

Dr. Dobb's Journal

December 2012

Next

ALSO INSIDE

Getting DevOps Right:
The Lay of the Land >>

From the Vault:
The Shrinking Expanding World of CI >>

Microsoft's JavaScript Move >>

What Is DevOps?

Dr. Dobb's Journal

CONTENTS

December 2012



FEATURES

7 What Exactly Is DevOps?

By Neil Garnichaud

The latest trend in software development is the culture of DevOps, in which developers partner with operations staff to assure that software runs with a minimum of problems.

11 Getting DevOps Right: The Lay of the Land

By Scott Ambler

The DevOps movement offers the potential for an important bump in both professionalism and productivity in the IT marketplace. But, as with all movements before it, there is the danger that DevOps will be misunderstood and misapplied.

GUEST EDITORIAL

5 Microsoft's JavaScript Move

By Dino Esposito

Windows 8 is making JavaScript a first-class app platform.

14 From the Vault: The Shrinking Expanding World of CI

By Andrew Binstock

Continuous integration is undergoing something of a renaissance as continuous delivery (and its sidekick, DevOps) begin to find adoption in many enterprises. Simultaneously, as this February 2012 editorial notes, the number of viable CI packages is shrinking quickly.

3 Letters

By you

Readers comment on Windows 8, Oracle's stewardship of Java, and the poor man's design tool.

16 Links

Snapshots of the most interesting items on drdobbs.com including Git, assert statements, and an exploration of new features in WPF 4.5.

More on DrDobbs.com

2012 Jolt Awards: Mobile Tools

The best mobile development tools.

<http://www.drdobbs.com/joltawards/240012517>

Windows 8: The Most Confused OS Launch Ever

After making a courageous and daring bet to rework its entire product line and push itself in a completely new direction, Microsoft's disorganized product launch threatens to overwhelm any chance of success.

<http://www.drdobbs.com/windows/240009572>

Migrating from Subversion to Git and the Lessons Learned

How Atlassian moved a code base of 21,000 files with more than 47,000 commits on its best-selling product from Subversion to Git without stopping the development workflow.

<http://www.drdobbs.com/240009175>

Developer's Reading List

New books on C++11, parallel programming, CoffeeScript, requirements, and more.

<http://www.drdobbs.com/tools/240009592>

Discriminated Unions

Unions that depend on a separate field to determine access to members can be useful in some applications — but they require careful use.

<http://www.drdobbs.com/cpp/240009296>

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

Mailbag

Microsoft, Oracle, and MindMaps

Windows 8

In response to our editorial on the Windows 8 launch (<http://www.drdoobs.com/windows/240009572>), we received several comments:

You wrote “However, when I turn my slate into a real working device, rather than a browser or entertainment platter, I have to forgo the flashy new look and revert back to the Windows 7 look and feel that’s buried under the covers. In fact, I do much of my work there. The problem for many consumers, if recent articles in the mainstream press are accurate, is that almost no one seems to know that the Windows 7 look and feel can be enabled for desktops and tablets.” Please tell us how to get Win 8 to look like Win 7!

— **Bernar Liengme**

Andrew Binstock replies: “Click or Tap on the tile entitled ‘Desktop.’ That’s your desktop page. Put your various app icons on it and you’re back working in a Win7 environment.”

I have to say, good riddance to Microsoft and their high-handed business practices and insane profits. Old Bill Gates has more than enough money, as does anyone else at or near the same level. Nothing like re-inventing the same thing over and over again and calling it something new.

— **John Savage**

Andrew Binstock replies: “We received several such replies to my editorial, which was strictly about the launch of Windows 8, not about the product as such nor even Windows in general. I’m not sure there is any relationship between the ex-CEO’s wealth to anything I wrote. As to Microsoft ‘Reinventing the same thing over and over again and calling it something new,’ there is no basis whatsoever for this assertion. The very product whose launch we’re discussing has a completely original and new UI, designed by Microsoft.”

Oracle’s Stewardship of Java

I was interested to read your recent article on the last two years of Oracle’s stewardship of Java (“Two Years Later: A Report Card On Oracle’s Ownership of Java” <http://www.drdoobs.com/jvm/240008729>).

Whilst I thought that many of your points were well-taken, I’d have to disagree with your assessment of the state of Oracle’s community involvement and the JCP.

I’m the JCP Executive Committee member for the London Java Community, one of 2 Java User Groups who are represented on the JCP EC, the other being the huge Brazilian user group, SouJava. In the roughly 18 months that we’ve been on the committee, we’ve seen a lot of change. What we haven’t seen is any evidence of the undefined state that you refer to. Instead, Oracle and the other members of the EC have shown a real commitment to reform, to transparency and to updating the procedures of what was still very much a pre-Internet era stan-

IN THIS ISSUE[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

Microsoft's JavaScript Move

Windows 8 is making JavaScript a first-class app platform.

