange Park, Florida, and I'm proud to be associated with such a fun and dedicated group.

I hope you enjoy this issue! (And don't forget to tell your friends about *Better Software* magazine!)

Ken Whitaker

kwhitaker@sqe.com

Twitter: @Software_Maniac



BOB AIELLO is a consultant, technical editor for CMCrossroads.com, and coauthor of *Configuration Management Best Practices: Practical Methods that Work in the Real World*. He has more than twenty-five years' experience as a technical manager in several top New York City financial services firms. Bob served as vice chair of the IEEE 828 Standards working group (CM planning) and is a member of the IEEE Software and Systems Engineering Standards Committee management board. Contact Bob at bob.aiello@ieee.org or <http://www.linkedin.com/in/bobaiello>.



JOY BEATTY is a VP at Seilevel, a business analysis consulting firm whose mission is to redefine the way software requirements are created. With fifteen years of experience, Joy evolves new business analysis methods and helps customers implement best practices that improve their requirements process. She worked with IIBA to develop their BABOK, writes articles about requirements methodologies, and presents at business analysis conferences. Joy is the coauthor of *Visual Models for Software Requirements* and *Software Requirements*, 3rd Edition. She can be reached at <http://www.seilevel.com> and joy.beatty@seilevel.com.



BOB GALEN is an agile methodologist, practitioner, and coach helping guide companies and teams in their pragmatic adoption and organizational shift toward Scrum and other agile methods. He is currently president and principal consultant at RGCG LLC. Bob regularly speaks at international conferences and professional groups on a broad range topics related to software development. He is a certified Scrum coach (CSC), certified Scrum product owner (CSPO), and an active member of the Scrum Alliance. Bob authored the books *Agile Reflections* and *Scrum Product Ownership*. He can be reached at bob@rgalen.com.



DAVID GREENLEES has been testing software for more than ten years, many of these spent in one of Australia's largest government departments, while more recently undertaking a consultant role in multiple organizations. He is extremely passionate about the betterment of the software testing craft and context-driven testing. David has published several articles and he blogs regularly at <http://www.martialtester.wordpress.com> and <http://hellotestworld.com>. In 2012 David founded the Australian Workshop on Software Testing. Currently authoring a book on the subject of software testing and martial arts, his passion outside of work. David can be followed on Twitter [@MartialTester](https://twitter.com/MartialTester).



KRIS IRWIN is a senior manager at ACME Business Consulting with ten years of consulting experience in business strategy and planning, transition management, mergers and acquisitions, operations governance, and design and delivery of new technology applications. Kris has led teams through complex business and technology change and enabled end-to-end business processes and systems for large, international companies, including eCommerce, merger and acquisition integrations, shared services and distribution centers, and ERP implementation projects. Kris can be reached at kirwin@acmebusinessconsulting.com.



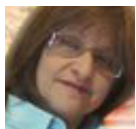
TONY MCCLAIN is a senior client partner at Geneca and brings more than thirty years of professional management experience to his role. He serves Geneca clients by leveraging technology to solve their business challenges and ensuring they get full value from their software investment. Tony has extensive industry experience in retail food management, enterprise software and services, media, financial services, and health care. He can be reached at tony.mcclain@geneca.com.



RYAN OLIVETT is an agile consultant at LitheSpeed, skilled in business process optimization, data analysis, lean management, and Scrum. A certified ScrumMaster and Scrum product owner, Ryan leverages his diverse background and skill set on client engagements to identify problems and provide creative solutions. He primarily works on organizational assessments and process documentation at companies that range in size from small startups to large enterprises. Ryan can be reached at ryan.olivett@lithespeed.com.



BOB PAYNE is the VP of enterprise consulting services at LitheSpeed and a leading proponent of agile methodologies and agile engineering practices, with twenty-five years of project management, software development, engineering, and business experience. An early adopter of Extreme Programming (XP), Bob has worked exclusively as an agile coach and practitioner since 1999. As host of the AgileToolkit podcast, he has produced more than 175 podcasts, recording a variety of industry leaders and agile practitioners. Bob is the founder of Agile Philanthropy and chairperson of the AgileDC conference series. Reach Bob at bob.payne@lithespeed.com.



LESLIE SACHS is a New York State-certified school psychologist and COO of Yellow Spider Inc. She is coauthor of *Configuration Management Best Practices: Practical Methods that Work in the Real World*. Leslie has more than twenty years of experience in psychology and has provided many interventions designed to improve social and educational functioning of individuals and groups. Reach Leslie at LeslieASachs@gmail.com or <http://www.linkedin.com/in/lesliesachs>.



KARL WIEGERS is principal consultant at Process Impact in Portland, Oregon, and coauthor with Joy Beatty of *Software Requirements*, 3rd Edition. Karl is the author of numerous articles and books, including a memoir of life lessons titled *Pearls from Sand: How Small Encounters Lead to Powerful Lessons*. When not at the keyboard, Karl enjoys wine tasting, volunteering, playing the guitar, and writing and recording songs. You can reach him at <http://www.processimpact.com> or <http://www.karlwiegers.com>.

Using DevOps to Develop Reliable Software

DevOps best practices are focused on building in quality from the beginning by automating the application build, package, and deployment.

by **Bob Aiello** | bob.aiello@ieee.org and **Leslie Sachs** | LeslieASachs@gmail.com

We interact with software for almost everything we do on a daily basis, from the cars we drive to the web-based systems we use to purchase products, pay bills, and—for many of us—perform our jobs. We all want software systems to be reliable and to provide the functionality we have come to expect. This includes ease of use, sophisticated features, and even the ability to learn our patterns of behavior and anticipate our every request.

The reality is that while computers do more today than ever before, we also witness many serious incidents that adversely impact us—glitches including faulty 911 call systems and incidents impacting high-speed trading firms and the trading exchanges themselves. Serious outages affecting high-profile trading markets such as the NASDAQ, New York Stock Exchange, Euronext, Chicago Board Options Exchange, and the Tokyo Stock Exchange have the potential to impact our economy and could even undermine our global financial services infrastructure.

The fact is that we, as technology professionals, know how to develop reliable software. This article describes some of the industry best practices, including DevOps, that can be used to do so.

Well-respected standards organizations such as the IEEE, ANSI, and NIST produce excellent guidelines, including Measures of Dependability [1] and Software Safety Plans [2], for developing systems that are used to sustain nuclear power plants, medical life support, missile defense, and other technology infrastructures that must be reliable and highly effective.

Agile and lean are helping to evolve our software development methodologies, and the ITIL v3 framework provides guidance for reliable IT service management. Despite these emerging best practices, there are still many high-profile systems experiencing serious technology glitches resulting in widespread systems outages. How can we prevent such outages from occurring?

Some in the industry believe technologies such as high-speed trading systems have reached a level of complexity that make outages inevitable, and that it is just too expensive and impractical to implement the same level of care in creating a trading system that you would for a nuclear facility or other mission-critical system. This view assumes that creating reliable software is simply too costly and overly burdensome. The fact is that we know exactly how to create reliable systems, and most of the recent high-profile glitches were related to a

lack of adequate software and systems development methodology, including configuration and release management.

Cost is indeed an important issue, although perhaps not in the way some folks think. We simply cannot afford to build systems that are *not* designed to be maintained, upgraded, and supported throughout their entire product lifecycle, from inception to retirement. It is too expensive to create systems that are not developed using methodologies

based upon industry best practices such as continuous integration, deployment, and DevOps.

So, how exactly do we build more reliable, complex systems in a way that is both pragmatic and economically feasible? Many of the DevOps practices provide the key to building better software that can be maintained, upgraded, and supported—from the first installation to the eventual retirement when the system is no longer required.

Building better software is a matter of following W. Edwards Deming's advice of building in quality from the very beginning, including the deployment process itself. [3] DevOps best practices are focused directly on building in quality from the beginning by automating the application build, package, and deployment from the early stages of the development effort. This approach leads to a well-defined and mature deployment framework, which helps speed development and ensures that systems can be maintained and upgraded as necessary.

“These measures form an emerging industry best practice that is becoming known as the secure trusted application base ...”

Not too long ago, we worked with a high-speed trading firm with a monthly release process that was constantly problematic. We approached fixing this problem in two very specific ways. The first was to get the organization to agree to more frequent releases that were smaller in scope and, consequently, much less risky. More frequent releases also enabled us to practice and improve our deployment process. Second, we fully automated every aspect of the release, including testing each step of the build, package, and even the deployment process. We discovered defects in the way the Ant and Maven build scripts were working that had long gone unnoticed until too late. By automating each step of the deployment pipeline and testing each step, our automated procedures were ultimately more reliable. When something tested was faulty, the system failed early on, giving us an opportunity to address the issue earlier in the process than before.

Most companies only start automating the build, package, and deployment toward the end of the development lifecycle. The right approach is to design systems to be deployable from the very beginning. It is essential to build software that uses embedded, immutable version identifiers to ensure that a physical configuration audit can verify that the correct code has been deployed, as well as cryptography (e.g., message authentication codes SHA1 and MD5) to detect any unauthorized changes. These measures form an emerging industry best practice that is becoming known as the *secure trusted application base*, and, from a DevOps perspective, they also should include the operating system itself.

Other essential practices include robust and feature-rich version control solutions that provide support for code variants and the ability to reliably merge code when necessary. Automated build procedures also should include code scanning tools and a method to help detect issues early in the process. Early, continuous integration and deployment are absolutely essential for successful deployment automation.

DevOps makes the case that we need to start earlier in the process by having development and operations (along with QA testing and information security) teams collaborate throughout the entire development lifecycle. We need to embrace the fact that developing a deployment framework is a first-class software development effort in itself. If you want reliable systems, then getting qualified build and release engineers involved up front is essential. This effort should include automating every step of the application build, package, and deployment, starting with builds based on secure source code baselines. It is essential to monitor and verify the environment itself, along with the interfaces to essential runtime components. This includes data feeds and legacy systems, which frequently become an unavoidable single point of failure. The reality is that there are always risks that need to be identified and addressed. When a legacy system becomes a single point of failure, then automated monitoring tools are a must-have.

