



The Business Value of Continuous Delivery



Contents

3	Introduction
4	Faster, Better, Cheaper
7	Game Changer for Business
9	Challenges
9	Customer Experience
10	Conclusion

Introduction

Continuous delivery allows you to deliver new software and update existing software faster, with lower risk.

Reducing risk is important, but the processes that underpin continuous delivery translate into even more important values to the business:

- » **Accelerate time-to-value.** A small business doesn't need an MBA to recognize that continuous delivery helps them get things done. A big business that has mapped out its value stream and has complex investments and obligations across a large organization will find that continuous delivery helps accelerate time-to-value.
- » **Data-driven decision making.** Deploy, measure, adjust. You can still push larger scale releases, but your processes will be better suited to continuous data gathering. That will shorten the feedback loop with your customers. It sharpens your ability to respond, plan your next move, and keep ahead of the competition.
- » **Quality.** Behaving like you're releasing continuously forces you to raise your quality bar and fully automate test practices. Better quality means happier customers, lower costs, fewer fire-drills, and less unplanned work.
- » **Experimentation = innovation.** Developers and lines of business are free to try new ideas cheaply, unlocking innovative ideas that have been penned-in behind long high-investment release cycles.
- » **Reduce cost.** Big releases have big costs and big consequences when things go wrong. Keeping deliverables in a release-ready state drives the cost of delivery downward.

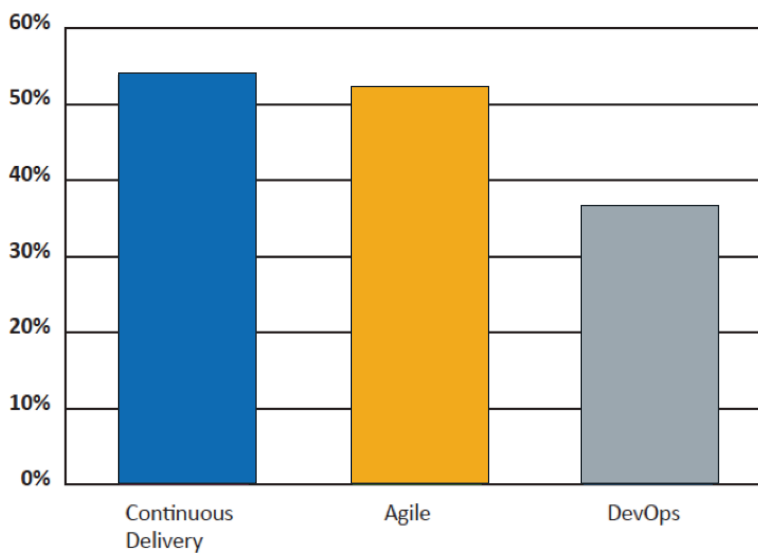
Together these values make continuous delivery a real game-changer for business. While adoption can begin and be proven at a team/project level, the nature of continuous delivery is that it crosses organizational boundaries in ways that require a real investment and top-down commitment. Choosing a continuous delivery toolchain that complements and coexists with your existing investments is a key step toward success.

Faster, Better, Cheaper

Developers talk about “removing friction” that slows them down from delivering code. But developers are often not even aware of why a particular gate or friction point exists. For one thing, developers are prone to local optimization – delivering their own code faster does not mean that the overall organization is delivering faster and may even slow the larger org down. Some friction points in a release process serve the same function as metering lights on a freeway, optimizing for more throughput at the expense of individuals having to wait a bit to get in the flow.

Continuous delivery, which is predicated in large part on test and process automation, allows the individual to get work done in parallel, while the necessary friction points in your delivery process serve their function. This is why continuous delivery is mostly adopted at a development team level, or by collaboration between development teams and ops teams in specific areas. Those teams would like to push the approach across the larger organization, but they typically don’t control many of the friction points. So, they push the limits within the boundaries they have. From a business standpoint of a larger organization, a broader commitment is really needed to maximize benefits.

Which initiatives is your organization looking to implement?



Key Findings

Continuous delivery takes the lead, with agile a close second. About 36% selected DevOps, placing it third.

64% said their initiatives have executive level visibility and sponsorship within their organization.

Figure 1: Approximately 54% of survey respondents said their organization was looking to implement continuous delivery. Continuous delivery was closely followed by agile, with about 51%.

¹ <http://go.xebialabs.com/rs/xebialabs/images/XEBIALABS-survey-Report-2014.pdf>

A key tenet of continuous delivery is to make small deliverables routine – patches, small projects, UI changes, upgrades, new features. When you do something routinely, you get good at it. You shave off all the rough edges that would otherwise be tolerable in something you only have to deal with occasionally. Intuitively, this leads to faster delivery. Part of the speed comes from reduced “integration cost” that typically happens at the end of a longer release cycle. The integration is done continuously using a tool like Jenkins, providing notification early and often of any problems. The continuous style means that when a problem is encountered, it's more easily identified and fixed, since it hasn't been masked by or mixed up with other work. The beneficial effects are multiplied when your team is distributed, or work has to be coordinated across time zones. Making smaller deliverables continuously means your overall project is delivered faster.

