For: Application Development & Delivery Professionals

Continuous Delivery Is Reshaping The Future Of ALM

by Kurt Bittner, July 22, 2013

KEY TAKEAWAYS

Continuous Delivery Equals Competitive Advantage

Releasing software many times a day may sound like science fiction, but it's reality for a growing number of firms. Dramatically reduced cycle time will become the new normal, enabling more direct engagement with customers to deliver high-value experiences. Firms that can't deliver applications rapidly will find themselves left behind.

AD&D Is Pivoting From The Department Of No To The Enabler Of Go

Business leaders are frustrated with an unresponsive AD&D and endless excuses for why it can't deliver solutions faster. Continuous delivery, a shift to assembly of components and services, and a new DevOps compact provide the means to pivot away from the frustrations of the past to become the enabler of new, exciting digital customer experiences.

You Can't Succeed In The Future With The Organization Of The Past

The AD&D organization of the future is flat, lean, and responsive. Chains of command will give way to empowered, accountable, cross-functional teams. Top-down directives will give way to innovation from everywhere. Micromanagement of task assignments will give way to servant leadership, self-directing teams, and transparent communication.

Continuous Delivery Is Reshaping The Future Of ALM

Vision: The Application Life-Cycle Management Playbook

by Kurt Bittner
with Phil Murphy, John R. Rymer, Jeffrey S. Hammond, Tom Grant, Ph.D., and Steven Kesler

WHY READ THIS REPORT

The novelist William Gibson has noted that "The future is already here — it's just not very evenly distributed." That's precisely the current state of affairs with application life-cycle management (ALM). A previously quiet revolution in ALM called DevOps is now noisily elbowing its way into the mainstream. If Agile software development was the opening act to a great performance, continuous delivery is the headliner. The pace at which consumers expect change is causing a crisis in application development and delivery (AD&D), but it's exactly what's essential to create a sense of urgency. Agile was a good start, but it wasn't sufficient to drive better business results; continuous delivery forges the broken link in the value chain, connecting business strategy with business results. Even if fully continuous delivery is not your goal, proceeding toward it will help you increase reliability, reduce risks, and decrease costs. The revolution is just beginning, but it's real — and the time to act is now.

Table Of Contents

Notes & Resources

Forrester interviewed vendor and user companies including Atlassian, Avanade, CA, CollabNet, General Electric, HP, IBM, Joyent, Lean Minds, Microsoft, Motorola, PTC, and ThoughtWorks.

Related Research Documents

To The Victor Go The Spoils: How The Need For Speed Is Reshaping The ALM Landscape
July 11, 2013

Next-Generation Portfolio Management: Strategic, Lean, And Delivery-Enabled
May 13, 2013

The Forrester Wave™: Application Life-Cycle Management, Q4 2012
October 23, 2012

WELCOME TO HYPERSPACE

What seemed like science fiction just a few years ago is now the norm: Smartphones have taken over the Internet, smart devices are creating digital experiences unimagined just a few years ago, and cloud computing is everywhere. Historically, waves of technical innovation have swept away competitors unable to keep pace with change. Today is no different, and that innovation is happening at an increasing pace. The days when a successful organization could release software once every 12 to 18 months are over, and while organizations are releasing faster, there's still much room for improvement (see Figure 1). Time scales have compressed, customer expectations have soared, and releasing software daily is no longer the stuff of science fiction. Leading firms are already doing it, and their competitors are racing to catch up.