Verification and validation are fundamental to any software and systems delivery effort. Automated build scripts should

also include unit testing procedures as part of the build stream along with both static and dynamic code scanning which can identify sections of the code that should be refactored or even areas where runtime performance may be a problem. Test driven development (TDD), continuous integration, and continuous delivery are all effective and popular techniques to help ensure code quality and, ultimately, software reliability. Requirements management is also an essential area of focus for any software and systems delivery effort. Requirements that are poorly defined are an all too common cause of defects later in the process. One issue is that business analysts and end user representatives are often asked to define the requirements before enough information is available to make the best choices. Iterative development is providing a very effective way to develop a portion of the system that can then be shown to the folks writing the requirements so the best choices can be made. Of course traceability between requirements and test cases is essential in order to ensure that all of the desired functions work as designed and required.

Finally, let's not forget the people supporting your systems. It has been our experience that many software glitches are a direct result of inadequate knowledge management and sharing or employees who are afraid to act. It is very common for organizations to have only a few critical subject matter experts who may not necessarily be on duty when outages occur. Capturing essential knowledge and training sufficient support resources are vital aspects of any system support approach.

Building reliable software is not only possible; it is the only reasonable approach to creating the multitude of mission-critical systems our modern society relies upon on a daily basis. DevOps embraces many of these industry best practices and, most importantly, focuses on improved communication between the technology experts creating the software and the operations experts minding the data center day and night. Your organization will be more successful and your systems will be more reliable if you embrace these industry best practices. **{end}**

Sticky Notes

For more on this topic, go to StickyMinds.com/bettersoftware.

■ References

Matthew Heusser

Years in Industry: **15**

Email: matt.heusser@gmail.com

Interviewed by: **Noel Wurst**

Email: editors@sqa.com

Most folks agree that “just” development isn't enough to get to our end result, which is high quality delivered. The question mark is how we get there—but most of us at least agree that some part of that is testing.

“

Programming is the creative process; testing is viewed as the destructive one. I'm curious if we can move beyond that.

”

Some of the programmers will guess one thing, and some of the testers another, and you'll have a stupid argument, which requires a decisionmaker to intercede—a lot of waiting, recoding, and retesting.

Most of the teams I work with have significant opportunity to reduce failure demand, and ATDD is one way to do it—getting the software “more righter” in the first go.

The tester needs to be technical enough to say, “Hmm ... this function is getting mighty long” or “We sure are passing in a lot of variables here” or, perhaps, “Is it about time to write a unit test?”

If nobody really knows what they need until someone takes a crack and fails, then I'll probably try for some rapid prototyping method, or “tracer bullet” approach over ATDD.

For the full interview, visit
<http://well.tc/FOETA15-6>

Featuring fresh news and insightful stories about topics that are important to you, TechWell.com is the place to go for what is happening in the software industry today. TechWell's passionate industry professionals curate new stories every weekday to keep you up to date on the latest in development, testing, business analysis, project management, agile, DevOps, and more. Here is a sample of some of the great content you'll find. Visit TechWell.com for the full stories and more!

Choosing Lifecycles and Tools in Geographically Distributed Teams

By Johanna Rothman

You have any number of choices for your lifecycle if you are a geographically distributed team transitioning to agile. I do recommend a servant leader agile project manager for coordination and risk management. With people all over the world, it's difficult to coordinate the project, which leads to more risk.

Scrum is often not the best approach for geographically distributed agile teams. That does not mean the distributed team should not go agile. It just means they should not use Scrum. If the distributed team can all travel to one place so they can get trained by the same Scrum trainer together, and if they can take the opportunity to talk together to discuss what they need from the ScrumMaster, then maybe they can use Scrum, especially if they use their retrospectives well. It's a lot of responsibility for a team new to agile and new to Scrum. If you're a team new to agile and to Scrum and you try this, do yourself a favor and use a coach.

Continue reading at <http://well.tc/wJ6>

DevOps in the News

By Joe Townsend

Much like configuration management (CM), you won't hear Brian Williams talking about DevOps on the evening news or see Matt Lauer interviewing a DevOps expert on the couch of The Today Show. However, recently in IT-related news, DevOps is a huge story on the wire.

I want to start with what a writer on VentureBeat coined "An Idiot's Guide to DevOps." This article first explains what DevOps is and then goes into a Q&A session with Barton George of Dell who works on the company's cloud strategy. They cover every aspect about DevOps, from its relationship with agile and open source to the different goals of development and operations.

Continue reading at <http://well.tc/wo9>

The Pretty and the Ugly Sides of Social Networks When Job Hunting

By Rajini Padmanaban

Landing yourself in the right job and setting yourself up for a successful career progression is a combination of skill and luck. No job is cast in stone, and it is not uncommon for people to switch jobs a few times in a quest for professional and personal satisfaction to align with all of their requirements. People have dream jobs that they are always chasing, whether they are rookies in the industry or are seasoned veterans.

There is no right or wrong answer as to how many years a person should work in a position before considering a move. Despite the interviewing process, which can often be tough, people are willing to give companies a try when searching for better placements.

Continue reading at <http://well.tc/wJP>

How IT Management Professionals Reach Nobility

By Anuj Magazine

Leaders create value; managers count value. Leaders exercise influence, managers exercise power. Leaders inspire and motivate, managers simply plan, organize and coordinate.

Many articles that explore the differences between leadership and management do it in a way that undermines the importance of a manager, while projecting leadership as the superior of the two. There is often more goodness to the role of management than some of these articles really capture.

Continue reading at <http://well.tc/wJB>

Usability's Growing Role in the Development Lifecycle

By Mukesh Sharma

Apple continues to be a dominant player in the consumer hardware and software business with its soaring market share in the mobile, music, and tablet businesses. One key point that has helped Apple maintain this position is its focus on user-centric design. Apple values end-user experience highly, as is evidenced by Steve Jobs' vision of starting with a user-centric experience, then working backward by aligning technology with a usable design.

Usability has always been a part of a product development lifecycle but is often an afterthought categorized as a time-permitting exercise. Apple has changed the landscape and made the IT world rethink its usability strategy in product design.

Continue reading at <http://well.tc/woG>

Post-it Teams Up with Evernote to Take Sticky Notes Digital

By Beth Romanik

In an age when everything is digital, Post-it notes are getting with the times.

3M, the company behind the popular sticky notes, announced a partnership with personal organization app Evernote, which many people use for note-taking and document storage. The two joined forces to create software that will recognize photos of Post-its taken with a smartphone and let users read, store, and sort notes.

The software also will recognize the signature neon colors of the Post-it notes and allow users to categorize them accordingly. So, for example, you could use all pink for shopping lists, green for scheduling, and yellow for those important “Don’t forget!” messages.

Continue reading at <http://well.tc/wor>

Empirical Evidence Lacking for Cost-of-Change Curve

By Bonnie Bailey

Everybody knows that defects are more expensive to fix the later you find them. Everybody, right?

Well, back in 1976, a software engineer named Barry Boehm said defects are more expensive to fix the later they are found, and we’ve been agreeing with him ever since. His “curve” represented the successive phases of the waterfall software development life cycle on the horizontal scale, with a diagonal line traveling up and away on the vertical scale, which corresponded to the “relative cost” of fixing a defect detected in a given phase versus fixing the same defect in a different phase.

The curve was modified and enhanced over the years, and a second exponential relationship was added to it with a smaller slope designated for “Smaller Software Projects.” The diagram was further evolved in ensuing years by other members of the software engineering profession into other versions like pyramids and histograms.

Continue reading at <http://well.tc/wo7>

The Four Cs of Extending DevOps with Cloud Automation

By Sumit Mehrotra

Combining DevOps with a hybrid cloud architecture breaks down the traditional barriers between development, QA, and IT operations organizations and removes the silos that constrain enterprise development. Companies can change their software development paradigms and rapidly increase software delivery timeframes by adapting enterprise IT architectures to include cloud computing resources that enable agile development, empower software development teams with self-service, and support DevOps.

Cloud automation enables greater linkage between development and operations functions. With it, teams can avoid clumsy handoffs between the software release and deployment stages, as well as historical cultural clashes that have occurred when operations personnel felt that developers “threw code over the wall” that they then had to figure out how to operate reliably.

Cloud automation enables continuous development and delivery of enterprise applications across the entire lifecycle, making the entire continuous delivery process seamless and frictionless for developers, testers, and operations personnel. Organizations benefit from the scale and flexibility of hybrid cloud environments.

Continue reading at <http://well.tc/woW>

The Dirty Secret about Cloud Service Level Agreements

By Beth Cohen

Has your cloud service provider had an outage that affected your users or business? If so, you are not alone. Ask Netflix, Instagram, and Vine—just a few of the high-profile companies that have had major service disruptions due to cloud infrastructure failures in recent months.

If the cloud is supposed to be so reliable, can companies offset some of their losses through the cloud service provider’s service level agreement (SLA)? Based on a recent review of the fine print, I would not count on it, because cloud service providers’ SLAs are notorious for being unreadable and unenforceable.

Neither hardware nor software is infallible of course. No matter how well designed an application for reliability, failure of the underlying cloud infrastructure is always a possibility. For most consumers, cloud outages are an inconvenience, but for cloud based SaaS companies and on-line retailers, losses can easily run into millions of dollars an hour. Amazon itself suffered downtime on August 19, 2013 that lasted approximately forty minutes. That translates to an estimated income loss of \$1,104 per second. Ouch!

Continue reading at <http://well.tc/wod>

Why Even Experts and Professionals Should Use Checklists

By Naomi Karten

Checklists can help remind busy people what they need to do. Even more importantly, they can prevent people from omitting essential steps in routine activities. In his popular book *The Checklist Manifesto: How to Get Things Right*, surgeon Atul Gawande points out how easy it is to make mistakes even when carrying out simple, familiar procedures—or maybe especially when carrying out such procedures.

Gawande is talking about experts who are highly skilled and experienced yet make serious mistakes. In his book he offers numerous examples of how simple checklists can help these professionals avoid tragic errors. For example, a checklist used at eight hospitals for carrying out the basic steps in surgery led to an amazing 35 percent reduction in deaths and complications.

Continue reading at <http://well.tc/wZ3>

REQUIREMENTS REUSE

*Fantasy
or Feasible?*

**By Karl Wieggers
and Joy Beatty**



Reuse is an eternal goal for those seeking increased software productivity. People think most often in terms of code reuse, but reusing requirements also offers many advantages. The benefits include faster delivery, lower development costs, consistency across and within applications, fewer defects, and reduced rework. Reusing trusted requirements can save review time, accelerate the approval cycle, and speed up testing. From the user's perspective, requirements reuse can improve functional consistency across related members of a product line or among a set of business applications.

