Zend Blueprint for Continuous Delivery

How to Calculate the ROI of Continuous Delivery

by Ido Benmoshe

Continuous Delivery Economics

What does it take to build and sustain market leadership? Increasingly, the ability to deliver quality applications rapidly to support business goals and drive innovation has become critical to a company's success.

As applications become more central to a company's competitiveness, an increasing focus is being placed on how to deliver the right applications to market rapidly. This has led organizations embrace the concept of Continuous Delivery, a methodology created to streamline and automate the process of software delivery.

By streamlining application delivery across the entire development lifecycle, applications can be iteratively developed, automatically packaged and tested, and then released to production on demand in a rapid, consistent manner.

Executives often ask:

How can I measure the expected return from an investment in **Continuous Delivery?**

This paper will propose a methodology that can be applied to assess the potential financial gains and build the justification for adopting Continuous Delivery methodologies.

What kind of returns can I expect from a Continuous Delivery investment?

When implemented collaboratively by dev and ops teams, Continuous Delivery will drive agility in the software delivery practice, allowing frequent software releases that are able to address the changing requirements of the business with fewer dependencies on IT related considerations.

Reducing time to market of new and enhanced capabilities while ensuring higher quality delivers financial benefits from both revenue increase and cost savings. Continuous Delivery also maximizes the IT organization's efficiency and productivity. This enables organizations accelerate their time to market while maintaining flexibility, reducing risk and significantly increasing application quality.

Average Results of Investment

21% increase in new software and services delivered

22% Improved quality of deployed applications

19% increase in revenue

50% fewer failures

Sources: [1] Study of 1,300 IT executives and managers - CA Technologies, Sept 2013; [2] Survey of 4,000 IT Operations professionals – Puppet Labs, April 2013

Average ROI of Continuous Delivery Investments

Expectations for web application performance and availability are rising as a result of the importance of applications to business performance, and increased adoption of mobile, social and cloud.

A calculation by incomediary.com [3] found that Amazon's website for example generates over $1,000 per second. Google is not far behind with over $900 per second, Yahoo! and eBay each generate over $200 per second and most of the top 20 global websites by revenue generate over $24 in revenue per second.

However assessing the actual impact on the business financial performance and evaluating the return on investment (ROI) of Continuous Delivery methodologies is a fairly complex task.

The traditional ROI formula divides the net gains from an investment, by the cost of the investment.

$$\text{ROI \%} = \frac{(\text{Gain from investment - Cost of investment}) \text{ x } 100}{\text{Cost of investment}}$$

For process improvement calculations, the potential gains include both tangible and intangible elements. The tangible elements are derived from revenue growth, headcount and infrastructure costs, and reduced cost of failures. The more intangible are elements are areas such as team efficiency, productivity and risk reduction.

Continuous Delivery involves both a mindset change and the introduction of a new process and methodology. The good news is that while implementing Continuous Delivery demands skills adjustment, it does not typically require additional headcount or upfront investment.

It is common to see adoption of best practices occur in stages. Incorporation of supporting technologies happens over time in a gradual manner, where the process is continuously improved while already acting on metrics and feedback.

The tangible gains that can be assessed are divided into the following categories:

Revenue **gains** from accelerated **time to market** of new functionality (GTM)

Gains from enhanced IT team productivity and cost reduction of IT **headcount** waste (GHC)

Gains from cost reduction of application failures resulting from increased **quality** (GQL)

Gain from **flexibility** in the IT environment (GFX)

Assessment of the total gains and the corresponding ROI formula will therefore look like this:

$$\text{ROI \%} = \frac{(\text{GTM} + \text{GHC} + \text{GQL} + \text{GFX} - \text{Cost of investment}) \times 100}{\text{Cost of investment}}$$

In this paper we will focus on measuring the tangible gains from Continuous Delivery, however, it is worth pointing out that there are other "soft ROI" components to the equation.

Soft ROI Components

- **Reducing** the **risks of failure** and inconsistencies
- Higher **customer satisfaction**
- **Competitive advantages** gained
- Improved **market perception**
- Improved **employee motivation**

Assessing the Gain from New Functionality Arriving Earlier to Market (GTM)

Implementing Continuous Delivery allows organizations to deliver features, enhancements and fixes much faster. This faster release of software to production (without compromising quality), allows organizations to respond more quickly to market demands and gain competitive advantage.