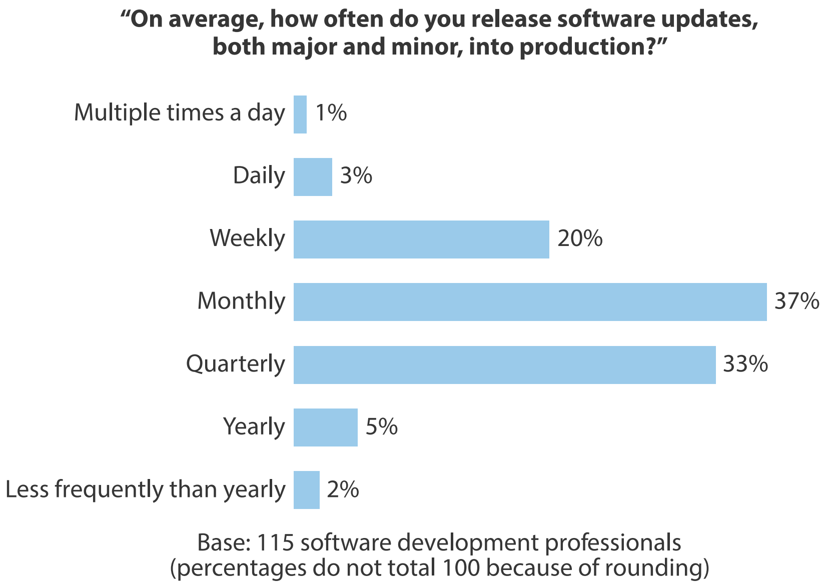
- **The future has arrived — and just in time.** Companies such as Adobe Systems, Amazon.com, Facebook, Flickr, Google, Netflix, and uTest — as well as mainstays like HP, IBM, and Microsoft — are showing us the future of ALM: lean, fast, and focused on customer experiences. ALM delivers innovation to customers as a constant stream of capabilities, and feedback flows back to drive decisions in real time. Ponderous requirements and design processes are a thing of the past. Sources such as Eric Ries' book *The Lean Startup* have informed this new model: Rapid delivery of a minimum viable product (MVP) results in validated learning, which drives the cycle of innovation and business value realization.

- **Continuous delivery is the logical evolution of Agile.** Continuous delivery has always been a part of some Agile approaches, but most Agile adoption in the past decade has seemed to focus on optimizing workflow within a development team rather than across the entire value chain. What the business really wanted was to go faster — that is, to get solutions into the hands of real customers faster. Continuous delivery is now coming into sharp focus, finally giving the business what it wanted all along. Continuous delivery does not replace Agile approaches; rather, it enables Agile to finally deliver on its promise to business leaders: faster delivery of genuine business value. Recent additions to offerings from ALM industry mainstays such as CA, HP, IBM, and Microsoft that focus on continuous delivery are a sign of the central importance of this capability to future ALM.

- **DevOps is breaking the logjam.** The force behind this revolution is DevOps: development and operations working together with business sponsors and quality assurance (QA) to deliver a continuous stream of innovation into production. The movement began as an application of Lean principles to the software delivery life cycle, resulting in a highly automated process that simplifies or eliminates handoffs, collapses roles, and eliminates error-prone and time-consuming manual work. Coupled with small batches of changes flowing through the value chain at any one time, the result is a dramatic foreshortening of the time from "go" (development) to "show" (use) in a real production environment.

- **Innovation when and where you need it.** Not everyone needs to release continuously for all system types, but everyone needs to release in very short cycles for at least some of the time for some applications. Whether it's to support a critical fix or to deliver a game-changing new business capability, business needs — not technical constraints — should drive the choice of when to release. Every organization has a hot-fix process to deploy emergency bug fixes; what most firms lack is a predictable, stable approach to accommodating rapid change as the norm. Continuous delivery practices provide the means to replace the manual processes and personal heroics associated with emergency fixes with a regular, controlled, and rapid deployment pipeline. Although rates of change will vary by the type of system and type of organization, *effective* continuous delivery simplifies and accelerates the innovation process for all types of systems (see Figure 2).