Even if the implementation of a software system varies in different environments, the requirements might be about the same. An airline's website might have a feature to let a passenger check in for a flight, pay for seat upgrades, and print boarding passes. The airline might also have self-service check-in kiosks at airports. The functionality for both check-in operations will be nearly identical and hence reusable across the two products, even though the implementations and user experiences might be very different. You may have noticed the same phenomenon between desktop and mobile software applications.

Reuse is not free, however. It presents its own challenges, both with respect to reusing existing items and to creating items with good reuse potential. It will likely take more time and effort to create high-quality reusable requirements than to write requirements you intend to use on just one project.

Dimensions of Requirements Reuse

Table 1 describes three independent dimensions of requirements reuse: the extent of assets being reused, the extent to which an item must be modified for use in its new setting, and the mechanism used to perform the reuse.

Dimension	Options
Extent of reuse	<ul style="list-style-type: none"> Individual requirement statement Requirement plus its attributes Requirement plus its attributes, context, and associated information such as data definitions, glossary definitions, acceptance tests, assumptions, constraints, and business rules A set of related requirements A set of requirements and their associated design elements A set of requirements and their associated design, code, and test elements
Extent of modification	<ul style="list-style-type: none"> None Associated requirement attributes (priority, rationale, origin, and so on) Requirement statement itself Related information (tests, design constraints, data definitions, and so on)
Reuse mechanism	<ul style="list-style-type: none"> Copy-and-paste from another specification Copy from a library of reusable requirements Refer to an original source

Table 1: Three dimensions of requirements reuse

Extent of Reuse

The first dimension has to do with the quantity of material that you are reusing. You might reuse just a single functional requirement. Or you might reuse a functional requirement along with any associated attributes, such as its rationale, origin, priority, and more if those are relevant to the target project. In some cases you can reuse not just the requirement,

but also associated artifacts: data definitions, acceptance tests, relevant business rules, constraints, and so on. Often, a set of related requirements can be reused, such as all the functional requirements associated with a particular feature.

In the ideal scenario, you can reuse a whole package of requirements, models, design components, code, and tests—that is, you reuse an entire chunk of implemented functionality unchanged from a related product. This level of reuse can work when the same capabilities appear in multiple applications on a common platform.

Extent of Modification

The next dimension to consider is how much modification will be needed to make existing requirements reusable on a new project. In some cases, you'll be able to reuse a requirement without any modification, as with the airline's check-in kiosk and website example. In other cases, you might reuse a requirement statement unchanged but have to modify some of its attributes, such as its priority or rationale in the new system. Most often, you'll want to modify an existing requirement to suit the new purpose. Even if you don't change the requirement, you might need to modify some of its corresponding designs and tests. An example is porting functionality from a PC to a tablet that has a touch screen rather than the traditional mouse and keyboard interface.

Reuse Mechanism

The most rudimentary form of reuse is simply a copy and paste of a piece of requirements information from another specification or from a library of reusable requirements. Copying and pasting has some limitations. You don't retain a history of where the original information came from, and it will increase the size of your specifications because you're duplicating information. It's also possible to modify a requirement copied like this even if the intent was to keep it synchronized with the original source. However, even this simple method beats no reuse at all.

In many cases, you're better off reusing existing content by referring to it instead of duplicating it. Give each object in a collection of reusable requirements a unique identifier. To incorporate that information by reference, enter the identifier for each object you want to reuse in your document. If possible, include a hyperlink directly to the reused object. If you maintain a collection of reusable artifacts properly, those links and the destination information will always be current.

An even more effective way to reuse by reference is to store requirements in the database of a requirements management tool. Depending on the tool's capabilities, you might be able to reuse a requirement that is already in the database without replicating it. These tools often retain historical versions of individual requirements, which allows you to reuse a specific version of a requirement or set of related requirements. If someone modifies that requirement in the database, the older version that you are reusing still exists. You can then tailor your own version of that requirement to suit the needs of your project without disrupting other reusers.

Figure 1 illustrates this process. Project A creates the initial version of a particular requirement. Project B decides to reuse that same requirement, resulting in both projects sharing a common version. Then Project A modifies that requirement, thereby spawning version 2. However, version 1 still exists unchanged for use in Project B. If Project B needs to modify its copy later, it creates version 3, which does not affect any other project using any version of that same requirement.

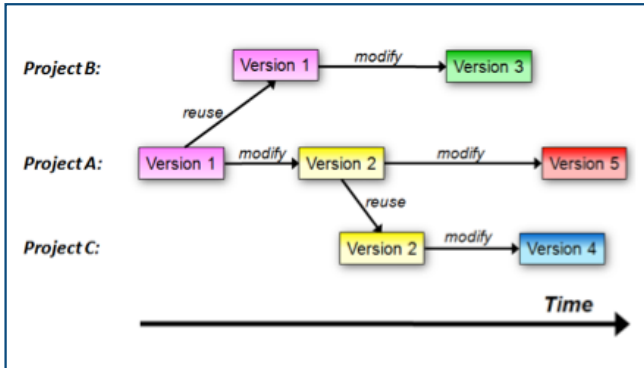


Figure 1: How a requirement can evolve through reuse in multiple projects

Types of Reusable Requirements Information

Some types of assets have broad reusability across an enterprise, including a glossary of terms, stakeholder profiles, user class descriptions and user personas, business rules, security requirements, and accessibility requirements. Table 2 identifies some other types of requirements-related assets with good reuse potential.

Scope of reuse	Potentially reusable requirements assets
Within a product or application	User requirements, specific functional requirements within use cases, performance requirements, usability requirements
Across a product line	Business objectives, business process models, context diagrams, ecosystem maps, user requirements, core product features, usability requirements, compliance requirements, certification requirements, data models and definitions, acceptance tests,
Across a business domain	Business process models, product features, user requirements, acceptance tests, data models and definitions, compliance requirements
Within an operating environment or platform	Constraints, interfaces, infrastructures of functionality needed to support certain types of requirements (such as a report generator)

Table 2: Some types of reusable requirements information

A set of related requirements in a specific functional area offers more reuse value than single, isolated requirements do. For example, there's no reason for every project team in an organization to reinvent security requirements for user logon and authentication, changing and resetting passwords, and so on. A good set of requirements for these common capabilities can be reused many times to save time and provide consistency across applications. Other groupings of related requirements information to reuse include the following:

- Functionality plus associated exceptions and acceptance tests

- Data objects and their associated attributes and validations
- Compliance-related business rules, such as Sarbanes-Oxley, other regulatory constraints by industry, and organization policy-focused directives
- Symmetrical user functions such as undo/redo
- Related operations to perform on data objects, such as create, read, update, and delete

Tools to Facilitate Reuse

Serious reuse efforts demand the use of a requirements management tool. A tool should enable the reuse of a specific requirement by sharing it across multiple projects or baselines. In this case, you'll need to consider what should take place if someone changes either the original requirement or its clones. Some tools let you lock the content to permit editing only the original instance of the requirement. This ensures that all locations where that requirement is reused are also updated with the edit. If you do want to modify a reused requirement, you'll want to employ a reuse mode that lets you change only the copied requirement without affecting other instances.

When copying a requirement that has associated traceability relationships, you might want to bring along everything that is linked to it. Alternatively, you might wish to extract just the requirement (including its children and dependent requirements) into a new project. This would be the case if the function is the same but the delivery platforms differ, as is the case with applications that run on a web browser, tablet, smartphone, and kiosk. Look for a requirements management tool that supports the reuse capabilities you have in mind.

Requirements Reuse Barriers

The most obvious barrier to requirements reuse is that the requirements developed on previous projects weren't documented. And even if you find a relevant requirement, it might be badly written, incomplete, or a poor fit for your present circumstances. The business analysts on previous projects might have used a wide variety of requirements representation techniques and conventions. It's best to adopt some standard notations for documenting requirements to facilitate reuse, such as using requirement patterns as described in *Software Requirement Patterns* by Stephen Withall. [1]

Requirements that have embedded design constraints will offer little opportunity for reuse in a different environment. Similarly, requirements that are tightly coupled to specific implementation environments, business domains, or platforms are less likely to generate reusable requirements or to benefit from an existing pool of requirements knowledge.

Another barrier has to do with ownership. If you're developing a software product for a specific customer, its requirements are likely the proprietary intellectual property of the customer. You might not have the legal right to reuse any of those requirements in a system you develop for your own company or for other customers.

Requirements Success Factors

An enabling tool for effective large-scale reuse is a searchable repository in which to store requirements information. This repository could take several forms:

- A single network folder that contains previous requirements documents
- A collection of requirements stored in a requirements management tool that can be searched across projects
- A database that stores sets of requirements culled from projects for their reuse potential and enhanced with keywords to help future business analysts know their origin, judge their suitability, and learn about their limitations

No one wants to reuse junk, so writing high-quality requirements in the first place is another success factor. Of course, that will benefit the original project for which you write the requirement, as well. Watch out for requirements that have dependencies on each other such that a whole group of requirements must be reused together. Use traceability links in a tool to identify these dependencies so people know just what they're getting into when they select a requirement for reuse.

Establishing common terminology and definitions across your projects facilitates reusability. Glossaries and data dictionaries are good sources of reusable information. Rather than incorporating an entire glossary into every requirements specification, create links from key terms to their definitions in the shared glossary.