By Dino Esposito

The JavaScript language has had at least nine lives. It made its official debut in 1996 as part of the Netscape browser and has happily survived the first 15 years of the Internet era. JavaScript was designed to be a simple language to help authors make HTML pages more attractive. Instead of being a full-fledged programming language, it was just easy to use — and worked very well. Since then, JavaScript has gone through ups and downs. The low point came in the early 2000s, when the first version of ASP.NET was released with the express purpose of limiting developers' exposure to HTML and JavaScript. A few years later, with the advent of Ajax, JavaScript began its comeback.

Although JavaScript used to live within the boundaries of a web-browser engine, smarter and especially faster engines have been created recently that can be hosted outside the browser, giving JavaScript a much broader scope (think Node.js, for example).

Today, JavaScript is a critical skill for developers. And although it's still easy to get acquainted with, it isn't trivial to master its more powerful present incarnation.

JavaScript And ECMA-262

Many people who should know better still perceive JavaScript as the

language of the browser and are skeptical about its wide adoption. In reality, the JavaScript language relies on a solid standard defined by the ECMA-262 and ISO/IEC 16262:2011 papers.

So what's all the fuss about?

It's important to understand that "JavaScript" is a generic term. ECMA-262 is the recognized standard and its many implementations. Of course, a programming language's implementation makes sense only if it takes place in a runtime environment where instructions can be compiled and run. In turn, a runtime environment needs to be hosted somewhere within the context of an executable program. At present, there are several possible scenarios. The most common one is a browser that hosts a scripting engine based on an ECMA-262-compliant language. A concrete example of this would be Google Chrome (the browser) hosting V8 (the scripting engine) and consuming JavaScript (the ECMA-262-compliant language). So, essentially, JavaScript is the name of a dialect based on the ECMA-262 standard. It would be even more precise to say that Google Chrome targets ECMA-262 edition 5. Firefox, which is equipped with Gecko as its scripting engine, also targets ECMA-262 edition 5. Internet Explorer uses its Chakra engine and a dialect named JScript. Adobe Flash hosts a dialect known as ActionScript, and Acrobat Reader hosts a language still called

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

JavaScript that targets ECMA- 262 edition 3. Finally, Opera uses ECMA-Script, which is a strict implementation of the standard.

As you will see, confusion about JavaScript goes deeper than the name.

The JavaScript Dialect

A programming language's dialect is subordinate to its standard language. This subordination produces slight differences (mostly extensions) to the standard. For example, JScript offers some additional commands to support access to ActiveX objects and the local computer. This support isn't in the ECMA-262 standard but is supported if you use the dialect. The same can be said for ActionScript.

So what about the JavaScript dialect? It's a sort of sub-standard managed by Mozilla. JavaScript standard follows its own versioning, which indicates which edition of the ECMA standard is being targeted. JavaScript, JScript, ActionScript, and other permutations rely largely on the same standard and can be considered the same.

In 2002, Microsoft shipped the first version of the .NET Framework and included an ad hoc compiler for an adapted version of JScript called JScript.NET. It's safe to say that JScript.NET never conquered the masses: There's no tooling for it today, not even in Visual Studio.

We've seen this pattern too many times to consider it a mere coincidence: Microsoft will produce a great idea but a not-so-great first-time

implementation of that idea. Using a Web-friendly language to develop applications for the new .NET Framework was an excellent idea. However, it was challenging to target much more than just the elements in the host document. As a result, Microsoft came up with a managed language that provided direct access to the .NET Framework classes. This allowed use of the syntax of JScript to author, say, Windows Forms applications — however, using the Web paradigm centered on HTML and CSS to build the same Windows Forms application was out of question.

With Windows 8's release, Microsoft is now making JavaScript a first-class platform for building applications: It wisely decided to allow for the classic Web paradigm. Developers use HTML5 to build layout, CSS3 to style it, and quick-and-dirty JavaScript (or, more precisely, ECMA-based script) to manipulate page elements. Developers also have access to a variety of system-provided libraries that offer UI widgets and tools to access Windows 8-specific capabilities, such as live tiles. Microsoft's full embrace of JavaScript in Windows 8 is indeed quite late, but it's most certainly quite welcome.

— *Dino Esposito is a frequent Dr.Dobb's contributor on Microsoft developer technologies.*

[Comment](#)

**VeriSign® SSL,
now from Symantec.**
More features. More protection.

[Get more details now ▶](#)

 **Symantec.**
Confidence in a connected world.

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

What Exactly Is DevOps?