The impact of the rate of innovation on bottom line revenue varies between industries and increases with more interactive user engagement models. The recent "State of DevOps" survey [3] found that high performing organizations ship code 30 times more often compared to their peers who are not following DevOps practices. This allows advanced organizations to release new code multiple times a day while close to 30% of the mature organizations possess the capability to release software on demand. In an extreme example, the mean time between production code deployments at Amazon.com, for example, can reach 11.6 seconds [2].

These findings are backed by a survey commissioned by CA which evaluated the measurable impact of adopting DevOps methodologies [1]. Companies have experienced an average of 20% reduction in time to market of new applications and 21% growth in delivering new software that would otherwise not be possible to complete. The same survey finds an average of 19% revenue increase from the application.

Even when taking into account a more conservative ratio of a 15% gain compared to the results demonstrated by surveyed companies in this research, the gains are substantial.

The gains from accelerated time to market can therefore be estimated at:

GTM = [Revenue Increase estimation] x [Revenue per year] = 115/100 x [Revenue per year]

As an example: for an application in the top twenty web applications by revenue range, a 15% revenue increase equals an additional $113M per year. For an application generating around $100M in revenue or equivalent business value, the acceleration in features arriving to market as a result of a Continuous Delivery practice adoption can present an additional $15M gain.

Assessing gains from application failures reduction due to higher quality (GQL)

Whereas revenue is a well defined and measurable value, evaluating the impact of IT failures and the effects of delivering higher quality software is more complex. The cost of application failures is assessed in part from tangible, measurable costs associated with complete failures or chronic poor performance. However unless an application experiences a complete outage, the definition of a failure is subjective.

Often intermittent failures such as temporary slowdowns and occasional user transaction faults are not detected or occasionally considered by the business as acceptable at certain occurrence frequencies. These failures however tend to have a long term impact on user satisfaction and new customer acquisition. The state of DevOps survey found that high performing organizations experience 50% fewer failures and restore service 12 times faster than their peers [3].

Teams with DevOps:

50%
fewer failures

12x
faster time to restore

Source: The State of DevOps Report (by Puppet Labs), 2013

In order to get a better estimate of the potential revenue from improved quality, we also need to evaluate the differences in recovery time. Organizations practicing DevOps can achieve a mean time to recovery of less than an hour in 47% of incidents compared to 17% in a traditional IT practice [5]. A separate survey commissioned by RebelLabs found that 40% of respondents not implementing DevOps take 60 minutes or longer to restore service with the majority of incidents lasting at least 30 minutes. In contrast, 40% of the companies taking advantage of DevOps practices are able to restore services in under 30 minutes and only 22% take 60 minutes or more [5].

DevOps oriented teams are almost twice as likely to recover in 10 minutes or less compared to their peers [5]. Considering a very conservative time to resolution average by assuming that resolution time of all incidents is capped at three hours, we can estimate an average of 28.3 minutes improvement per incident by adopting DevOps practices.

The average number of failures per month reported by the same RebelLabs survey [5] is 2.13 with the median being 1 failure per month or a total of 12 failures causing outages per year. For web applications that are close to the top 20 by revenue range, or just shy of $24 in revenue per second, the savings from the tangible application quality increase (GQL) can therefore result in average savings of between $489K and $1.04M per year. These savings are attained from reduction of the application downtime and improved time to recovery.

Teams with DevOps:

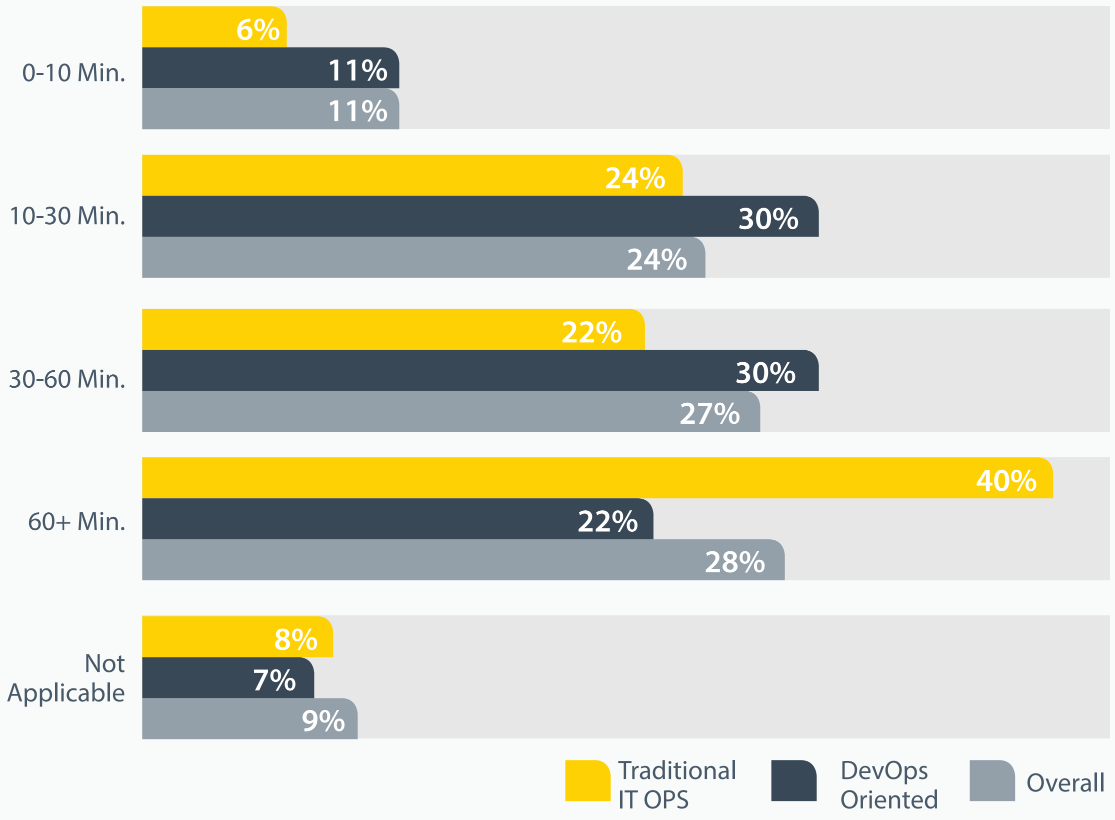
2x More

likely to recover in 10 minutes or less

Average of 28.3

minutes improvement over non DevOps centric teams

How Long do recoveries take?



Source: IT OPS & DEVOPS PRODUCTIVITY REPORT 2013 TOOLS, METHODOLOGIES AND PEOPLE, RebelLabs

Recovery time from application downtime for Traditional IT Ops vs. DevOps teams

A survey of 67 datacenters held in 2013 supports this downtime cost assessment and reports an average loss of revenue of $183K per unplanned outage incident which sums to an even higher cost of $2.2M in lost revenue from a total of 12 failures per year [6].

These estimations do not account for partial service interruptions or reputational and repeat business impact on future revenue. These longer term effects are unique to each application and usage pattern but can be definitely sizable for a highly visible or regulated web application.

The tangible impact assessed from quality improvement is therefore:

$$\text{GQL} = [\text{Failures per Year}] \times [\text{Avg minutes to recover difference}] \times [\text{Revenue per minute}]$$

It is important to note that each stage of adoption of Continuous Delivery can bring additional benefits in decreasing the rate of failure. Although a company may not start out by fully automating every process, each level of automation makes a difference, because the company becomes more able to detect and resolve problems earlier (and more efficiently) and is then able to deploy fixes to production immediately.

As an advanced and more fully automated example, Amazon.com is now able to continuously deliver and deploy software to production over 1000 times per hour, and has achieved a 75% reduction in outages and 90% decrease in outage minutes [6]. The more advanced and repeatable the delivery capability is, the more the applications environments are standardized, the more the behavior of the software is predictable and the level of quality rises.

Assessing the gains from IT headcount waste reduction and productivity increase (GHC)

Inefficiencies, lack of visibility and reliance on repetitive manual work force development and operations teams to waste valuable time on various chores that do not contribute business value.

On top of the loss of revenue from application outages, each incident causes a productivity loss and resource waste which averaged $54K per incident according to a survey of multiple datacenters [6].

However, waste is not limited to meltdowns but rather the majority of the IT resources waste is actually observed during normal operation. Considerable resources are wasted on an ongoing basis even when no critical production incidents are occurring.