Management should encourage business analysts both to develop high-quality components with real reuse potential and to effectively reuse existing knowledge. A culture that encourages business analysts to borrow first and create second and that invests in making requirements reusable can increase the productivity of analysts and development teams and build better products. **{end}**

karl@processimpact.com
joy.beatty@seilevel.comx



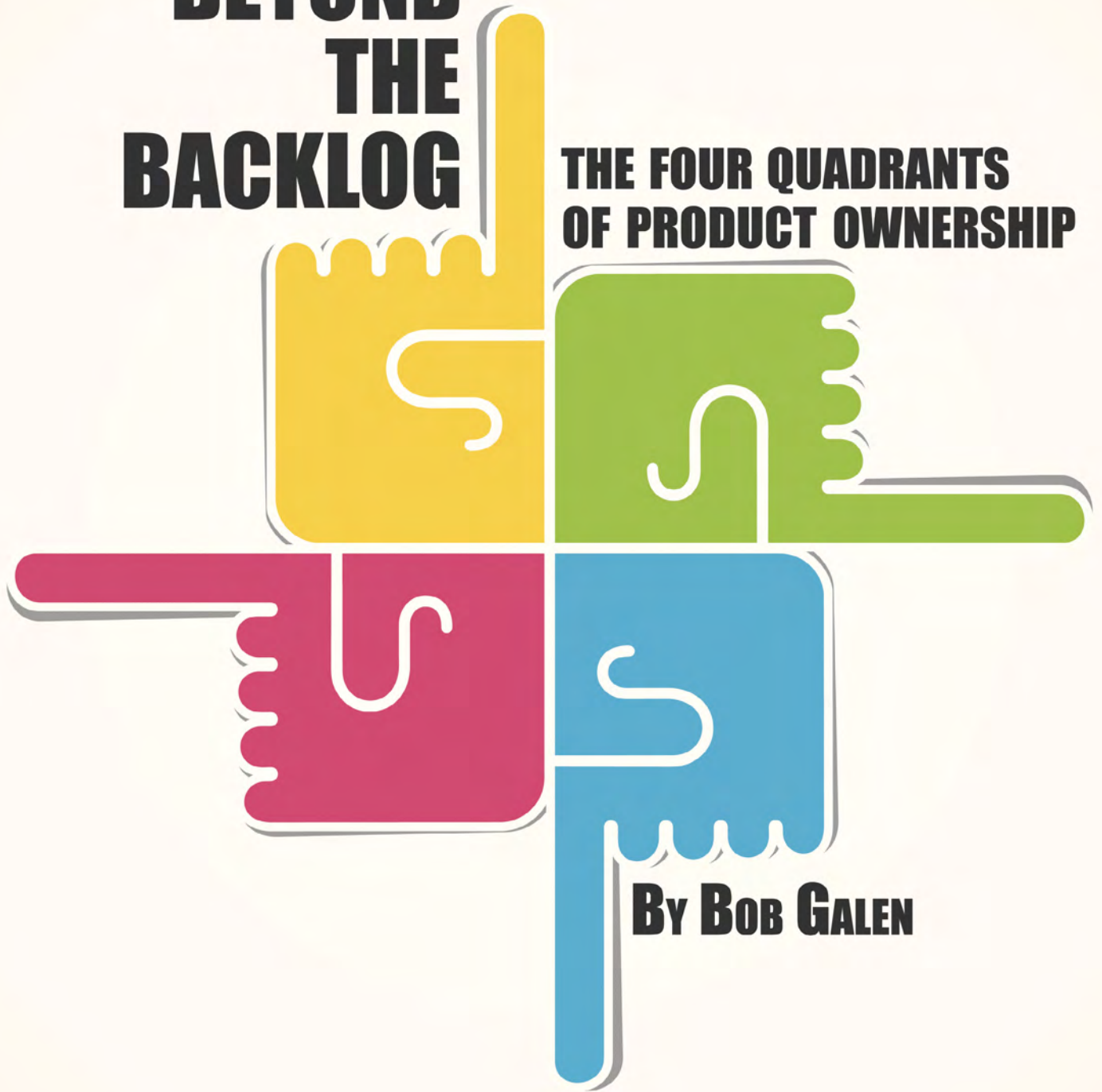
Sticky
Notes

For more on this topic, go to
StickyMinds.com/bettersoftware.

■ References

MOVING BEYOND THE BACKLOG

THE FOUR QUADRANTS
OF PRODUCT OWNERSHIP



By Bob Galen

When you hear the term “product owner,” what do you think of? And when you hear the term “product backlog,” what do you think of? Answering these two questions gives me a good hint about your level of experience with agile practices.

A common answer to the first question is that the product owner owns the backlog. An equally typical answer to the second question is usually in the form of a list of attributes such as the following:

- It’s a list of things to do (ordered by priority or business value).
- Individual work items are sized by the team.
- It is created by the product owner to be implemented by the team.

Usually, there’s an implied relationship in the answer—that is, the product owner role embraces the product backlog, and the product backlog defines the product owner role.

To be a balanced and broadly effective product owner, I’ve established a view toward four key areas a product owner must focus on: product management, project management, leadership, and business analysis. Let’s explore each quadrant in detail while discussing how these quadrants are important to the effective execution of the overall product owner role.

Quadrant 1: Product Management

The first quadrant of the product owner role is classic product management. This role is generally outward-bound with frequent communication with stakeholders and investors to define a product’s vision and roadmap. Product managers facilitate focus groups and interview customers to determine their problems and challenges that should result in defining innovative product solutions.

Product managers are viewed as product champions who are in the very best position to display the product. They understand high-level product workflows and can demonstrate critical functional scenarios that will resonate with customers. They have often worked in the problem domain, understand customer challenges, and are always searching for ways to solve those problems.

Another key role is establishing a product mission and vision. Product owners often establish a release roadmap with key stakeholders by gathering everyone’s visions and then aggregating them into a cohesive whole. These roadmaps lead to release milestones and customer commitments that require management with each product release.

There’s also a true marketing component to this role that involves pulling together functional overviews and white papers that explain the product and provide the necessary product information for the sales team. In addition to preparing sales channels for a successful product release, they’ll prepare collateral, pricing, and ROI models. These product management activities are critical to the success of the sales and customer support functions.

Quadrant 2: Project Management

Whenever I present the project management quadrant to product owners, there’s instant resistance. Most people have a picture in their heads of a traditional software project manager, and they simply don’t connect it to the product owner role.

A good place to start is by envisioning the product backlog as more than a prioritized list of requirements. A product backlog is the single most important artifact that captures the features, the work, the flow, dependencies and risks. In fact, you could view a product backlog as essentially a work breakdown structure for agile projects that requires product owner involvement. The list below is a summary of some of its key activities:

1. Enabling a team to step back from a sprint-by-sprint focus to find the most effective way to deliver on the original project goals
2. Aligning stakeholder expectations based on a team’s capacity and ability to deliver
3. Establishing early architectural and design work that forms a framework for supporting the content: not an opportunity for taking on Big Design Up Front, but more of a “lean lookahead.”
4. Embedding testing activities and strategies in the backlog, particularly in high application integration or regulated environments
5. Sprinkling milestones (rallying points, integration points, and demonstration goals) throughout the backlog that show how the team will be building functionality up toward the release
6. Ensuring that the team considers all work that is required—from the concept phase to release and into the hands of customers

These traditional project management activities are performed as part of the creation of a project charter. A product owner is tasked with instantiating or chartering the project by establishing the initial set of business goals, requirements, constraints, and delivery measures for success.

One of the central activities associated with the Extreme Programming agile methodology is called release planning. This is tightly coupled to the creation of user stories that produces a work list (or product backlog). The photograph in figure 1 represents the results of placing sticky notes on a whiteboard that I facilitated for four teams across eight sprints during a recent release planning meeting.

Before whiteboards were invented, I would use masking tape to represent each iteration on a long conference table. The result looked a lot like swimming lanes! Usually our releases took at least ten iterations, resulting in quite a few lanes spread across the table.

The next step in this process involved the team’s distributing the work, captured in user story cards, across each of the iterations. The team would engage in conversation about the most effective way to deliver the functionality, workflow, risk

mitigation, handle unknowns and dependencies, and plan integration milestones. Quite often the discussions would result in a new user story that was added to the mix. These new user stories add background (or glue stories) that elaborates on a product's features or functionality.

After an hour or so, the team produced a balanced workflow leading to a release. Because the amount of work usually exceeds the expected time frame, typically you'll enter a period of negotiation and scope tradeoff decisions with stakeholders. The result should be a bought-in, realistic release plan where user stories are collected to become the product backlog. Now, the team can begin sprinting.

Release planning merges stakeholder needs with realistic delivery expectations into envisioned, prioritized, high-value bodies of work that align with roadmap expectations.



Figure 1: Release planning board for four teams

Quadrant 3: Leadership

From an overall team perspective, a product owner must be a leader. Instead of going it alone, consider partnering with your ScrumMaster to provide effective leadership where your team needs it.

Product owners have a responsibility to establish a vision for the team's work and to set project goals. So many Scrum teams forget that sprint planning starts with a sprint goal the product owner brings to the meeting.

Beyond goal setting, the product owner needs to provide individual leadership to team members. This implies that you connect with the team to establish belonging and loyalty throughout a project lifecycle. When developing a product backlog, you need to consider your team's strengths, weaknesses, and interests. It is truly a balancing act between delivering on customer needs while maximizing and challenging your team's capabilities. A good practice is to frequently listen to your teams perspectives on the work and actively include it as you craft the product backlog.

A key component of your leadership responsibility is to be the primary communication voice for the team. Sure, part of a ScrumMaster's role includes communication, but you're the one expected to communicate with senior leadership, stakeholders, and customers. It is a good idea to frequently communicate all aspects of your team's efforts and progress. And if someone unjustly questions your team's efforts, come to the team's defense.

Quadrant 4: Business Analysis

The product backlog is typically composed of user stories, which serve as a product's requirements for the team. The stories need to be well written and developed with the team, and that's where your business analysis background comes in handy.

But it can be a challenge to write solid stories. User stories not only contain a functional description but also conditions of acceptance. As a product owner performing a business analysis role, you must communicate the value proposition and behavioral nuance you're looking for in each feature for the team.

The conditions of acceptance provide incredible benefit to the team. From a developer's perspective, they communicate key design constraints. From a tester's perspective, they define the risk-based testing strategy for each story. Ultimately, the acceptance criteria drive open collaboration and discussion centered on questions, answers, and ongoing clarification.

You'll find that business analysis activities will take place just about every day. The key is to elaborate and collaborate as a team around your requirements, stories, and backlog. An anti-pattern that I've often seen repeated is when product owners take this on by themselves. They get stuck in the perception that they "own" the backlog and need to write everything themselves. And nothing could be further from the truth.

Essentially, the product owner acts as the conductor of the backlog while engaging the whole team toward its care and feeding. Think of it as an ongoing responsibility for the team. Not only will this create better definition in the backlog and better products but it also will reinforce overall team ownership and empowerment.