The latest trend in software development is the culture of DevOps, in which developers partner with operations staff to assure that software runs with a minimum of problems.

By Neil Garnichaud

Many organizations find themselves in a challenging situation with the rise of cloud-based Web applications that allow for quick releases in response to issues or requests coming in from the user community. Responsiveness to their user base is the goal of every software development shop, but it can put a strain on the functional teams within the organization. This strain often results in more defects and consequent disruption to the team. DevOps attempts to solve this problem by developing a partnership between Development and Operations (hence, the name). In this structure, the development team supports operational requirements such as deploy scripts, diagnostics, and load and performance testing from the beginning of the cycle; and the operations team provides knowledgeable support and feedback before, during, and after deployment.

DevOps is the direction in which many software development teams are going. They have to, given the pressures organizations are under to produce higher-quality code faster with less time available for QA. This is a new environment and many developers will need to adjust if they want to prosper. With timelines compressed, the walls separating development, QA, and production are barriers to agility. DevOps at-

tempts to break through those walls. Now, team-playing skills are as important as technical skills. So, too, is a singular focus on the end-user experience and how that is affecting the business. Rather than a new set of tools or organization, DevOps is a new culture and process. It's development, QA, and operations working together to expedite development and problem resolution.

Why Developers Should Want DevOps

DevOps is good for developers. There are three principal reasons a developer would want to work in a DevOps-oriented organization:

- A better quality of life. Developers working in DevOps-mode receive fewer calls in the middle of the night to resolve production issues. That's because they see issues before they become catastrophic problems due to an orientation of proactive monitoring rather than reactive alerts.
- Pride of ownership. In a traditional software process, once software is developed, it's "thrown over the wall" to QA, which later throws it over another wall to production — so what the end-user

IN THIS ISSUE

[Guest Editorial >>](#)

[What Is DevOps >>](#)

[Getting DevOps Right >>](#)

[Continuous Integration >>](#)

[Letters >>](#)

[Links >>](#)

[Table of Contents >>](#)

ultimately sees might be quite different from what the developer wrote. But under the DevOps model, what you write goes live because you continue to have visibility and access to the code even after it goes to QA and production. In other words, developers own the delivery of the code from creation to implementation.

- More relevant work. Developers, like most human beings, get greater satisfaction from work that has relevance in the real world. Because developers in a traditional organization are isolated, they often work on simulated problems in made-up user scenarios — and they only find out that these approximations were wrong when something breaks. In a DevOps model, scenarios are real. Environments are load tested, for example — before they're put into production — to see if they work correctly. Another example is that test scripts are, themselves, tested for realism by being deployed in the production environment, not just test labs. Sharing these test results with developers gives them the opportunity to see how their code performs under real-life conditions.

What “DevOps-Ready” Means

Perhaps your organization has already adopted the DevOps model. Three questions can help clarify where you stand on the DevOps curve:

- Do you, as a developer, have access to troubleshooting information in real time?
- Does your production environment use tests and other tools from the development team to validate that the production environment is working?
- As a developer, do you view the networking team as your partner?

The diagram illustrates the DevOps Toolchain as a continuous flow from Development to Production. At the top, the 'urban{code}' logo is displayed. Below it, a cloud labeled 'Cloud-Ready' sits above the main title 'DevOps Toolchain'. The toolchain is represented by a horizontal line with an arrow pointing right, passing through several stages: Development (with a person icon), Package Repo (with a green cylinder), Build (with a factory icon), DEV (with a blue server rack), TEST (with a green server rack), and PROD (with an orange server rack). Below the toolchain, four key components are highlighted in black boxes with white text: 'uBuild' (Enterprise Class Build Automation), 'uDeploy' (Application Deployment Automation), 'uProvision' (Private Cloud PaaS Engine), and 'uRelease' (Release Coordination). A blue 'Learn more' button is located at the bottom right of the diagram.

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

If the answers are “no,” you’re not there yet. Here are some things that can be done to improve the situation. Let’s start with your tools. While DevOps is more about culture and process than organization, tools can help enforce best practices — specifically, the sharing of troubleshooting information across silos. That requires adding more instrumentation in your software to see how your software is performing in QA and production, not just in development. This is code that traps errors, checks system parameters, reports function timeouts, and returns other values during program execution, which it then writes to log files.

In a siloed environment, developers often won’t see these log files again once the code is released into production. In a DevOps world, developers have visibility to the files regardless of where the software is run — in development, QA, or in production. Not only do defects get fixed faster, but the same defects are less likely to reappear in future releases — making development, itself, faster and more responsive to the business. That brings agile quality to agile development.