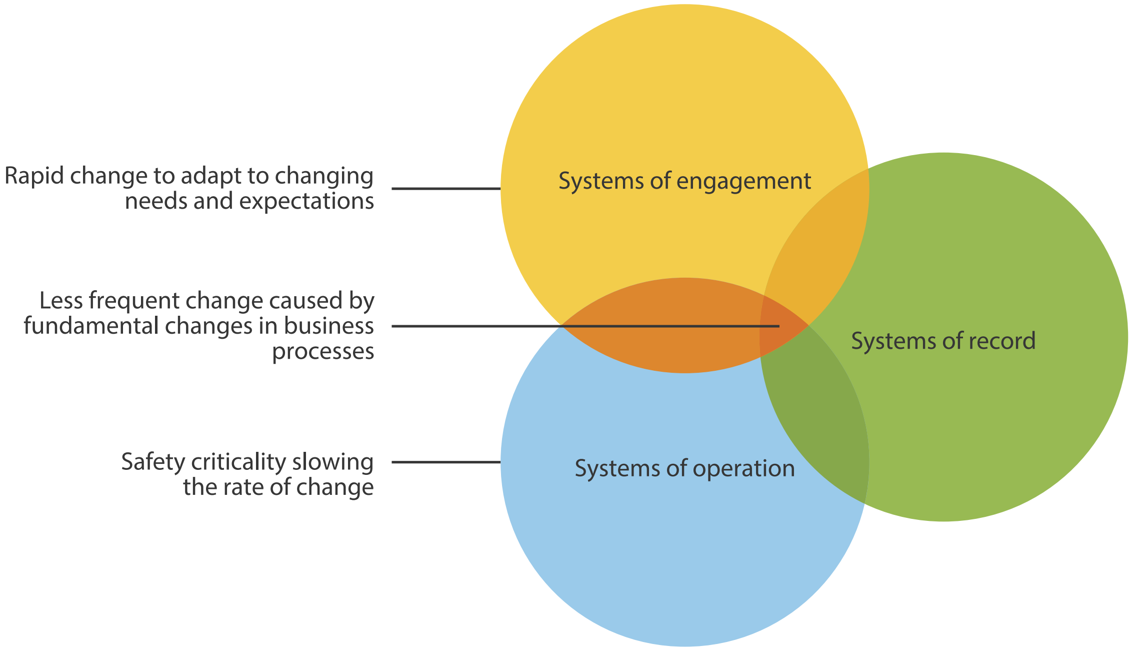
Figure 1 Release Cycles Are Getting Faster, But There Is Room To Improve



Source: March 2013 Global Application Life-Cycle Management Online Survey

99321 Source: Forrester Research, Inc.

Figure 2 Different Systems Are Under Different Pressures To Change


Rapid change to adapt to changing needs and expectations
Less frequent change caused by fundamental changes in business processes
Safety criticality slowing the rate of change
Systems of engagement
Systems of record
Systems of operation

99321

Source: Forrester Research, Inc.

A NEW SDLC IS EMERGING

The old software development life cycle (SDLC) was based on an industrial model: directives from on high filtering down to a highly specialized workforce whose actions were meticulously planned and monitored. Companies measured work output in terms of task completion and believed that good performance resulted from standardizing work production, activities, and tools. Change was painful and disruptive, so firms minimized it. Despite being an endeavor focused on automating business processes, the old SDLC was strangely manual, requiring heroic efforts and constant planning to coordinate a surprising amount of low-skilled work. As a result, management focused on reducing the unit cost of labor through outsourcing. Once it made funding decisions, projects took on a life of their own with little regard for the broader needs of the organization.

Fast-forward to today: We have realized the old SDLC model no longer meets the needs of a rapidly evolving business environment. The new SDLC is based on an adaptive, goal-directed model; the business sets the destination and the delivery team adaptively steers toward the goal. Even then, there is far more collaboration to determine the shape of the solution than before. Rather than micromanaging teams with task assignments, teams work out for themselves what they need to do to reach the destination. Change that results in a better solution is welcomed. Rather than focusing myopically on labor costs, the new SDLC focuses on total life-cycle cost and streamlines processes while applying heavy doses of automation to improve predictability and repeatability. Finally, the company continuously evaluates the value stream of the initiative in the context of the entire portfolio of opportunities (see Figure 3).