Wrapping Up

Whether you're operating as an Extreme Programming customer, Scrum product owner, or other agile customer-centric role, as the product owner you have to balance your efforts across the four quadrants. While focusing your attention on the product backlog is traditionally considered to be your primary role, it doesn't represent the true depth and breadth required to perform the role well. And if you find yourself limited in skill or time to fully deliver any of the quadrants, go get some help. Well-rounded product ownership is key to your team's successful delivery of customer value. **{end}**

bob@rgalen.com

Sticky
Notes

For more on this topic, go to
StickyMinds.com/bettersoftware.

■ Further Reading

USABILITY TESTING WITH

CAN I
USE
THIS?

BY DAVID GREENLEES

THINKSTOCKPHOTOS.COM

Inspired by my wife while she was working as a usability analyst, and by “Test Mobile Applications with I SLICED UP FUN!” [1] by Jonathan Kohl, I have developed a mnemonic for usability testing: CAN I USE THIS?

Developing test ideas represents the essence of our profession. When faced with unfamiliar contexts, it can be difficult to come up with ideas. I created this approach to help foster idea generation when undertaking usability testing.

When using this mnemonic approach, the order of the letters and their corresponding ideas has no particular importance other than making an acronym that is easier to recall. So let’s break down each letter of CAN I USE THIS? into specific test ideas.

Dissecting Each Letter of CAN I USE THIS?

C: COMPARABLE PRODUCTS

A very good source of test ideas is a comparable product or products. A fabulous explanation of what’s comparable has been written by Michael Bolton. [2] These ideas can be particularly valuable for usability testing.

When testing a new online banking solution for a client, I would often refer to solutions used by other banking institutions. It is worth researching competitors and then using their solutions as a benchmark. It can be as simple as comparing the number of clicks it gets for a particular process with the number your product gets.

If you are testing a product that is new to the market and you cannot determine another that is comparable, don’t get hung up on the product as a whole. Break it down to comparable features, functions, and themes.

Other Cs worthy of note:

- **Compatibility:** While issues with compatibility may not be technically defined as usability issues, I have heard many a user complain about the difficulty in navigating on a website, filling out a web form, or issues relating to a specific version of a browser.
- **Context:** Be aware of the project environment and technical environment. How quickly are your deadlines approaching and how important are they? How much importance is put on usability issues in your project? What platform is the product being developed for? These are all questions that will assist your testing.

A: ACCESSIBILITY

Usability and accessibility are two different things, but they do complement each other. The accessibility of a software product is extremely important. For example, government software products have to conform to legislation that governs how accessible software needs to be to its users.

The intended user base also will drive the accessibility requirements. A website for a particular blind society would be a good example of one that should meet certain accessibility needs. Can a user expand the size of the text? Are there audio recordings for areas of lengthy text? These are just some of the ques-

tions you will likely ask yourself while testing for accessibility.

There’s one more A I’d like to mention:

- **Accuracy:** When conducting testing or evaluating the software while others are, it’s a good idea to note overall user accuracy. How many mistakes did you or the users make? Were these mistakes serious and to be avoided at all costs? This information can help drive product improvements.

N: NAVIGATION

Navigation is typically the most important when assessing usability.

There are many different elements to navigation, including movement through a particular flow and being able to know where you are at any point in time. For example, when testing websites, make sure there are no orphaned pages that require the use of a browser’s back button.

Think about how your users will be utilizing common computing devices. Does the product work best via mouse clicks or do the links need to be emphasized for finger taps on a mobile device? And if a user wants to leave the current web page, is there a simple way to return to the site’s home page?

Other Ns you may wish to consider:

- **Needs analysis** is a formal process that focuses on the human elements of product requirements. This is a valuable exercise to undertake when attempting to understand a user’s needs and can be an excellent way to create usability test ideas.
- **Nielsen, Jakob:** [3] If you’re looking for more ideas and lessons about usability, research Jakob Nielsen. Known as the guru of web page usability, his work spans more than twenty years of research and is well worth investigating.

I: INTUITIVE

Intuitive is often used as a synonym for easily learned. For example, stakeholders often ask if the product is intuitive or whether a product is intuitive enough.

We all learn in many different ways, so testing intuitive behavior is difficult. Instead, focus on intended user demographics and use this information to drive your perspective on how to test the product. While this seems obvious, you would be surprised how many times it is forgotten during the early design phase of a product. Identifying intuitive product issues late in a project lifecycle usually results in scheduling disasters!

U: USERS

This is where the whole process begins, and focusing on user demographics is critical. Just like the importance of the one ring in *The Lord of the Rings*, this is the one letter prompt to rule them all. How can you test the usability of a product if you don’t know who the users are? Short answer: You can’t. And knowing the users is merely the beginning—you need to understand them, as well. How large are those input buttons used on mobile applications for weightlifting programs? I hope

those big fingers can tap them easily!

Know your user, understand your user, and practice testing personas. [4]

S: STANDARDS

You might not be aware of usability standards, but included in the International Organization for Standardization's requirements for the ergonomics of human-computer interaction is ISO 9241-11, [5] which attempts to explain how to identify necessary information for specifying or evaluating usability in terms of measures of user performance and satisfaction.

The best thing a standard in usability can provide is consistency in use of products and features. It is far more enjoyable and easier to use a product that is consistent with its previous version or other similar products. Microsoft and Adobe have gone to great lengths to ensure that applications that are part of a suite operate similarly. This consistency can reduce technical support costs and improve user satisfaction.

E: EMOTIONAL RESPONSE

This is a very important element of usability. It can make or break a product, especially in the world of mobile applications.

Imagine your friend has just downloaded a new application to his mobile device and his emotional response is one of anger and frustration. Would you then download the same application? Would your friend recommend it to others? I think not.

A good practice is to map out the product and its features against areas of use, and within those areas, the types of emotions that users would commonly have. For example, bidding on online auction sites can create user anxiety, especially if it's with an unfamiliar seller. To test the usability of that process, you may wish to simulate that anxious feeling, which could assist in your evaluation of its usability.

You will never be able to anticipate all emotional responses, but at the very least keep the idea in the back of your mind while testing.

Other Es worthy of note:

- **Efficiency:** How efficient is the product at achieving what it was designed and developed for? Achieving a goal is one thing, but achieving it in a fast, clear, and efficient manner is another. You can compare the time it takes users to complete a task with other users and other products.
- **Errors:** Along with accuracy, how many errors do you or the users make while trying to execute a particular process or flow within the product? Are the errors handled well by the product? Are they easily understood by the user?
- **Ethnography Research:** This is a qualitative research design that explores cultural phenomena. It has become more common in user experience design methods in recent times.

T: TRUNK TEST

[Taken from Steve Krug's *Don't Make Me Think: A Common Sense Approach to Web Usability*] [6] A trunk test

specifically relates to the navigation of a web page, but it can also be used as a tool within other products by simply adjusting the user elements you are assessing.

The title comes from the analogy of being blindfolded and stuffed into the trunk of a car, being driven around for a while, then being freed. The task is to assess where you are by asking a series of questions. For the usability test, choose a web page within the site you're testing and print it. Hold the paper at a distance of approximately an arm's length away and try to locate the following:

- Site ID (What site is it?)
- Page name (Where am I?)
- Sections (Are the site's main sections outlined?)
- Local navigation (What can I do next?)
- Where you are (Is there a "You Are Here" label?)
- How you can search (generally for larger sites)

Once you are satisfied that all the elements can be located and your questions can be answered at a glance, you can be reasonably comfortable that the web page has a workable and user-friendly navigation system.

Other Ts you'll want to consider:

- **Training:** How much training are the users going to need and how complex will that training be? Training can require a significant amount of energy, time, and money.
- **Tools:** There are many sophisticated tools that can assist with your testing. For web applications, I have used Firebug's and Chrome's element inspection to make quick updates to HTML tags in order to change page headings, button names, and so on.

H: HEURISTIC EVALUATION

A heuristic evaluation is an inspection method that assists you in identifying potential usability issues so they can be addressed as part of the iterative process. [7] While testing, you examine the product against a pre-defined set of usability principles (also known as heuristics).

The most common set of usability heuristics was developed by Jakob Nielsen. [8] Nielsen's ten usability heuristics should provide you with all the coverage you need to get started:

- Visibility of system status
- Match between system and the real world
- User control and freedom
- Consistency and standards
- Error prevention
- Recognition rather than recall
- Flexibility and efficiency of use
- Aesthetic and minimalist design
- Help users recognize, diagnose, and recover from errors
- Help and documentation

Yet another H to consider:

- **Hallway Testing:** This method is particularly good early in the design process to highlight critical issues. This pseudo-random approach is analogous to picking

people to test your product as they pass by you in the hallway. Select a few random people to use the product and only provide them with basic instructions. This may be the best way to truly test how new users will react to the product under development.

I: INSTRUCTIONS AND HELP TEXT

Product instructions in the form of user documentation or help are a valuable element to any product. It is vital to assess if this content is usable and has a positive impact on a user's experience.

When instructional aids are too complex to understand, the result can be worse than a product without them. By putting yourself in a user's shoes and demographic, you'll be able to assess if the instructional content meets the needs of the intended audience.

S: SATISFACTION

Once you have spent some time testing the product, stop and reflect. How satisfied are you with your use of the product? Would you want to use the product if you were a potential user?

Particularly with e-commerce sites, a user's level of satisfaction with the buying process can be just as influential as the level of satisfaction with the actual product being bought. Satisfaction drives return traffic and the building of positive reputations. Alternatively, a bad user experience can spread like wildfire doing more harm than good.

Questions

The question mark is officially a part of this mnemonic. Questions are the cornerstone of our profession, and, as testers, we should be constantly questioning. Make sure any questions you encounter are asked—and subsequently answered—by the appropriate people. When using this mnemonic, use the question mark as a reminder that there are no stupid questions. Even the simplest of questions can save extensive amounts of energy, time, and money.

Summary

This is in no way an exhaustive list of areas to focus on when testing a product for usability. The use of a mnemonic is simply a tool for test idea generation and can be expanded or simplified as needed. Take it with you from project to project and use it as a reminder of potential usability issues even if your focus is not solely on usability. **{end}**

xtremedmg@gmail.com

**Sticky
Notes**

For more on this topic, go to StickyMinds.com/bettersoftware.