Break Old Habits

DevOps is also about breaking old habits — like the natural tendency to focus on software bug counts as a measure of quality. Fixing a single bug will not provide much leverage for creating bug-free software faster. A better measure is process bug counts. In other words, where is the process broken that led to the bug in the first place? For example, is the code that developers are building on their local machines somehow different from the code that gets deployed to QA or to production? Or is the code behaving differently in one environment because there is something present there that is not present in the other environments?



dtSearch®

Instantly Search Terabytes Of Text

- 25+ fielded & full-text search options
- dtSearch’s own file parsers **highlight hits** in popular file & email types
- Spider supports static & dynamic data
- APIs for .NET, Java, C++, SQL, etc.
- Win / Linux (64-bit & 32-bit)

“Lightning Fast” – *Redmond Mag*

“Covers all data sources” – *eWeek*

“Returns results in less than a second”
– *InfoWorld*

www.dtSearch.com

Fully-Functional Evaluations

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

Unless code versions are tightly synched across environments, and unless the environments themselves are tightly synched, it is hard to tell whether an issue is a logic problem, a data problem, an environment problem, or something else. This is another case where tools can enforce consistency — as in automated deploy to push the same code to multiple environments all at once.

Partners or Finger Pointers?

Probably the most important adjustment developers will need to make is in their day-to-day interaction with other team members. Do developers address software issues proactively (such as by monitoring operations logs daily) or do they wait for problems to come to them through support? And when there is a problem, how is it resolved? Are team members partners or finger pointers?

A lot depends on leadership — whether management is preaching the DevOps vision, leading by example, providing the necessary training and support, and rewarding developers for team contributions, not just technical prowess. DevOps requires an orchestra leader. If that job isn't filled yet where you work, maybe you should apply. Selling a DevOps environment is about understanding what's important to management: Is it moving faster? Is it moving higher quality? Is it about developers being more accountable for their code? All these things come about by way of a DevOps environment.

— *Neil Garnichaud is the Host Solutions Business Manager at SmartBear, responsible for product development and software development.*

[Comment](#)

What Happened!?

Web Site Response Time



Learn how you can save time and money with “New User Experience Monitoring”

Webinar: **How to Elevate Your Web Site Performance**

Register Now!
Date: December 12, 2012

 **SMARTBEAR**

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

Getting DevOps Right: The Lay of the Land

DevOps means different things to different stakeholders, but the basic components are the same.

By Scott Ambler

During the past year or two, there has been much ado about DevOps in the media. The number of DevOps voices has risen to a cacophony and, in tandem, the level of confusion among listeners is rising, too. The DevOps movement offers the potential for an important bump in both professionalism and productivity in the IT marketplace. But, as with all movements before it, there is the danger that DevOps will be misunderstood and misapplied. This article, and several that follow on, will try to cut through the confusion by providing coherent, no-nonsense advice.

Let's begin with some definitions. First, I use the term "production" throughout this article in two ways. When I use the IT-phrasing "release into production," I also mean to imply "release into the marketplace" when the context deals with a commercial product. When I use the word "production," I mean a combination of both operations and support (sometimes referred to as the "help desk"). Many organizations treat these as two separate concepts, although some combine them. "DevOps" is a portmanteau word for the combination of development and operations. Development in this context includes all activities that

occur before a solution is released into production, from formulating the initial concept through project initiation through construction to deployment. Operations in the DevOps context includes all post-deployment activities; that is, the "production stuff," which includes both operation and support of deployed solutions.

Defining the term DevOps is easier said than done because there are several complimentary viewpoints to consider — from each of the major DevOps stakeholders. Depending on whom you talk to, you're going to get a different definition of what DevOps is all about. The DevOps stakeholders, and their viewpoints, are:

- Developers, particularly experienced Agile developers, talk about DevOps in terms of a continuous flow of delivery into production, potentially several times a day.
- Operations professionals often view DevOps as promoting a more effective relationship with development teams, both throughout the development lifecycle as well as once the solution is in production. Experienced operations people also realize that their in-

IN THIS ISSUE

[Guest Editorial >>](#)

[What Is DevOps >>](#)

[Getting DevOps Right >>](#)

[Continuous Integration >>](#)

[Letters >>](#)

[Links >>](#)

[Table of Contents >>](#)

ternal processes, often based on ITIL (<http://is.gd/lfKIMQ>) or ITSM (<http://is.gd/ASQECa>), need to be streamlined so that they are in a better position to collaborate with development teams.

- Support professionals, sometimes called help desk professionals, view DevOps similarly but with a support twist: They want to work with development teams to ensure that their needs are understood and met before a solution is released into production. They also want to ensure that there is an effective escalation process in place to handle change requests (including defects) once the solution is in use.
- Senior management views DevOps as an effort to increase the overall efficiency of the IT department by streamlining how everyone works together.