Figure 3 A New Software Development Life Cycle Is Emerging

Old SDLC	New SDLC
Directive	Adaptive
Task-oriented	Goal-directed
Specialized roles	Empowered teams
Resistant to change	Optimally responsive
Outsourced	Automated
Project optimization	Portfolio optimization

99321 Source: Forrester Research, Inc.

THE NEW NORM IS MULTIPLATFORM AND MULTIDEVICE

Modern applications are composite, spanning multiple devices and system types. Building and testing modern apps is a complicated mess of platforms, languages, and technology stacks; deploying them is even worse, as systems change on different time scales across various geographies and device updates are largely out of your control. The complexity of the problem is still increasing, although the rate of change is slowing; ALM tools from established vendors such as HP, IBM, and Microsoft and newer players such as Atlassian, CollabNet, and GitHub are tackling various aspects of this complexity by providing a single environment in which teams can develop and test multiplatform, multidevice apps. While the challenge will never completely go away, help is starting to arrive.

- **Powerful applications are easier than ever to build — but complexity has also increased.** Service orientation has paid off in big ways: Firms increasingly assemble modern applications from APIs and services, combining on-device apps with cloud-based services to provide much richer interactions. This API-driven ecosystem provides richer interoperability than ever before, but it has also introduced new complexities. Knowing what services might be available to solve a particular problem, designing for component replaceability, and automating API-driven testing of composite applications all create new challenges. Service catalog software that helps organize and communicate available services and their interfaces is becoming an essential enabler of ALM success.

- **No application is an island.** Social media is an ever-present force in today's application environment. If the ability for consumers to share experiences, good or bad, on social channels such as Facebook, Instagram, and Twitter doesn't both excite and frighten you, it should. Successful applications of the future will weave social media into the experience fabric from the start to enhance the customer experience. The result will be a new set of skills for teams to master, a new set of capabilities to test, and a production environment that merges the public cloud and private environments in new and seamless ways.

- **Context awareness will become critical.** The Internet of Things is here, and it will enable your applications to interact with a whole new set of devices to become more context-aware. Just as mobile phones can now interact with other nearby phones to share files and pictures, the devices of tomorrow will become even more aware of their surroundings and each other. Imagine, for example, your phone interacting with your hotel room thermostat and lighting controls to tailor your experience to your preferences. Now imagine the developers of that same room environment control software having to deal with the possibility of multiple people coming and going from hotel rooms or conference rooms: Which device controls the temperature? Awareness creates the opportunity for rich interactions, but current application development approaches barely deal with today's complexity, let alone leverage the possibilities of tomorrow. Tools and techniques will need to evolve to maximize the possible.

THE AD&D ORGANIZATION OF THE FUTURE IS FLAT, LEAN, AND RESPONSIVE

The old walls are coming down. In a slower-moving age, organizational silos and role specialization made sense, or were at least tolerable. As cycle times fall, specialization and hierarchies get in the way of delivering faster; role handoffs and resource coordination waste precious time. The new model features empowered, cross-functional teams and servant leadership that focuses on clearing roadblocks and improving flow. Institutionalizing these new practices will take time and concerted effort.

- **Business and DevOps will forge a new alliance.** The current arm's-length relationship between business leaders, development, and operations is giving way to a dynamic collaboration that results in rapid cycles of innovation. In organizations where software *is* the business, application development is merging into the line-of-business organization. In other companies, tight alignment based on shared goals is sufficient to achieve the rapid response and throughput needed to meet the needs of rapidly changing markets. In both cases, forming a tight alliance with operations to ensure smooth and frequent releases will be essential to the execution of business strategy.

- **Centers of excellence (COEs) will evolve into communities of practice (COPs).** When specialized skills are scarce, COEs are a common way to organize and allocate scarce resources and share them across an organization. COPs are less formal, peer-to-peer mechanisms that focus on sharing experiences and building skills. In the AD&D organization of the future, team members will exercise a wide range of skills that cross traditional role boundaries. In order to eliminate handoffs that stall progress and increase cycle times, shared resource pools — especially those housing QA professionals and business analysts — will dissolve and firms will distribute their members into delivery teams. What will remain is a need to improve specialized skills across teams — not individuals. The COP model, coupled with broader professional social networks, will fill this gap. As a transitional step, specialists whose skills are in high demand but scarce supply will take on a player/coach role, doing work for teams while they also transfer skills to team members.