■ Further Reading



ACHIEVING A PROJECT'S BENEFITS AND GOALS

HOW CHANGE MANAGEMENT CAN HELP

BY KRIS IRWIN

Businesses and organizations today face rapid change like never before. Due in large part to the growth of technology, this constant state of transition requires a targeted methodology and a dedicated team to maximize benefits and minimize negative impact on those affected. Change management serves this function by adapting a group of people to a change. According to management consulting firm McKinsey & Company, only 30 percent of change management initiatives are a success. This successful minority likely employs several key tools and practices harnessing change management.

Put the Individual First

People are the key—if they don't know what to do, realize how to do it, or care to do it right, the project will not achieve its desired outcomes. The trick is to get everyone engaged, motivated, and committed to the project objectives and benefits. Change management addresses not only the needs of an organization to be ready for the changes coming but also the individuals affected by those changes, such as employees, customers, and suppliers. Even if you deliver the project successfully and aligned with the organization's strategy, if the people are not ready, it is all for naught.

At the Association of Change Management Professionals' conference last spring, Dan Markovitz, founder and owner of business consulting firm TimeBack Management, led a session called "Forget about change. Try problem solving." He said that instead of focusing on the change and associated change activities, we should instead focus on the idea that if the people being affected are engaged in the decision-making and implementation phases of a change, they are more likely to adopt them. [1] Markovitz argued that this approach of involving people in solving business problems could be used throughout a project. He was also quick to point out that the type of methodology used to engage people does not matter as long as people are truly engaged.

Identification of Project Benefits and Goals

Have you ever led a project where the benefits were not clearly defined up front? How about being assigned the project manager role where the scope, timeline, or resources of that project are limited and you simply do not agree with them? What if the project success metrics are not identified and documented?

As a project manager, you work with many people—including project team members, stakeholders, and the general population—who will be affected by the changes the project will introduce once it is completed. As you determine the amount of change, you may come to the conclusion that the assessment, execution, and reinforcement phase activities need a dedicated person to ensure the project meets its success metrics. This is when bringing a change manager on board will benefit the project as a whole.

Change management professionals help identify and clarify a project's benefits and goals. They also serve as a third-party mediator and risk monitor. A key part of a change manager's role includes communicating with all stakeholders in order to identify those individuals who don't yet understand the project's goals and objectives, who may resist the project, and who may need to be more involved than others.

A change manager engages team members in the initial project planning stages when the project outcomes, goals, and business benefits are outlined. Identifying this infrastructure while gaining consensus in a group setting early in a project's lifecycle goes a long way in obtaining employee buy in. Part of this identification process involves determining specific success metrics for the department or for individuals. This, in turn, will help demonstrate whether the changes the project introduced will be successfully adopted.

There are a few steps that a change management team can follow to help identify and address any concerns for a group. Take a hypothetical example of a marketing team impacted by a new system that should result in numerous changes to their business processes. As a first step, a designated change management lead would host a one-on-one meeting with the

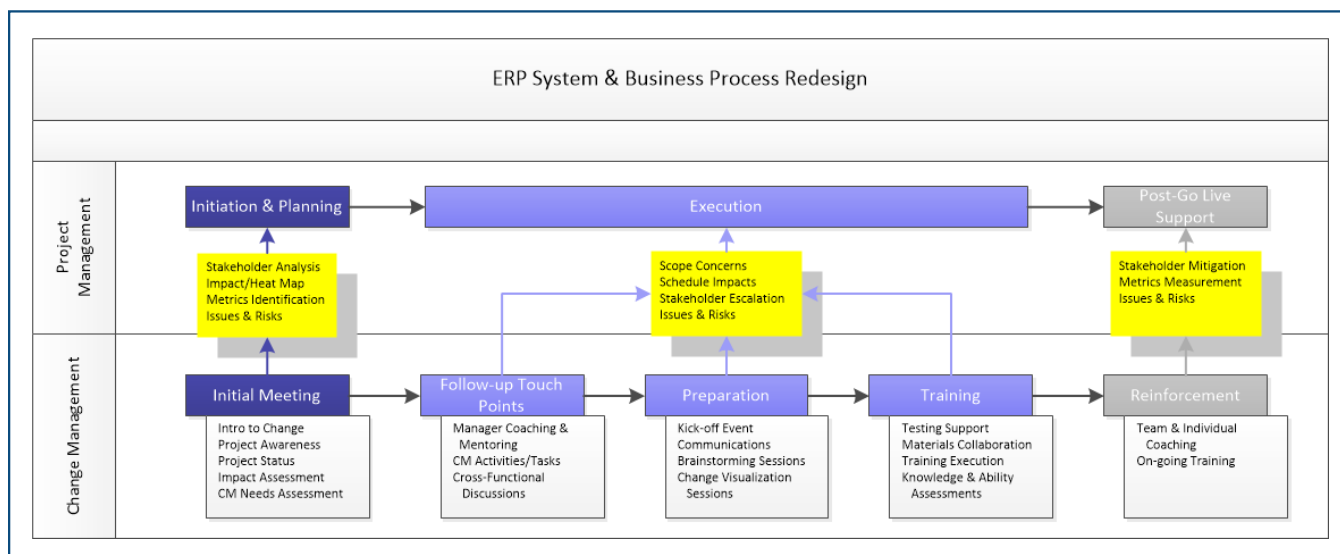


Figure 1: Hypothetical marketing stakeholder engagement process

marketing director to touch base on her current status. Afterward, there should be follow-up touch-point meetings with the director along with additional activities with the department employees to help build awareness and desire around project goals. Figure 1 shows this process taking shape over time. At any touch point during the change management process, collaboration and escalation to a project lead or team member may occur to help mitigate any challenge that may arise. This helps address risks before they become issues and should proactively help address any issues that could lead to resistance.

Communication Basics

Ongoing communication is the key to a project's success. When interviewing stakeholders, the first challenge a project or change manager encounters is creating basic awareness of the project. If team members do not know about the project's benefits and goals, how can they even begin to think about the changes they will need to make in order to achieve those goals?

Identifying and establishing the “what’s in it for me?” (WIIFM) objective requires knowing the project objectives, goals, and benefits; how the project impacts people; and what conversations will need to be had with key stakeholders to lead them through the change acceptance process.

Employees and their representing stakeholders will require ongoing, consistent communication to understand the project, benefits, and goals. Stefanie Simon, a training instructor at change management research company Prosci, has argued that you can communicate a message “five to 1,499 times” because using a ridiculously high number serves to remind us that it is important to communicate until the recipient of your message gets it. [3] The goal of communication is to reach people—through email, online newsletter, website updates, social media sites, one-on-one meetings, business unit or company-wide meetings, and any other means available. You’ll want to use many of these methods until you can confirm that people have heard, comprehend, and buy into the message.

Communication deliverables that a change management lead produces includes project one-pagers, newsletters, and flyers. Whether you want to build awareness or directly address resistance to change, each of these communication vehicles can be very effective. A change management communication plan can be an invigorating change of pace because it lies outside the typical project status updates.

Extras Can Mean a Lot

A change manager is often put in the position of communicating with stakeholders across the organization, and doing so partially independent of the project planning and decision process. Because of their unique position, some change management leaders actually refer to themselves as part therapist because stakeholders feel comfortable venting and escalating their issues to them. This insider role gives change managers the opportunity to identify what other benefits not addressed by the project process or systems are needed.

A change manager’s candid conversations with stakeholders about project objectives, status, and next steps often can lead to detailed questions. If it is determined that nothing can be

done, the change manager works with stakeholders to create positive messages to communicate with the team, acknowledging that some concerns cannot be addressed. Identifying these concerns earlier in the process helps to mitigate resistance and feelings of exclusion for that group of people.

These frank discussions with stakeholders help build a trusting relationship and project buy-in. Stakeholder alignment and relationship building represent key ingredients of the people side of change management.

After a Project Goes Live

It is a common misconception held by company executives, managers, and even project team members that once a project goes live, their roles are done. It is often a mistake to think that training is complete once the project is live. Usually it’s the opposite—the training is just beginning! Ongoing training, coaching, and mentoring after launch is a must if the project’s benefits and goals are to be met. Human nature is to revert to what is familiar and comfortable. Very few people like to be outside of their comfort zones for very long. This can lead to the tendency of team members to revert to the old way of doing things or create a work-around, which is not in line with the change process just introduced. To circumvent this, it is critical for managers to be in place to perform follow-ups and reinforcement activities to ensure people continue to embrace the new way of doing things and to improve on their performance. Not planning for continued re-enforcement of change is similar to releasing a software product without planning for ongoing maintenance.

Enabling Future Success for Change Management

The final key phase in change management is reinforcement. By incorporating change management activities and resources into a project and leveraging team members’ knowledge, skills, and deliverables, a change manager can build a framework for change in future projects. If the use of these new processes and tools are not reinforced as part of future change initiatives, a company will not fully maximize potential benefits. A change manager adds great value to the company by arming future project leaders with all the tools, assets, and best practices developed on the previous project. This framework also can provide the backbone for a company to create a change management office (CMO). Successful and ongoing organizational change management requires individuals to build their own internal competencies at leading change from wherever they are in the organization. A successful CMO is cross-functional, with team members representing all aspects of a company or organization and serving as advocates to create and foster a diverse community of change agents. {end}

kirwin@acmebusinessconsulting.com

Sticky Notes

For more on this topic, go to StickyMinds.com/bettersoftware.

■ References

Parasoft Announces Parasoft Service Virtualization

Parasoft announced the general availability of the new Parasoft Service Virtualization, an open, automated infrastructure for continuous testing. With the latest release, teams can provision simulated test environments and launch associated test scenarios via a readily-accessible API. This enables fully-automated continuous testing, which is a critical component of continuous delivery and continuous release. Even if the team needs to launch a myriad of different test environments, each of which runs an extensive set of test scenarios, all the necessary provisioning and test execution can be fully automated.

<http://www.parasoft.com/jsp/resources/virtualization>

CollabNet's CloudForge Enhanced With TeamForge 7

CollabNet, an enterprise cloud development and federated ALM company, announced the availability of TeamForge 7 in its CloudForge cloud development platform. CloudForge helps developers manage and build websites and mobile/cloud/web applications, and rapidly develop and deploy business software.