Disciplined DevOps

Now let's look at Disciplined DevOps. Figure 1 depicts a before-and-after picture of what Disciplined DevOps strives to achieve. In many organizations today, the development and operations groups struggle to collaborate effectively, with process and organizational barriers erected between them.

Development teams will deploy irregularly — “fast” teams putting out one or two releases a year with the occasional patch as required to address a production problem.

The operations department, in turn, will push change requests, including defect reports, back to the development organization. The two organizations work together just enough to ensure that these activities are successful, but there is significant room for improvement. Disciplined DevOps introduces those improvements by promoting strate-

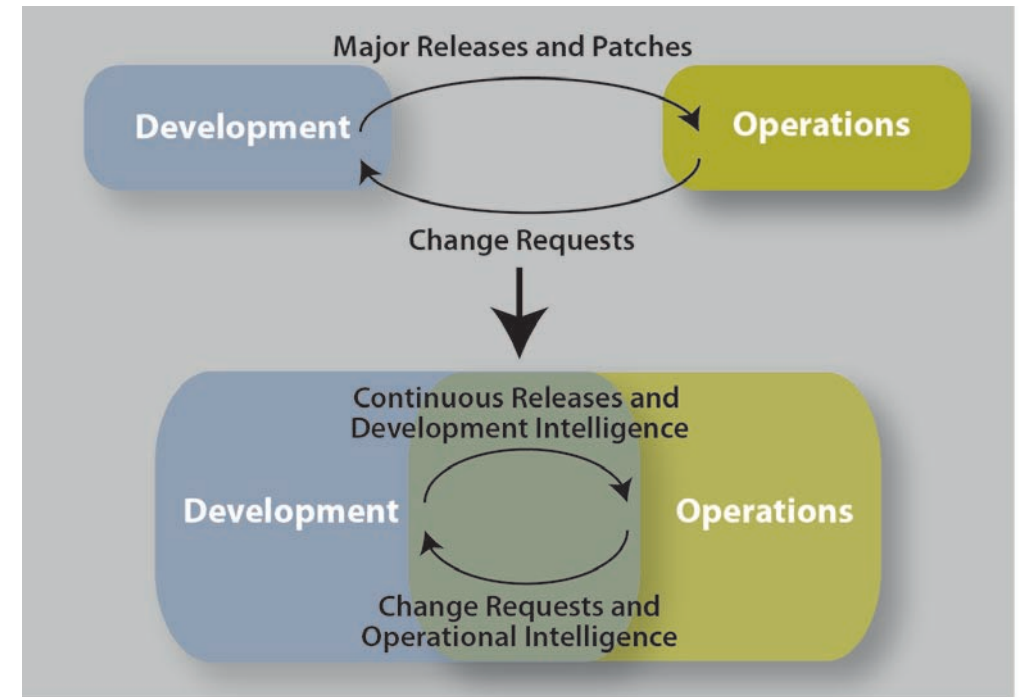


Figure 1: Closing the DevOps gap.

gies to increase the collaboration between development, operations, and support staff; introducing continuous delivery practices to development; introducing new organizational structures within IT; and adopting business intelligence technologies and techniques to support both development and operational intelligence, which in turn, supports improved IT governance. The difference between a disciplined approach to DevOps and, if you will, an undisciplined one is that the disciplined approach takes a holistic view that considers the desires of all DevOps stakeholders and doesn't focus on just one viewpoint.

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

To succeed with Disciplined DevOps, you need to address the 5Ps of improvement: People, Principles, Practices, Products, and Process. These issues are listed in priority order. People and the way that they interact

“People and the way that they interact together are the primary determinant of success in any IT endeavor”

together are the primary determinant of success in any IT endeavor, and Disciplined DevOps clearly requires people to rethink their skillset, how they define their roles, and how they work with others.

IT organizations adopting DevOps need to rethink the underlying principles that guide their decision making; for example, adopting a principle of being more reactive to business will motivate them to adopt ways to release into production more often.

Organizations will need to adopt practices such as continuous integration, continuous deployment, operational intelligence, collaborative support, and many more that may be new to them if they're going to make a go of DevOps. New products, including development tools, business intelligence tools, and operations monitoring tools, may need to be adopted. Finally, process frameworks such as Disciplined Agile Delivery (DAD), which bake DevOps strategies into the development process, as well as updated versions of ITIL or ITSM, will need to be considered as well.

Misconceptions

There are common ways that organizations seem to be running aground with DevOps. I'm concerned about the mistaken view that “cloud=DevOps,” which seems to be growing in popularity. While adopting cloud technologies can make some aspects of DevOps easier, it is only one of the 5Ps (in this case, product). Similarly, the tool-driven messaging from some vendors, and some open-source communities, is also a bit worrisome. New tools are part, but only part, of the DevOps picture. A third problem, mentioned earlier, is being overly focused on one DevOps stakeholder viewpoint. Particularly common is adopting a development focus due to the obvious coolness, and potential for improved productivity, of continuous delivery practices. In this case, the problem is focusing on only the practices portion of the 5Ps.