- **Generalists will trump specialists.** In the classic "factory" model of software development, specialists with narrow but deep skills worked in a highly orchestrated fashion to deliver software projects. Orchestration took time and led to frequent handoffs and skill gaps when specialist skills were not available. Organizations are now eliminating specialization and the resultant handoffs in order to reduce cycle time and deliver faster. This will increase the demand for generalists who can not only develop but also test and understand customer needs. While some specialization will still exist, it will do so within a generalist framework: A developer may be a little better at user interfaces or at build and test automation but will still be capable of doing typical development work. Technology specialization will increasingly happen behind the interfaces to services and frameworks provided by other organizations. With the move to generalization, organizations will need to reimagine and realign career paths with their needs.

- **Self-directing teams will still need leadership — just a different kind.** With a move to empowered, self-directing teams, middle managers will struggle with their new role. Coming from an environment in which they are accustomed to directing people and making decisions, they find that they no longer "run projects." Instead, they will provide "servant leadership," clearing roadblocks and supporting the delivery teams with the resources they need to get things done. They have the organizational awareness to know whom to contact to cut through organizational layers and resolve issues quickly. The challenge in refocusing middle management is deep and cultural: These managers have become accustomed to a world in which directive behavior and boxes on an organizational chart were highly valued. Senior leaders will need to recognize this and alter the reward structure in conscious ways to drive the desired results.

- **Outsourcing models will change to enable flexibility and speed of delivery.** The past decade saw outsourcing, and especially offshoring, become prevalent at firms of all sizes. The primary motivation behind this was to reduce direct labor costs; it tended to focus on the developer and QA roles. With outsourcing comes increased complexity of coordination and, typically, more formal handoffs between team members. With reducing cycle time and delivering continuously as the primary objectives, the waste and delay that these handoffs create are intolerable. Future sourcing and team collaboration models will focus more on seamlessly integrating team members from a variety of sources into a single cohesive team, regardless of their geographic location. The increase in QA automation will shift teams away from low-skilled manual testing work to higher-value contributions. Future sourcing models will focus on flexibly balancing resources while providing highly skilled team members who quickly integrate themselves into existing teams.

- **Crowdsourcing will play a significant role.** Crowdsourcing is, in effect, open source at scale across the entire SDLC. Opportunities to leverage crowdsourcing for innovation ideas, test solutions, drive feedback, and even develop parts of your solutions make crowdsourcing something you can't ignore. Whether it's managing feedback streams or finding the useful ideas among the empty pizza boxes after the hackathon ends, the availability of ALM solutions like TopCoder and uTest will expand to address this new set of challenges and opportunities.

PORTFOLIO-DRIVEN ALM WILL DELIVER A CONTINUOUS STREAM OF VALUE

While ALM is evolving, portfolio management is also undergoing a transformation driven by continuous delivery. Annual planning cycles will become a thing of the past, and the old project structure will break down in an environment where there is a continual need to deliver innovation. Instrumented applications will not only provide feedback to delivery teams but also inform portfolio decisions, blending application and portfolio-level planning in a continuous cycle.[1]

- **The voice of the customer will be that of the actual customer.** Many organizations have implemented programs to solicit customer input. These "customers" tend to be focus groups with relatively small sample sizes, usually drawn from the population of power users at large, influential clients. The small sample sizes and relative bias toward richer capabilities often lead organizations to increase the complexity and narrow the appeal of their applications. Instrumented applications will provide an alternative view of what the entire customer base is doing, balancing and augmenting focus-group data.[2]

- **Continuous delivery will change demand management.** The ability to deliver quickly confers both a benefit and an imperative to break down demand into small batches of work. Doing so results in rapid value realization, reduces risk, and improves throughput. There is a significant hidden benefit: When releases are infrequent events, there is a natural tendency to pile on scope "just in case" the next release is not for years. The problem is more complex than it first appears: Not only does the business not get what it asks for, it asks for things that no one ever uses and expects things it never asked for. The result is overloaded projects, delay, and gold-plated features. Pushing small changes quickly through the value chain significantly reduces or even eliminates scope bloat.