Built for the enterprise, it offers a fast and practical path to grow and scale organization-wide, through a flexible role-based framework that makes it easy for developers to use and for administrators to scale from a few to thousands of users. CloudForge with TeamForge 7 includes adaptive planning boards for agile and hybrid product and release planning, inline editing to enhance productivity and team ramp-up time, and usability improvements that include intuitive navigation, web drag and drop, and a more powerful search tool to improve the user experience.

<http://www.collab.net>

CA Technologies Introduces New Version CA LISA Release Automation

CA Technologies introduced a new version of CA LISA Release Automation. The new version optimizes a large number of complex enterprise-class releases across the application life-cycle, from development to testing to production. The solution is a critical piece to implement DevOps methods as it can reduce software release times by simplifying and standardizing an organization's entire release process, and minimizing errors. The continuous application delivery solution takes release automation to the next level and provides an agile and collaborative delivery chain, helping enterprises innovate and succeed.

When the operations team is bogged down by repetitive release tasks, it becomes increasingly difficult to complete periodic application updates within a scheduled timeline. Application release automation can optimize the entire process and enable quality assurance teams to directly handle the task of releasing software, thus allowing the service delivery teams to provide more strategic services.

<http://www.ca.com/us/default.aspx>

SmartBear Software Releases SoapUI 4.6

SmartBear Software, a software quality tool company, used by more than one million developers and testers worldwide, is re-inforcing its support for REST API testing with SoapUI 4.6. The new version of the open source tool, with more than one million users worldwide, directly answers user requests to simplify REST API testing using SoapUI. The latest version also aligns the SoapUI community with a more pragmatic REST approach.

This is the first major release to include a wide range of testing capabilities for REST APIs used in a majority of public APIs for web applications. Together, they shorten learning curves, improve time-to-market, and help to easily identify more bugs. Essentially, the new version allows testers to begin testing their REST-based applications faster and more effectively. SoapUI 4.6 is the first product from a new SoapUI development team and signals the first of a range of new releases that aims to simplify REST API testing.

<http://www.soapui.org>

Xebialabs Unveils XL Release

Xebialabs, a delivery acceleration software company, unveiled XL release, an enterprise-class release coordination product. XL release enables teams to integrate release management and execution for the first time while providing a path to fully automated delivery pipelines, getting features and enhancements to your customers and end users faster.

XL release allows all team members to collaborate effectively and simply on application releases—to plan, execute, and track both manual and automated tasks from code drop to end-user. Integration with Xebialabs' application release automation software Deployit provides a complete solution to deliver higher quality software faster.

<http://www.xebialabs.com>

Pentaho Corporation Announces Pentaho Business Analytics 5.0

Pentaho Corporation announced the availability of Pentaho Business Analytics 5.0, a completely redesigned data integration and analytics platform. Pentaho 5.0 provides a full spectrum of analytics for today's big data-driven businesses, regardless of data type and volume, IT architecture, or analysis required. The new, modern interface simplifies the user experience for all those working to turn data into competitive advantage.

Pentaho 5.0 enables analysts to create cleansed, architected blends directly from diverse big data sources with the ease of use and real time access demanded in today's agile analytics environments. Additionally, with the big data integration capabilities in Pentaho 5.0, analysts can confidently blend all data in near real-time and immediately analyze the results.

<http://www.pentaho.com/5.0>

WANdisco Announces New Release of SVN MultiSite Plus

WANdisco, a provider of high-availability software, announced the next release of SVN MultiSite Plus, the company's performance, scalability, and continuous availability solution for large enterprises using Subversion. SVN MultiSite Plus, announced earlier this year, incorporates the latest enhancements to WANdisco's patented replication technology and delivers availability, performance, flexibility, and scalability improvements over standard SVN MultiSite. SVN MultiSite Plus also allows administrators to manage the entire implementation from one location and add and remove Subversion repositories and servers on the fly without any downtime.

One of the key new features included with SVN MultiSite Plus 1.1 is enhanced notification. With enhanced notification, when a server or network outage occurs or a server's disk or memory usage approaches capacity, automatic administrator alerts are generated. This simplifies management during the failover and recovery process and enables administrators to address server capacity issues before bottlenecks occur.

<http://www.wandisco.com/subversion/multisiteplus>

Seapine Software Releases TestTrack 2013.1

Seapine Software announced the release of TestTrack 2013.1, the newest version of its tool suite for managing product development. The 2013.1 release brings the power of the Windows, Macintosh, and Linux TestTrack clients to the browser with a modern, unified web application.

TestTrack 2013.1 replaces its three previous web clients—TestTrack Pro web client, TestTrack RM Reviewer, and TestTrack TCM Test Runner—with a single, new unified application. The improved, modern user interface allows software developers and testers to create and review requirements, work with issues, and execute and track tests from their web browser. Also included in the TestTrack 2013.1 release is the ability to visualize project information in dashboards created with the Seapine ALM Reporting Platform. Teams can use the dashboards to quickly review and share team metrics and project status data from all TestTrack projects.

<http://www.seapine.com/almnewfeatures.html>

FP Complete Launches FP Haskell Center

FP Complete, a developer of commercial Haskell software tools and services, announced the launch of FP Haskell Center, the world's first commercial Haskell IDE and deployment platform. Haskell is the new way to make software that will change the economics of software development. FP Haskell Center enables developers, researchers, and students to experience the power of Haskell as a convenient web service.

<http://www.fpcomplete.com>

Syncfusion, Inc. Updates Orubase

Syncfusion, Inc., a .NET component vendor, today released a major update to Orubase, its framework for developing cross-platform, hybrid mobile applications. The release comes just a month after Syncfusion's release of Essential Studio for JavaScript.

At the forefront of the Orubase update are several JavaScript controls for visualizing data and support for HTML and JavaScript-based web applications. With the latest updates, various editors, gauges, and charts are available as JavaScript controls. Developers have the choice to work with ASP.NET MVC or JavaScript when designing hybrid mobile apps.

<http://www.syncfusion.com>

SOASTA Inc. Launches TouchTest Lite

SOASTA Inc., a cloud and mobile testing company, launched TouchTest Lite, a fully cloud-hosted free version that enables users to start testing mobile apps in as few as five minutes. New features in TouchTest provide mobile developers and testers with additional control, management, and continuous testing capabilities for native, hybrid, and mobile web apps while improving collaboration and access to diverse teams. New partnerships and integrations expand the reach and speed of mobile testing with popular commercial and open source tools.

Unlike traditional testing solutions that rely on manual testing, device emulators, or fragile external object recognition, TouchTest utilizes SOASTA's in-app approach to mobile app test automation that captures and plays back all actions and gestures with correct motion and velocity on real devices.

<http://www.soasta.com/free>

Juniper Networks Makes Juniper Networks Contrail Available

Juniper Networks announced the availability of Juniper Networks Contrail, a standards-based and highly scalable network virtualization and intelligence solution for software-defined networks (SDN). Contrail is a production-ready SDN solution that is based on stable and proven networking standards. Contrail creates a virtual network, enabling seamless integration between physical and virtual networks while providing service providers and enterprises with a solution that is simple, open, and agile.

Network and cloud service providers are under tremendous pressure to develop, deploy, and monetize new services, but most services are difficult to reconfigure or adapt to changing user demands. Contrail empowers these customers to quickly bring new services to market by providing a smooth transition with automatic provisioning and dynamic service chaining in cloud environments that run on x86-based servers.

<http://www.juniper.net/us/en>

Fusion-io Delivers Open APIs for Flash-Aware Application Acceleration

Fusion-io announced new milestones in the development of flash-aware applications. The Fusion-io Atomic Writes API contributed for standardization to the T10 SCSI Storage Interfaces Technical Committee is now in use in mainstream MySQL databases MariaDB 5.5.31 and Percona Server 5.5.31 as well as the upcoming Percona Server 5.6. Fusion-io will be contributing its nonvolatile memory key-value interface to flash, NVMe, to the Open Compute Project, and has posted the first flash-aware Linux-kernel virtual memory Demand Paging Extension to GitHub for community testing.

The latest releases of MySQL databases MariaDB and Percona Server deliver the option for flash-aware operation. These two popular MySQL distributions are the first enterprise applications to ship using Atomic Writes, streamlining the software stack by replacing the need to write twice to maintain atomicity or database ACID compliance.

<http://www.fusionio.com>

Accusoft Enhances Comprehensive Imaging Toolkit with New Features

Accusoft, a provider of document, content, and imaging solutions, announced version eighteen of its ImageGear Professional software development kit (SDK) in three editions for DLL 64-bit, DLL 32-bit, and ActiveX development. The ImageGear Professional SDK gives developers the tools they need to add high-performance imaging capabilities, such as scanning, compression, annotation, image editing, and color-image processing to their applications.

Applications built with ImageGear Professional can now load and save image files with Unicode symbols in their file names, preventing the need to rename files. They can update metadata in TIFF, JPEG, JPEG 2000, and JPX image files with Unicode symbols in their file names, and load, display, and save Unicode text strings in vector files (SVG, DWG, DXF and DWF). The new version features many smaller enhancements, such as support for new PDF and PS file formats, and improved annotation "burn in" for images containing alpha channels.

<http://www.accusoft.com/ig-pro.htm>

Hansoft Introduces Ready-Made Integration with Git

The Hansoft-Git integration works out of the box with minimal effort for IT professionals setting it up. The integration allows you to link Git commits with items in Hansoft with only one click. The integration is bi-directional, and allows the user to tag Git commits on submission or after the commit has been made.

<http://www.hansoft.com/git-integration>

WSO2 Debuts WSO2 App Factory Cloud-Enabled Enterprise DevOps Platform

WSO2 debuts WSO2 App Factory, which enables continuous cloud delivery through a shared, self-service, multi-tenant and

elastic enterprise DevOps platform. WSO2 App Factory allows teams to collaboratively create, develop, and deploy enterprise applications into cloud environments—without the delay of server provisioning. Robust lifecycle management and governance ensure compliance with IT, corporate, and government policies.

WSO2 App Factory enables the integration of popular software development lifecycle (SDLC) tools, such as Apache Subversion (SVN), Git, Jenkins, Maven, Redmine, Clover, FindBugs, and JUnit. At the same time, the open design of WSO2 App Factory also allows the integration of any SDLC toolset with widely adopted tools, such as Atlassian JIRA and Bamboo. Additionally, App Factory integrates with existing user repositories via LDAP or Microsoft Active Directory.