These misconceptions, the ensuing problems they can cause, and more-detailed advice on rectifying them, will be discussed in future DevOps articles.

— *Scott Ambler spent more than 30 years working with IBM Rational, where he helped clients adopt and adapt Agile technologies. He now consults in that area. He is a long-time contributor to Dr. Dobb's.*

[Comment](#)

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

From the Vault

The Shrinking Expanding World of CI

Continuous integration is undergoing something of a renaissance as continuous delivery (and its sidekick, DevOps) begin to find adoption in many enterprises. Simultaneously, the number of viable CI packages is shrinking quickly.

— DDJ

By **Andrew Binstock**

Of all the best practices enunciated by the Agile movement, perhaps no other makes as much sense in practice as continuous integration (CI). It is predicated on the simple theory that if you rebuild the project every time source code is checked in, developers can be informed immediately of changes that break the build. The shorter the loop between the break and notification to the developer, the more likely the developer will still remember the details of the changes, hunt down what's wrong, and get it fixed. This approach is easy to understand, straightforward to implement, and very effective. If you're not using continuous integration, you're working under a substantial handicap.

A secondary, but important, benefit of CI is that by building the product regularly, the organization has in theory a working (if still incomplete) binary that it can use to demo the software if the need arises. (Without warning, the CIO might want to see where things stand.) Moreover, the usual challenge of discovering build problems — one of the final stages in waterfall-like methodologies — is moved up the cycle, so that the effects of changes on the builders are known all the way along. This step removes all the frantic patching and rewriting that occurs in non-CI shops when a project is first tested.

The next step in CI, called "Continuous Delivery" after the Jolt Award-winning book that came out in 2010 (<http://is.gd/bxuxTU>), takes CI

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

even further. It contends that the process automation of CI should be extended to delivering and installing the apps and running tests on the installed app. This added step gives the development organization an additional level of comfort. They can automate installation and verification with each new build, and know that the product subset they can deliver on any given day is completely ready to go should the client want to see it. The need for tighter integration between development and operations caused by the automation of continuous delivery is what forged the term “DevOps.” It is supposed to mean a closer relationship between developers and operators (a historically wide and sometimes acrimonious divide), but in fact, everyone seems to have different definitions of what DevOps really means.

Given the new emphasis on CI and continuous delivery, you’d think that the industry would be offering all kinds of new tools — somewhat akin to the sudden profusion of cloud technology vendors. But in fact, the reverse is happening. The high end is consolidating around two or three principal products, and the low-to-middle end is being dominated by two OSS solutions. The numerous other OSS CI products of the past have disappeared entirely or are simply awaiting word of their impending doom.

At the high end, there are IBM Rational BuildForge, ElectricCloud’s Electric Commander, and UrbanCode’s DevOps Platform. These products regularly handle the build, test, and deploy stages of codebases greater than 20 million LOCs. In fact, that kind of line count is their sweet spot. In the next tier are mid-range products such as Atlassian’s Bamboo. There then comes a very large gap as we enter the domain of CI servers for SMBs and enterprise departments. Historically, these were served by open-source tools, especially Cruise Control. Other competitors, although somewhat less capable than Cruise Control,

were Continuous, Hudson, LuntBuild, and others. Today, these are all more or less shrinking (LuntBuild seems to have ceased activity entirely) and being replaced by Jenkins (formerly Hudson).

Hudson was, for years, gaining ground on the other products listed here due to its ease of installation and numerous plugins. When it was acquired by Oracle as part of the Sun purchase, there arose a dispute over the Hudson name. Most developers, including the principal developer, decided to fork the project and rename it “Jenkins.” Hudson was then donated by Oracle to the Eclipse Foundation, where it is primarily maintained by Oracle contributors and staff from Sonatype (the Maven company). While the bulk of the new activity is occurring in the Jenkins fork, Hudson is undergoing a top-to-bottom cleanup. Which version you prefer depends mostly on personal choice as the differences between them are salient mostly to diehards within the respective communities. Either one will serve you well. And between them, they are clearing out all other competitors in the low to lower-middle strata of the market.

It’s not often that we see a market’s mission expand, while the number of its constituent products contracts. Nonetheless, I expect that both trends will continue. Continuous delivery will reach farther into development process automation and the number of companies who can keep up with the new demands will continue to shrink. As a result, it makes sense to take time to choose your CI platform with particular care. CI builders can be difficult to swap out if you made the wrong choice.