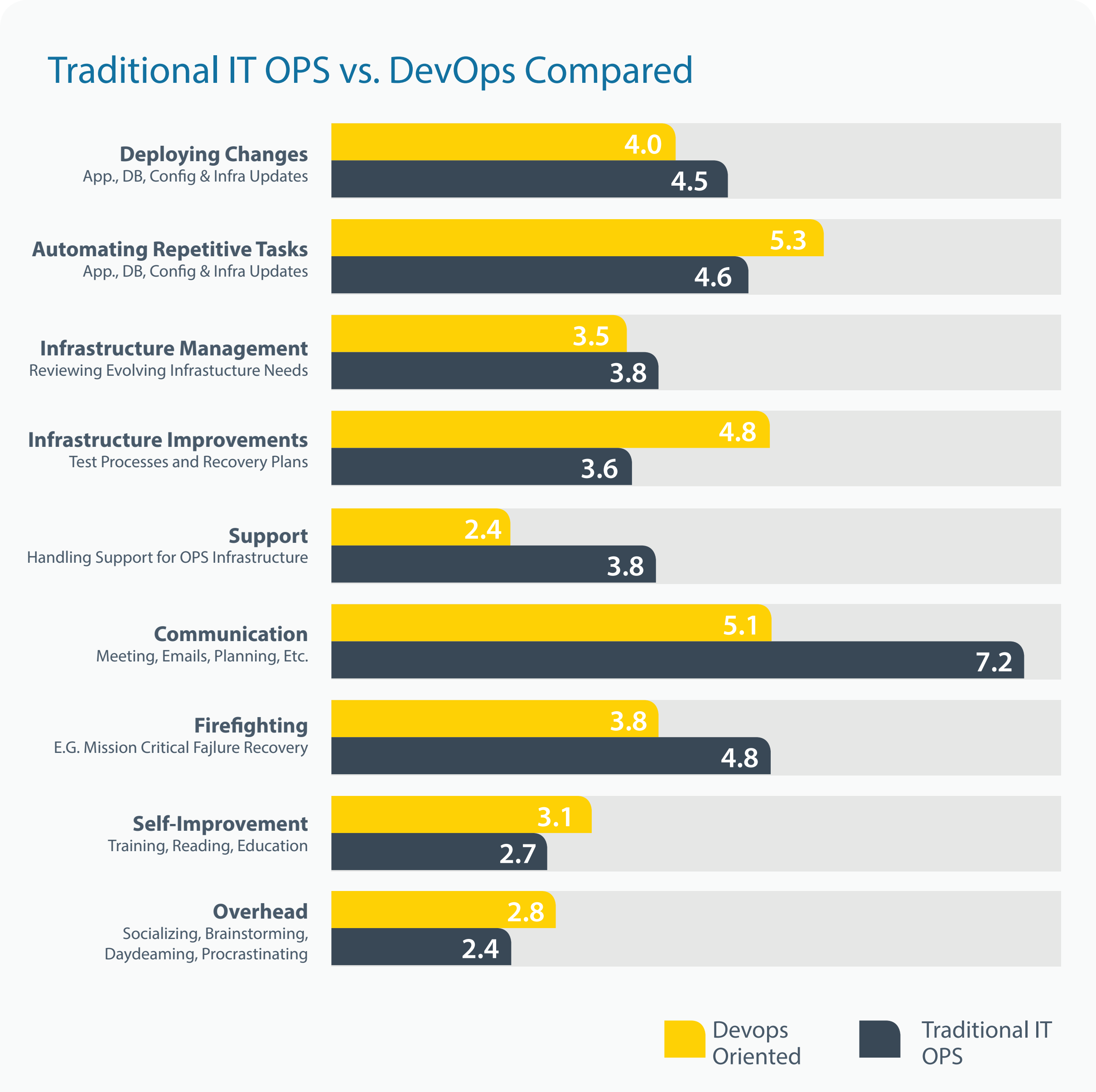
A number of key functions can attribute for most of headcount wastage:

1. Time spent by IT on building and maintaining environments
2. Time spent by developers on diagnostics and problem resolution
3. Time spent by developers on system integration
4. Time spent by testers on manual testing

IT operations teams divide their time between building environments, deploying software, resolving problems and maintaining the environments under their management. There are noteworthy differences in the time allocation and focus areas between organizations that practice DevOps methodologies and implement a Continuous Delivery workflow and their peers who operate in a more traditional fashion.