- **Product focus will replace project focus.** Projects are, by definition, temporary one-off endeavors focused on delivering a discrete result. The vast majority of what BT currently calls a "project" is hardly temporary: Most are sustaining efforts focused on developing or improving a product or service. Funding the work as a temporary endeavor only obscures the need for a stream of ongoing investment. Taking a long-term product focus enables organizations to better align strategic interests and current activities, and prevents short-term execution concerns from endangering long-term viability.[3]

NEW APPROACHES WILL FILL THE DEPLOYMENT PIPELINE

In this new world of fast delivery cycles, the bottleneck will shift from software release to requirements. Poor requirements are a perennial problem and continue to rank as a major source of frustration in realizing rapid delivery (see Figure 4). Current trends in the automation of deployment activities show that the build-test-deploy process can and will be streamlined and automated, leaving the front end of the process most vulnerable to disruption.

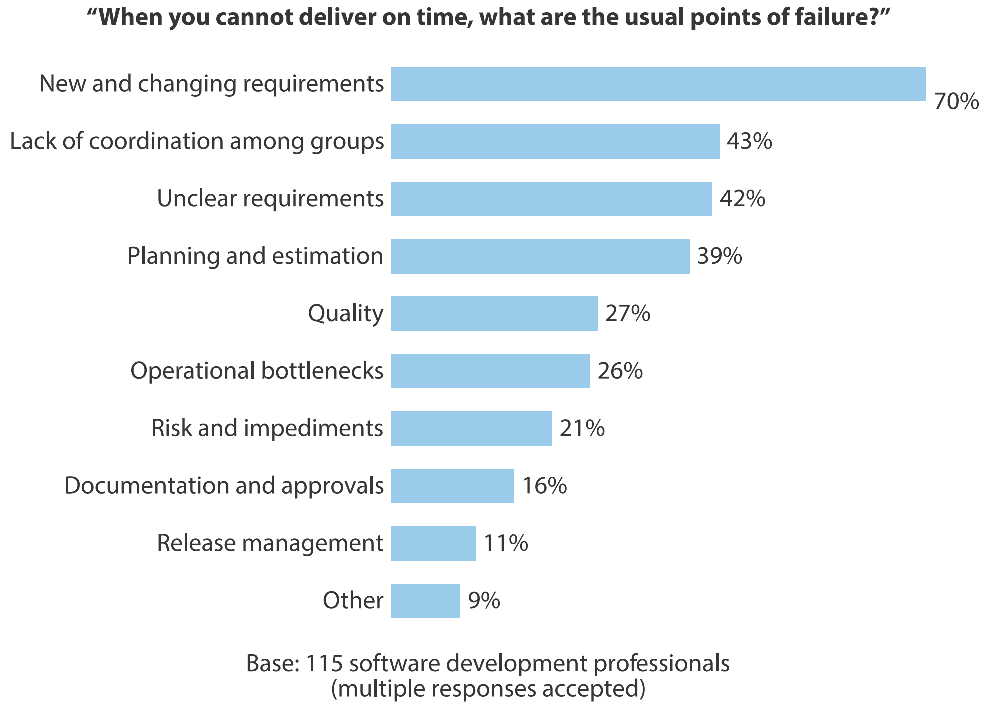
- **MVPs will replace BRDs.** Continuous delivery relies on a foundation of small batches of features that are easier to develop, easier to test, and easier and less risky to deploy. The result is a constant stream of innovation and faster realization of value. In requirements terms, this means that the focus is on finding the most valuable thing to work on; building, testing, and getting that thing out the door quickly; and then finding the next most valuable thing. MVPs are the new reality, and the "lean startup" model is leading to a new requirements paradigm.[4] The big requirements document (BRD) is a thing of the past.