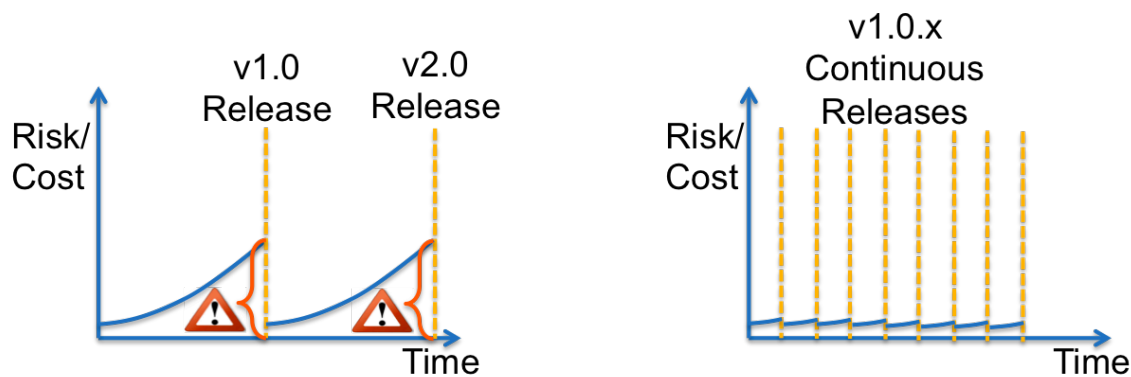


Figure 2: Continuous delivery reduces risk and cost compared to Big Bang releases

The cloud is an important enabler of continuous delivery. You can set up a continuous delivery pipeline entirely on-premise using traditional technologies, such as a managed set of virtual images and scripting. Setting aside the many variables associated with the nebulous term “cloud” (e.g., public v. private, IaaS v. PaaS, etc), at its heart, cloud is about serving up resources quickly when you need them. Those resources can be more infrastructure-centric (e.g., compute, storage, etc) or application-centric (e.g., runtime containers, surrounding systems/services used by applications). What they have in common is reducing friction via automation and providing more powerful ways to interact with the IT systems that support your team's work. Thus, cloud brings an entirely new set of tools that help you to achieve continuous delivery – instant access to resources, simple ways to stage software prior to production push, cheap exploratory work that can be thrown out or built upon.

A commitment to quality can feel amorphous from a business standpoint - how does that commitment translate into both costs and benefits? Traditionally, quality costs come in the form of people, supporting resources, and processes; benefits come in the form of customer satisfaction and lower support costs. Continuous delivery changes the equation. First, by definition, it forces a commitment to automation of test/ delivery processes, and those investments pay off not just in efficiency but in improved quality. Automated testing is not the same as high quality testing, but as developers and operations people automate their processes, reproducibility is improved feedback cycles are reduced. Second, by transforming existing work processes to smaller, testable, routine deliverables, you eliminate fire-drills and minimize the cascading impact of errors that plague “big bang” projects. Thus, as quality improves, you reduce the unplanned work that undermines schedules and forces ugly trade-offs as business-driven deadlines loom.