— *Andrew Binstock is Dr. Dobb’s Editor in Chief.*

[Comment](#)

IN THIS ISSUE[Guest Editorial >>](#)[What Is DevOps>>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters>>](#)[Links>>](#)[Table of Contents >>](#)

This Month on DrDobbs.com

Items of special interest posted on www.drdobbs.com over the past month that you may have missed

THE GIT REVOLUTION IS HERE

The movement from centralized to distributed VCS is accelerating. Enterprises and tool vendors are catching on and catching up with what open-source developers have been doing for a while. And at the front of the parade is Git.

<http://www.drdobbs.com/tools/240009161>

HOW EXPENSIVE IS FREE?

With modern technology, we all stand on someone's shoulders — and often many sets of shoulders.

<http://www.drdobbs.com/embedded-systems/240010825>

WHAT'S NEW IN WPF 4.5

A ribbon control, new dispatcher methods, and delayed data bindings head the list of new features in the recent update to Windows Presentation Foundation. Here's how to use them.

<http://www.drdobbs.com/windows/240009614>

ASSERT STATEMENTS SHINE LIGHT INTO DARK CORNERS

Let's explore the consequences of checking that invariants actually hold when they should — a state of affairs that, in principle, should always be true.

<http://www.drdobbs.com/cpp/240012746>

HOME THERMOSTAT MANAGEMENT FROM THE CLOUD

Arrayent's technology was recently embedded in an Internet-enabled thermostat from Hunter Fan. Mike Riley investigates.

<http://www.drdobbs.com/cloud/240012615>

AMD'S BOLD ARM SERVER GAMBIT

By combining 64-bit ARM processors with server-side technology, the company that led the x86 architecture into the 64-bit world is hoping to reinvent the data center and give itself new life.

<http://www.drdobbs.com/architecture-and-design/240012549>

IN THIS ISSUE

[Guest Editorial >>](#)[What Is DevOps >>](#)[Getting DevOps Right >>](#)[Continuous Integration >>](#)[Letters >>](#)[Links >>](#)[Table of Contents >>](#)

Dr. Dobb's

Andrew Binstock Editor in Chief, Dr. Dobb's
andrew.binstock@ubm.com

Deirdre Blake Managing Editor, Dr. Dobb's
deirdre.blake@ubm.com

Amy Stephens Copyeditor, Dr. Dobb's
amy.stephens@ubm.com

Sean Coady Webmaster, Dr. Dobb's
sean.coady@ubm.com

Jon Erickson Editor in Chief Emeritus, Dr. Dobb's

CONTRIBUTING EDITORS

Scott Ambler

Mike Riley

Herb Sutter

DR DOBB'S UBM TECH

303 Second Street,
 Suite 900, South Tower
 San Francisco, CA 94107
 1-415-947-6000

INFORMATIONWEEK

Rob Preston VP and Editor In Chief, InformationWeek
rob.preston@ubm.com 516-562-5692

John Foley Editor, InformationWeek
john.foley@ubm.com 516-562-7189

Chris Murphy Editor, InformationWeek
chris.murphy@ubm.com 414-906-5331

Art Wittmann VP and Director, Analytics, InformationWeek
art.wittmann@ubm.com 408-416-3227

Alexander Wolfe Editor In Chief, InformationWeek.com
alexander.wolfe@ubm.com 516-562-7821

Stacey Peterson Executive Editor, Quality, InformationWeek
stacey.peterson@ubm.com 516-562-5933

Lorna Garey Executive Editor, Analytics, InformationWeek
lorna.garey@ubm.com 978-694-1681

Stephanie Stahl Executive Editor, InformationWeek
stephanie.stahl@ubm.com 703-266-6030

Fritz Nelson VP and Guest Editorial Director
fritz.nelson@ubm.com 949-223-3608

David Berlind Chief Content Officer, UBM Tech
david.berlind@ubm.com 978-462-5315

ART/DESIGN

Mary Ellen Forte Senior Art Director
maryellen.forte@ubm.com

Sek Leung Senior Designer
sek.leung@ubm.com

INFORMATIONWEEK.COM

Benjamin Tomkins Managing Editor
benjamin.tomkins@ubm.com 516-562-5336

Roma Nowak Senior Director, Online Operations and Production
roma.nowak@ubm.com 516-562-5274

Tom LaSusa Managing Editor, Newsletters
tom.lasusa@ubm.com

Jeanette Hafke Web Production Manager
jeanette.hafke@ubm.com

Joy Culbertson Web Producer
joy.culbertson@ubm.com

Nevin Berger Senior Director, User Experience
nevin.berger@ubm.com

Atif Malik Director, Web Development
atif.malik@ubm.com

INFORMATIONWEEK BUSINESS TECHNOLOGY NETWORK

EVP of Group Sales, InformationWeek Business Technology Network, Martha Schwartz
 (212) 600-3015, martha.schwartz@ubm.com