- **Direct customer feedback will drive the innovation cycle.** One of the flaws in requirements approaches is having someone speak on behalf of real customers. Regardless of whether you call them "subject-matter experts" or "product owners," they all bring biases and preferences to the decision-making process. With the ability to instrument applications to see what people use, don't use, and struggle with, we can now go directly to the source. Instrumenting applications gives us these insights; the result will be apps that better meet real needs. Analytics tools like PreEmptive Solutions' PreEmptive Analytics foreshadow a new kind of ALM tool that will become essential for gathering and aggregating customer feedback.

- **Don't think requirements — think testable hypotheses.** The very term *requirement* is a misnomer. Many "requirements" are not required; they sometimes aren't even nice to have. A better term that is starting to come into favor is "testable hypothesis."[5] Most requirements specify a solution to some problem, even though no one may have explicitly identified or articulated that particular problem. Explicit outcomes are the backbone of good solution design, meaning that defining the criteria for evaluating the "goodness" of the product is part of the product definition. Being explicit about a problem or outcome sometimes often yields a discussion among team members that leads to an even better way to achieve that outcome. The shift to a testable hypothesis-driven approach means that traditional requirements techniques — which haven't yielded very good results anyway — will gradually disappear.[6]

> "We feel the ripple effects of bad requirements all the time; they make it harder for us to build a good user experience, test the right things, and prioritize our backlog effectively." (Software professional)

Figure 4 Requirements Are A Consistent Source Of Delivery Problems



Source: March 2013 Global Application Life-Cycle Management Online Survey

99321 Source: Forrester Research, Inc.

ALM IN THE CLOUD WILL BECOME THE RULE, NOT THE EXCEPTION

The same aspects that make cloud-based computing attractive for production environments also apply to ALM tooling: effective load balancing, reduced administrative overheard, simplified support, on-demand scaling, and instant access to the latest versions of software. Nearly all vendors offer cloud-based versions of their ALM software, with downloads available for those organizations that still feel the need to install locally. Even integrated development environments are becoming available in the cloud. Given the compelling advantages, it's likely that the cloud will become the ALM environment of choice for most organizations. Cloud-based ALM is already becoming a reality, with solutions from CollabNet, GitHub, HP, IBM, Microsoft, and others.

APPLICATION MODERNIZATION IS A WAYPOINT ON THE JOURNEY

Like the modern city that rests upon roots of ancient civilizations, the modern enterprise rests upon a foundation of decades of applications developed when today's needs were unimagined. Monolithic applications of the past can't support the need for continuous delivery, but support them they must. The solution is simple in theory but Herculean in practice: Componentization and refactoring monolithic code bases into callable services will break the monolith into pieces that can be changed with far less complexity and far more agility, enabling change at the speed of delivery demanded by modern enterprises. Aging applications, appropriately refactored and retooled, will continue to serve the needs of organizations for years, if not decades, to come.

RECOMMENDATIONS

DON'T WAIT FOR THE FUTURE — IT'S ALREADY HERE

As pioneering computer scientist Alan Kay noted, "The best way to predict the future is to invent it."[7] With continuous delivery, the long-standing goal of delivering solutions faster is within reach; trailblazing organizations are making this a reality today. That future awaits your organization. To make continuous delivery a mainstream practice in your organization, get started now by laying a foundation and then systematically building a new ALM organization atop it. To get started:

- **Build on a foundation of continuous integration (CI).** CI will enable you to reliably build software whenever code changes are checked in, running unit tests to ensure the basic integrity of the product. You won't know if it solves the right problem (yet), but if you can't do basic CI, you need to start there. Over time, increase the amount of testing to include smoke tests, integration tests, and even regression, performance, scalability, and security tests. Include automated code reviews over time as well.

- **Break down the walls between business, application development, and operations.** Start with early and frequent interactions and build up to continuous interaction throughout the life cycle. Work up to being able to deploy any build, even if you do not plan to do so. If there are reasons why you would not, attack and eliminate the barriers.