The IT Ops & DevOps productivity report [5] evaluated IT operations resource time management. Overall, IT engineers in traditional IT settings spend about 3 hours more a week to complete their tasks. DevOps oriented teams tend to consistently invest more in proactive automation and process improvement tasks. When discounting the time spent on improvement and education tasks, DevOps oriented teams save on average about 4.9 hours per week per person delivering their core IT tasks compared to traditional IT staff. They are also able to invest an additional 2.3 hours per week implementing improvements bringing the total time savings to 7.2 hours per week.

7.2
hours
savings
per week
per person



Source: IT OPS & DEVOPS PRODUCTIVITY REPORT 2013 TOOLS, METHODOLOGIES AND PEOPLE, RebelLabs

Key metrics for Traditional IT Ops vs. DevOps teams

Application platform inconsistencies and the manual nature of building and maintaining these environments are significant contributors to the waste. Application platform inconsistencies are responsible for over 40% of the delays in delivering code to production, forcing the operations team to devote more resources to problem resolution [7]. Only 28% of corporate PHP developers, for example, reported that their organization is standardized on a single application stack. Another 30% of the delivery delays are related to the manual nature of the deployment process and lack of automation.

Platform inconsistencies
40% of delays from code to production

Eliminating the environment inconsistencies throughout the application delivery process is achieved by automated provisioning and enforcing a standard, up-to-date, well maintained and supported application stack.

Automation of the deployment process is achieved by using an application packaging and deployment management mechanism that allows developers to package the application in a self contained format at the end of the integration phase and consistently deploying the same package in any environment that requires the application to be installed.

Standardization is key to ensuring that the IT team spends less time on building and maintaining multiple environments or resolving problems while at the same time reducing dramatically the amount of configuration related issues encountered due to the application platform inconsistency.

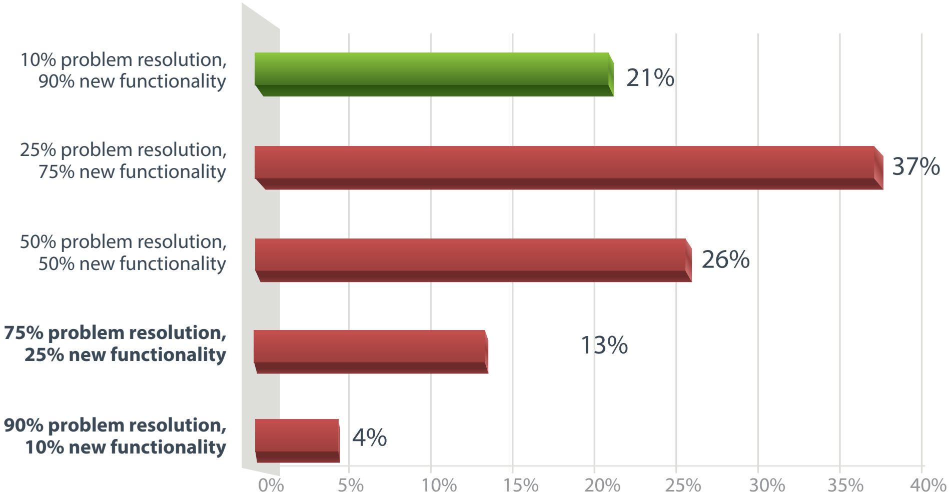
The savings form reducing the waste in the IT organization grows as the organization matures in the level of automation and standardization and peaks when reaching a fully automated Continuous Delivery workflow.

The net gain from reducing IT operations headcount waste is therefore:

GHC (IT Ops) = [IT Avg Salary] x [IT Staff headcount] x 0.16 (7.2/45 saving hours per week/total hours)

Developers divide the majority of their productive time between coding and problem resolution. Most of the wasted time is spent on problem diagnostics and resolution efforts. This happens in particular when developers have to analyze escalated issues from production. When asked how much of their time is spent on problem resolution and maintenance versus on developing new functionalities, the 2013 Zend Developer Pulse survey [8] found that almost 80% of developers spend 25% or more of their time on problem resolution, while 17% of developers spend 75% or more of their time. The majority of developers spend around 30-40% of their time finding and resolving problems and not implementing new functionality.

How much of your time is spent on production problem resolution and maintenance vs. developing new functionalities?