Sales Assistant, Salvatore Silletti
 (212) 600-3327, salvatore.silletti@ubm.com

SALES CONTACTS—WEST

Western U.S. (Pacific and Mountain states) and Western Canada (British Columbia, Alberta)

Sales Director, Michele Hurabiell
 (415) 378-3540, michele.hurabiell@ubm.com

Strategic Accounts

Account Director, Sandra Kupiec
 (415) 947-6922, sandra.kupiec@ubm.com

Account Manager, Vesna Beso
 (415) 947-6104, vesna.beso@ubm.com

Account Executive, Matthew Cohen-Meyer
 (415) 947-6214, matthew.meyer@ubm.com

MARKETING

VP, Marketing, Winnie Ng-Schuchman
 (631) 406-6507, winnie.ng@ubm.com

Marketing Director, Angela Lee-Moll
 (516) 562-5803, angele.leemoll@ubm.com

Marketing Manager, Monique Kakegawa
 (949) 223-3609, monique.luttrell@ubm.com

Program Manager, Diane Scala
 516-562-5476, diane.scala@ubm.com

SALES CONTACTS—EAST

Midwest, South, Northeast U.S. and Eastern Canada (Saskatchewan, Ontario, Quebec, New Brunswick)

District Manager, Steven Sorhaindo
 (212) 600-3092, steven.sorhaindo@ubm.com

Strategic Accounts

District Manager, Mary Hyland
 (516) 562-5120, mary.hyland@ubm.com

Account Manager, Tara Bradeen
 (212) 600-3387, tara.bradeen@ubm.com

Account Manager, Jennifer Gambino
 (516) 562-5651, jennifer.gambino@ubm.com

Account Manager, Elyse Cowen
 (212) 600-3051, elyse.cowen@ubm.com

Sales Assistant, Kathleen Jurina
 (212) 600-3170, kathleen.jurina@ubm.com

AUDIENCE DEVELOPMENT

Director, Karen McAleer
 (516) 562-7833, karen.mcaleer@ubm.com

BUSINESS OFFICE

General Manager, Marian Dujmovits
United Business Media LLC
 600 Community Drive
 Manhasset, N.Y. 11030
 (516) 562-5000



Copyright 2012.
 All rights reserved.

UBM TECH

Paul Miller, CEO

Kathy Astromoff, CEO, Electronics

Robert Faletta, CEO, Channel

Edward Grossman, President, Business Technology Media

Marco Pardi, President, Business Technology Events

David Berlind, Chief Content Officer

John Dennehy, Chief Financial Officer

David Michael, Chief Information Officer

Martha Schwartz, Chief Sales Officer, Business Technology Media

Scott Vaughan, Chief Marketing Officer

Simon Carless, EVP, Game & App Development and Black Hat

Lenny Heymann, EVP, New Markets

Harris Grayman, SVP, People & Culture

UNITED BUSINESS MEDIA LLC

Pat Nohilly Sr. VP, Strategic Development and Business Administration

Marie Myers Sr. VP, Manufacturing

INFORMATIONWEEK VIDEO

informationweek.com/tv

Fritz Nelson Executive

Producer

fritz.nelson@ubm.com

INFORMATIONWEEK BUSINESS TECHNOLOGY NETWORK

DarkReading.com

Security

Tim Wilson, Site Editor

tim.wilson@ubm.com

IntelligentEnterprise.com

App Architecture

Doug Henschen,

Editor in Chief

doug.henschen@ubm.com

NetworkComputing.com

Networking, Communications, and Storage

Andrew Conry-Murray, Editor

andrew.murray@ubm.com

PlugIntoTheCloud.com

Cloud Computing

John Foley, Site Editor

john.foley@ubm.com

InformationWeek SMB

Technology for Small

and Midsize Business

Benjamin Tomkins,

Site Editor

benjamin.tomkins@ubm.com

Byte.com

Larry Seltzer

Editorial Director,

larry.seltzer@ubm.com

Dr. Dobb's

Good Stuff for Serious

Developers

Andrew Binstock

Editor in Chief

andrew.binstock@ubm.com

Entire contents Copyright © 2012, UBM Tech/United Business Media LLC, except where otherwise noted. No portion of this publication may be reproduced, stored, transmitted in any form, including computer retrieval, without written permission from the publisher. All Rights Reserved. Articles express the opinion of the author and are not necessarily the opinion of the publisher. Published by UBM Tech/United Business Media, 303 Second Street, Suite 900 South Tower, San Francisco, CA 94107 USA 415-947-6000.