- **Refactor MVPs from monolithic proposals to reinvent portfolio management.** Inside every big idea is a small one dying to get out. Cull the MVP from your monolithic proposals by breaking demand into small batches, prioritizing them, and using them to fill a continuous delivery pipeline. Take steps toward the cultural changes required to adjust your funding models to support continuous streams of innovation rather than isolated and disconnected projects. Put feedback mechanisms in place to monitor the customer-derived feedback and measures of business value delivery.

- **Adopt skills and adapt roles.** Delivery teams need broad and deep generalist skills, not narrow specialist skills; start working now to broaden the skills of your people. Reward teams for delivering business results, not for hitting arbitrary milestones. Refocus middle management on servant leadership, clearing roadblocks and supporting teams.

- **Change your sourcing agreements to support and reward value delivery.** Ensure that you can build highly skilled, integral, cohesive, cross-functional teams no matter where you source the resources from. Give partners incentives to work in a way that eliminates handoffs and delays that disrupt flow. Avoid fixed-bid contracts that force you to define detailed requirements up front.

ENDNOTES

[1] For a more comprehensive view of the future of portfolio management, see the May 13, 2013, "Next-Generation Portfolio Management: Strategic, Lean, And Delivery-Enabled" report.

[2] Clayton Christiansen's book The Innovator's Dilemma describes how listening to customers can lead to failure to create breakthrough innovation. Existing customers generally want richer, more complex products, but market growth opportunities almost always favor simpler, less expensive solutions. Understanding what makes your applications hard to use, what features are actually used, and which ones are not used, are among but a few of the benefits of instrumenting applications for feedback. Source: Clayton M. Christiansen, The Innovator's Dilemma : When New Technologies Cause Great Firms to Fail, Harvard Business Review Press, 1997.

[3] In practical terms, product focus means providing a continuous funding stream for ongoing enhancement and evolution tied to the revenue model of the business the application supports. For more on this approach, see the December 23, 2009, "Product-Centric Development Is A Hot New Trend" report.

[4] Eric Ries' book The Lean Startup provides an articulate and compelling application of Lean principles to innovation in software products. Source: Eric Ries, The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, Crown Business, 2011.

[5] Source: Sam Guckenheimer and Neno Loje, Visual Studio Team Foundation Server 2012: Adopting Agile Software Practices: From Backlog to Continuous Feedback (3rd Edition), Addison-Wesley Professional, 2012.

[6] Acceptance test-drive development (ATDD) is an emerging practice that provides a compelling way to capture these testable hypotheses in the form of acceptance test cases. Open source tools such as Cucumber and FitNesse are popular enablers of this technique, which has the added benefits of increasing test coverage, increasing the level of test automation, and providing more direct means of assessing whether the application meets the needs of its users and stakeholders.

[7] The quote is attributed to Alan Kay at a 1971 Xerox PARC meeting, although similar quotes have been attributed to Peter Drucker, among others.

About Forrester

A global research and advisory firm, Forrester inspires leaders, informs better decisions, and helps the world's top companies turn the complexity of change into business advantage. Our research-based insight and objective advice enable IT professionals to lead more successfully within IT and extend their impact beyond the traditional IT organization. Tailored to your individual role, our resources allow you to focus on important business issues — margin, speed, growth — first, technology second.

FOR MORE INFORMATION

To find out how Forrester Research can help you be successful every day, please contact the office nearest you, or visit us at www.forrester.com. For a complete list of worldwide locations, visit www.forrester.com/about.

CLIENT SUPPORT

For information on hard-copy or electronic reprints, please contact Client Support at +1 866.367.7378, +1 617.613.5730, or clientsupport@forrester.com. We offer quantity discounts and special pricing for academic and nonprofit institutions.

Forrester Focuses On
Application Development & Delivery Professionals

Responsible for leading the development and delivery of applications that support your company's business strategies, you also choose technology and architecture while managing people, skills, practices, and organization to maximize value. Forrester's subject-matter expertise and deep understanding of your role will help you create forward-thinking strategies; weigh opportunity against risk; justify decisions; and optimize your individual, team, and corporate performance.

« **ANDREA DAVIES,** client persona representing Application Development & Delivery Professionals