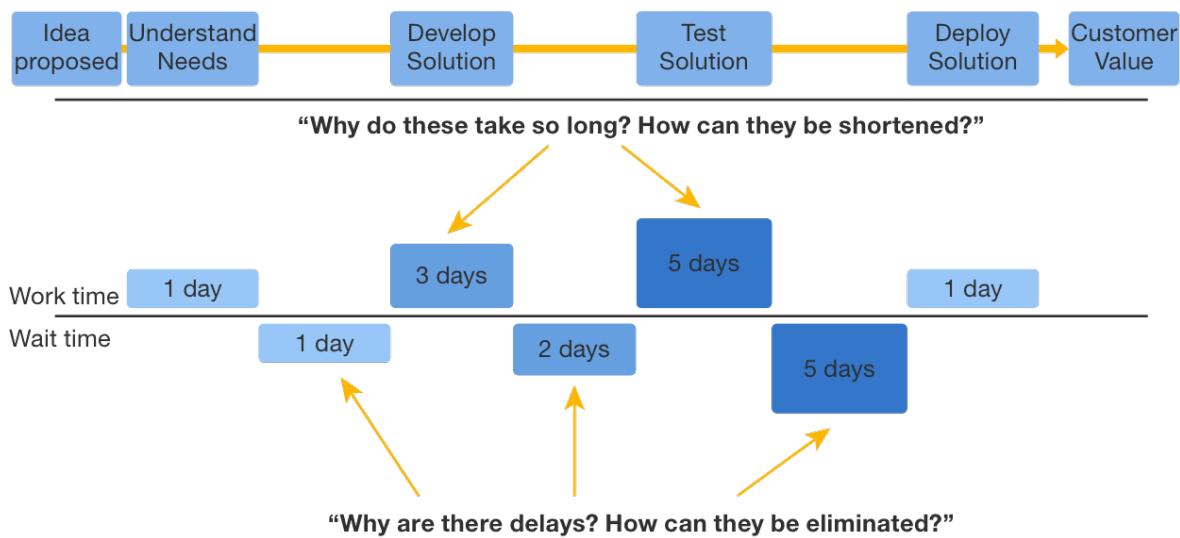


Figure 3: Continuous delivery reduces delays as a priority, speeding time-to-value²

Continuous delivery opens up entirely new ways to create better software. Your software will improve if you can fine-tune it with direct feedback. This feedback can be gathered through many avenues. For example, continuous delivery of a hosted service enables rapid A-B testing, phased roll-outs and “dark launches” (delivering new code that is not user-visible). These techniques give you live data about the impact of your choices with very limited risk. The data helps you fine-tune further investment and implementation details.

² “Application Delivery In The Modern Age”, February 12, 2014, Kurt Bittner, Forrester

Game Changer for Business

Continuous delivery is a hot topic at a business level for a number of good reasons:

- » Early adopters have proven its value. Mainstream adopters have both observed its advantages and felt the competitive sting as their more nimble competitors outpace them.
- » Devops as a movement has gained traction. Business people understand that having a common understanding between development and ops, breaking down siloed behavior, and developing a culture of responsibility across the organization is a key step to improved effectiveness and time-to-market. In many ways, continuous delivery is synonymous with devops.
- » As software “eats the world,” it’s becoming clearer every day to business leaders that IT must be used as a strategic asset. Being able to shorten delivery times, improve quality, and adapt to change quickly while properly dealing with security, availability, and compliance is the challenge. Continuous delivery, with its emphasis on automation and early, directed feedback, is the means to achieve some of those goals.
- » When you enable cheap, low-risk experimentation through continuous delivery, you can direct business investments with more information, and uncover opportunities you would otherwise completely miss.

Challenges

Getting to continuous delivery is a process – one that can be very hard and very long for enterprises, for a lot of good reasons. It may be virtually impossible or prohibitively expensive to automate testing of existing software systems. New software often has to interact with existing systems. Systems of record are subject to legal and procedural restrictions that prevent automation of end-to-end processes. There are typically organizational barriers between the development team creating new software and the operations team who is responsible for pushing changes to production and answering pagers in emergencies. The list of challenges to get to continuous delivery goes on. Many of those challenges are rooted in corporate culture and cannot be overcome by applying technology. None of this means you can’t make progress and enjoy some significant benefits by choosing the proper scope for continuous delivery within your organization.

Barriers to Continuous Delivery

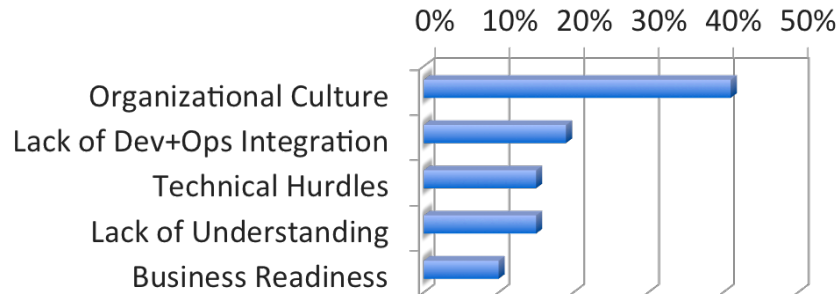


Figure 4: Technology is less of a barrier to continuous delivery than organizational culture

One of the most successful ways to get moving on continuous delivery is to start with a team on the software development side of the organization. The downstream consumers of their output have a vested interest in receiving “release ready” software, since they’re the consumers of the release (even though this may be far upstream of any end user-visible product). The downstream team will invest in the process if the value they receive (in the form of reduced headaches, fire-drills and predictability) is worth their investment. Top-down buy-in to continuous delivery is necessary, so they are empowered to make the investments in the face of conflicting requirements. When the downstream team is part of the same “development” tribe, the adoption process is simpler. As the handoffs begin to cross larger organizational barriers, your experience and data from implementing continuous delivery and management awareness can help to drive these kinds of localized cultural changes more broadly.