<http://www.wso2.com>

Skytap Introduces Enterprise-Ready Hybrid Cloud Capabilities for DevOps Professionals and Teams

Skytap Inc., a provider of self-service cloud automation solutions, introduced new enterprise-ready hybrid cloud capabilities that allow IT operations and development and test (dev/test) professionals to quickly and securely extend on-premise environments to Skytap Cloud. The new features, including Network Address Translation (NAT), Command Line Interface (CLI), and Single Sign-On (SSO), enable dev and test teams to more quickly and easily deploy virtual environments in Skytap Cloud, while providing the security, visibility, and control functionality required by enterprise IT.

Skytap's new hybrid cloud features provide on-demand advanced networking and programming capabilities for development and testing in Skytap Cloud, while maintaining the visibility and control features mandated by enterprise IT.

<http://www.skytap.com>

WSO2 Launches WSO2 App Factory

WSO2, an open source application development software company, has launched the WSO2 App Factory, which enables continuous cloud delivery through a shared, self-service, and elastic enterprise DevOps platform. WSO2 App Factory enables teams to collaboratively create, develop, and deploy enterprise applications into cloud environments—without the delay of server provisioning.

WSO2 App Factory provides an open source, open standards approach to cloud-enabled enterprise DevOps. It is built on—and integrates with—the open source WSO2 Carbon enterprise middleware platform and WSO2 Stratos, a platform as a service.

WSO2 App Factory allows for the integration of popular software development lifecycle (SDLC) tools, such as Apache Subversion (SVN), Git, Jenkins, Maven, Redmine, Clover, FindBugs, and JUnit. At the same time, the open design of WSO2 App Factory also allows the integration of any SDLC toolset with widely adopted tools, such as Atlassian JIRA and Bamboo.

<http://www.wso2.com>

FAQ

expert answers to
frequently asked
questions

by Bob Payne

bob.payne@lithespeed.com

Ryan Olivett

ryan.olivett@lithespeed.com

How Can I Develop the Ability to Collaborate?

Most companies have a static learning culture where management looks outside the company to fill skill and knowledge gaps. New employees are hired along with consultants and trainers to augment hiring needs. Agile and lean companies, on the other hand, tend to focus internally. Ideally, companies foster a learning culture by providing mentors and encouraging employees to experiment with new activities outside their current roles. One of the ways we help companies develop a learning culture is with the A3 process. This management philosophy helps employees take ownership of their career development, improves collaboration within and across teams, and increases the amount of feedback employees receive from their peers and managers.

The A3 process, developed by Toyota, is a tool to encourage empirical learning. [1] Employees identify problems, analyze root causes, and propose solutions for continuous improvement. Employees typically write on a paper no larger than A3 size (16.5 by 11.7 inches) to easily share ideas with others, obtain feedback, and revise frequently to reflect new information. The book *Managing to Learn: Using the A3 Management Process to Solve Problems, Gain Agreement, Mentor and Lead* by John Shook and Jim Womack is an excellent primer on the A3 process. We apply a variation of the A3 process to career development that we call the personal A3. The process helps employees think about their strengths and passions to develop a career roadmap that suits them but still aligns to organizational priorities. Employees select mentors and partners to help facilitate the process and share with their peers based on three main values: ownership, collaboration, and feedback.

Ownership: Instead of traditional career development, the personal A3 process allow employees to seek out a career path that is not bound by their current role or dictated by their manager. In short, they are able to find a path that matches their strengths and passions. Agile and lean working environments require that employees take a higher level of ownership over their work than traditional environments. Why not encourage ownership of career progression and skills development as well?

Collaboration: The personal A3 process facilitates collaboration, so knowledge becomes a shared resource. The person works directly with a mentor, partners, and peers throughout the process to define skills, values, goals, and more on the A3s. These people help identify ways they can collectively help the individual bridge the gap between the current state and the desired future career state, utilizing internal capabilities by sharing knowledge and pairing with other teammates on new activities or projects.

Feedback: The personal A3 process establishes a learning feedback loop that allows employees to learn about themselves as they ponder their strengths and passions while collaborating with others. Receiving feedback is essential for learning. Mentors and partners provide feedback throughout the process. As an individual's career roadmap is implemented, progress toward action items and specific goals are measured. A3s are revised frequently to reflect learning and progress.

The personal A3 is not the only way to facilitate ownership, collaboration, and feedback, but it is a simple and effective tool for doing so. Additionally, it socializes the more general A3 process so that employees can apply it to common business problems. Employees who have ownership and autonomy over their work and careers, who are encouraged to collaborate with their peers, and who learn continuously through feedback tend to be more satisfied with their jobs and are ultimately more productive workers. {end}

Sticky
Notes

For more on this topic go to
StickyMinds.com/bettersoftware.

■ References

PMO 2.0: Rebooting the IT Project Management Office

How to ensure that PMOs benefit organizations to provide oversight of a company's projects, project standards, and compliance.

by **Tony McClain** | tony.mcclain@geneca.com

"We all know there's resistance to a PMO [by project managers]. There's this image of a governing body being bureaucratic or creating red tape." This was an assessment from national Project Management Organization (PMO) thought leader Tom Nodar way back in 2005, quoted in a *Computerworld* article published under the headline "PMOs Viewed as Unneeded Bureaucracy." [1] Unfortunately, not much has changed in terms of the IT industry's perception of this important role, and there's a huge need for an effective PMO.

The PMO's mixed reputation is not entirely undeserved. Along with the technology industry itself, the function has proliferated in the past two decades and encountered substantial growing pains. A recent poll by the consulting firm PM Solutions Research shows 87 percent of companies in 2012 had PMOs, a growth of nearly double since 2000. [2]

In the rush to remain competitive, this tremendous growth undoubtedly led many organizations to form PMOs without entirely thinking it through. Project managers were promoted from other positions, the wrong people were hired from outside, and adequate training was often nonexistent. These recruits applied cookie-cutter solutions to problems and lacked the requisite passion for the role, resulting in an unproductive combination.

Charged with developing, implementing, and refining project standards and compliance across entire organizations, PMOs identify ways to make projects more efficient and consistently successful. Think of the PMO as the project management equivalent of the US Office of Weights and Measures. An effective PMO creates uniformity, predictability, and order in complex, project-based industries such as IT. Done poorly, a PMO does exactly the opposite, resulting in confusion where none existed before.

With organizations like the Project Management Institute, there's no shortage of substantive resources in the marketplace today that can help IT projects succeed by offering PMO best practices, training, and accreditation.

But all the training and certificates in the world can't push a project across the finish line if it has the wrong organizational champions. That's why I'm proposing a reboot of the PMO. In my experience, there are three important factors that determine whether a project succeeds or fails.

1. Obtain buy-in: Frequently, a project is doomed from the start because the right stakeholders do not support the project or its goals. Executive stakeholders must be completely aligned on a project's priorities and strategic direction and be totally

committed to implementing the necessary organizational and cultural changes. Stakeholders also must provide a clear vision and ensure that everyone can understand and believe in the story.

2. Conduct self-reflection: PMOs spend most of their time overseeing the work of others, so it's not surprising that they frequently overlook the importance of self-reflection when it comes to

their own roles. To successfully oversee projects, PMOs must also be aligned with the strategic direction of the organization and committed to their role in its victory.

They also must provide better visibility with ongoing projects, remain flexible, and collaborate with managers on the ground to determine what's working and what is not. These iterative feedback loops between the PMO and those in the trenches ultimately result in better overall processes for the organization. PMO leaders also must be able to reinforce and articulate the vision for a project's success.

3. Consider intangibles: PMOs are not about process for the sake of process or injecting whiz-bang technology as a way to justify budgets. Perhaps more than anything else, PMOs are about fostering the right talent and culture within an organization so that successful projects are a routine byproduct of the process.

To do so, PMO leaders must influence the right behaviors and remove impediments to success. This allows a project team to adjust and make changes when they need to and encourages all team members to participate. Once everyone is on the same page, these behaviors become woven into the fabric of the organization.

When teams take ownership of outcomes, they will always

"When teams take ownership of outcomes, they will always do what it takes to ensure a project succeeds."

do what it takes to ensure a project succeeds. With this combination of attitude and culture, the entire organization—and the customer—wins.

The same PM Solutions survey found that 40 percent of organizations without a PMO expect to add one within the next year. That's why, more than ever before, organizations are looking to PMOs to provide not only the steady hand necessary to meet today's IT challenges but also to reimagine the traditional metrics around these three factors. If your PMO has become overweight, ineffective, or out of touch with the needs of the organizations it serves, seriously consider a version 2.0 overhaul. If your PMO is or even perceived as being overweight, ineffective, or out of touch with the needs of the organizations it serves, seriously consider a version 2.0 overhaul. **{end}**

Sticky Notes

For more on the following, go to StickyMinds.com/bettersoftware.
■ References

WANTED! A FEW GREAT WRITERS!

Better Software magazine is always looking for authors interested in getting their thoughts published in a leading online magazine focused in the software development/IT industry. If you are interested in writing articles on one of the following topics, please contact me directly:

- Testing
- Agile methodology
- Project and people management
- Configuration management

I'm looking forward to hearing from you!

Ken Whitaker
Editor, *Better Software* magazine
kwhitaker@sqe.com

index to advertisers

3i Infotech	http://www.3i-infotech.com/content/services	15
ASTQB	http://www.astqb.org	23
Cognizant	https://fastest.cognizant.com/webapps/home	19
OpenMake Software	http://www.openmakesoftware.com	27
Ranorex	http://www.ranorex.com/whyBSM	Inside Front Cover
STARCANADA 2014	http://starcanada.techwell.com	2
STAREAST 2014	http://stareast.techwell.com	1
Telerik	http://www.telerik.com/test-studio	Back Cover

Display Advertising
advertisingsales@sqe.com

All Other Inquiries
info@bettersoftware.com

Better Software (ISSN: 1553-1929) is published six times per year: January/February, March/April, May/June, July/August, September/October, and November/December. Back issues may be purchased for \$15 per issue plus shipping. (Subject to availability. Volume discounts available.) Entire contents © 2013 by Software Quality Engineering, 340 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.