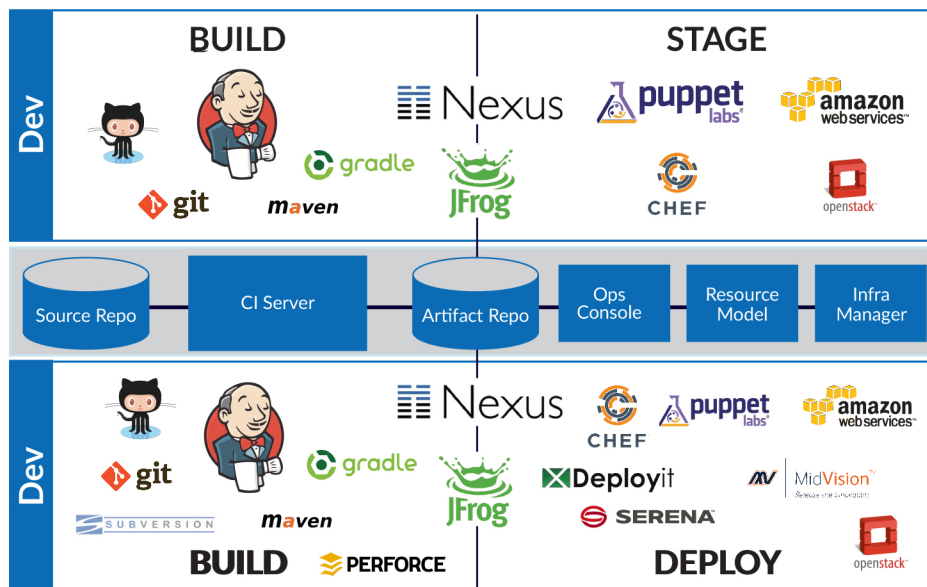


Figure 5: CloudBees Continuous Delivery Platform

³ <http://blog.devopsguys.com/2013/03/13/continuous-delivery-adoption-barriers/>

Jenkins, the most popular open source continuous integration server, is a common touch-point between development and ops teams. As such, it tends to play a central role in organizations on the path toward continuous delivery. The “business” may impose corporate standards and regulatory-driven requirements/systems on dev and ops teams, and Jenkins provides both of them with an automation tool that connects to existing systems. Furthermore, as the organization moves to adopt cloud technologies, both public and private, CloudBees Core makes operating Jenkins at enterprise-scale easier, whether on-premise or in the cloud.

Customer Experience

Helping customers across years of huge industry change means CloudBees has a deep roster of customer case studies showcasing the real benefits of continuous integration and continuous delivery. Whether you’re looking for insight from hipster mobile startups, industry disrupters, thought leaders or large scale enterprises, [you can find them using CloudBees](#). In the cloud, on-premise, or anyplace in between, CloudBees and Jenkins are strategic choices for companies who know their business depends on delivering software at the speed of ideas.

Here are some examples of business value being realized as a result of continuous delivery and DevOps transformations.

“With all the gains we have made in our CI and CD automation process, our developers have doubled their velocity...At a business level, ABN AMRO is able to respond to competitive threats more quickly and introduce new innovation on an ongoing basis.”

– Stefan Simenon, head of COE tooling and software development, ABN AMRO

“With DevOps...we are able to innovate faster and deliver more value and better service to our customers than ever before and with higher quality and security features than we ever thought possible. We have strengthened our market position and have been able to quickly expand into more regions which ultimately drives revenue.”

– Ryan Berrío Cardona, director of IT architecture, Suramericana

CloudBees Drives Enterprise DevOps and Continuous Delivery

CloudBees is the hub of DevOps, providing companies of all sizes with smarter automation solutions and actionable insights for accelerating software delivery. Our continuous delivery solutions offer DevOps teams the industry’s broadest suite of products, from on-premise to cloud native, from self-managed to self-service, from guided best practices to flexible choice. By making the software delivery process more productive, manageable and hassle-free, CloudBees puts companies on the fastest path to transforming great ideas into great software and returning value to the business more quickly.

Conclusion

Continuous delivery is transforming the way that businesses use their IT assets to connect with customers and partners. Continuous delivery builds on years of hard-won experience in agile processes and continuous integration to bring those benefits to a business level instead of simply being techniques used by development teams. Many of the keys to success are rooted in organizational and cultural transformation, as development and operations personnel learn how to collaborate and share responsibilities. For those organizations that make the transformation to continuous delivery and a supportive DevOps culture, the business value is real.

Learn More



- ↓ Read how CloudBees customers benefit from DevOps and continuous delivery
www.cloudbees.com/customers
- ↓ Visit our DevOps Resource Center to learn more about DevOps and continuous delivery
www.cloudbees.com/devops
- ↓ Learn more about the CloudBees product suite for transforming software delivery
www.cloudbees.com/products

Learn more
www.cloudbees.com/products