Base: 4,809 enterprise, SMB and independent developers worldwide
Source: Zend Technologies, Zend Developer Pulse Survey 2013

Developers typically fix problems fairly quickly when they have the proper diagnostics information and sufficient insight into the problem scenario. Approximately 75% of the time developers spend on problem resolution is spent on determining the root cause of the problem [9]. That time is wasted on attempts to gather sufficient diagnostics information and replicate the issue. When developers have limited visibility into production, they must recreate the same scenario with the same root cause which is extremely time consuming and causes up to a 30x increase in costs and resource investment dedicated to resolve production problems.

Preliminary Estimates of Relative Cost Factors of Correcting Errors
(Example Only)

	Where Errors are Found				
Where Errors are Introduced	Requirements Gathering and Analysis/ Architectural Design	Coding/ Unit Test	Integration and Component/ RAISE System Test	Early Customer Feedback/Beta Test Programs	Post-Product Release
Requirerments Gathering and Analysis/Architectural Design	1.0	5.0	10.0	15.0	30.0
Coding/Unit Test		1.0	10.0	20.0	30.0
Integration and Component/ RAISE System Test			1.0	10.0	20.0

Source: The Economic Impacts of Inadequate Infrastructure for Software Testing, NIST

A Continuous Delivery process can eliminate the time developers spend on problem resolution by at least 50-70% based on the maturity level of the process.

Teams with Continuous Delivery

50-70% less time

on problem resolution

There are three main drivers to this cost reduction.

1. Initially, the contribution comes from the notion that software has to always be production ready and in order to achieve such a state, rigorous testing is performed continuously. Testing the complete application often and early in the development cycle ensures that problems are found earlier, and problems found earlier take much less time to resolve.

2. The enforcement of standardized application environments (that are always provisioned automatically in the same manner resembling the production environment) eliminates time spent on configuration and environmental related issues.

3. Comprehensive application diagnostics and monitoring capabilities enable developers to solve problems without wasting significant time in reproducing the failed scenarios, before they can turn to resolving the issue.

Developers who have access to accurate and actionable data flowing from production and directly tied to the failure are able to resolve issues significantly faster and with less effort. This level of visibility into the execution of failed or poorly performing transactions in production is similar to the level of insight developers get in their familiar debugging environment on their desktops.

Time to fix a Bug Based on Discovery Point

Location	Hours	Current Distribution of Where Bugs are Founda	Weighted Average Hours
Requirements	1.2	7%	
Coding/unit testing	4.9	42%	
Integration	9.5	28%	
Beta testing	12.1	13%	
Post-product release	15.3	10%	
Total			**17.4**

Source: The Economic Impacts of Inadequate Infrastructure for Software Testing, NIST

The gains from reducing Developers headcount waste is measured as follows:

GHC (Developers) = [Developer Avg Salary] x [Dev Staff] x 0.18 (60%x75%x40% - Time reduction by implementing CD x Time spent on root cause analysis x Time spend on problem resolution)

The savings for example in an organization with 20 developers and 5 IT operations staff can result in $360K for the developers and $80K for the IT operations team. The gains in our calculation assess solely cost savings in IT capacity. However, more often the time saved by reducing waste in the IT organization is reinvested in developer and IT staff productivity to generate more revenue for the company.

The contribution of the extra IT staff capacity and their influence on revenue acceleration varies between organizations but it can be as high as a few multiples. The ZDnet cost of IT failures calculation, for example [10], assumes a 5x multiplier over the GHC savings value to assess the actual revenue gain impact. In the example above of 10 developers and 5 IT resources, the gain from additional productivity can reach $2.2M (5 x $440K).

Gains from flexibility in the IT environment (GFX)

Full control over the software manufacturing process and standardization on the application stack allows organizations practicing Continuous Delivery to take full advantage of the flexibility offered by virtualization and cloud technologies.

Zend's developer survey showed that 68% of developers intended to use public cloud services in 2013, though only 7 percent of applications currently reside in the cloud according to Accenture's High Performance IT study [11]. The study expects to see a growth to 33 percent by 2020 and states that companies will need to adopt a hybrid approach combining different environments and application platforms to address the needs of the business. But designing applications to support multiple, hybrid and changing environments is nearly impossible when the release process is manual and there is limited ability to detect problems and respond to changes.

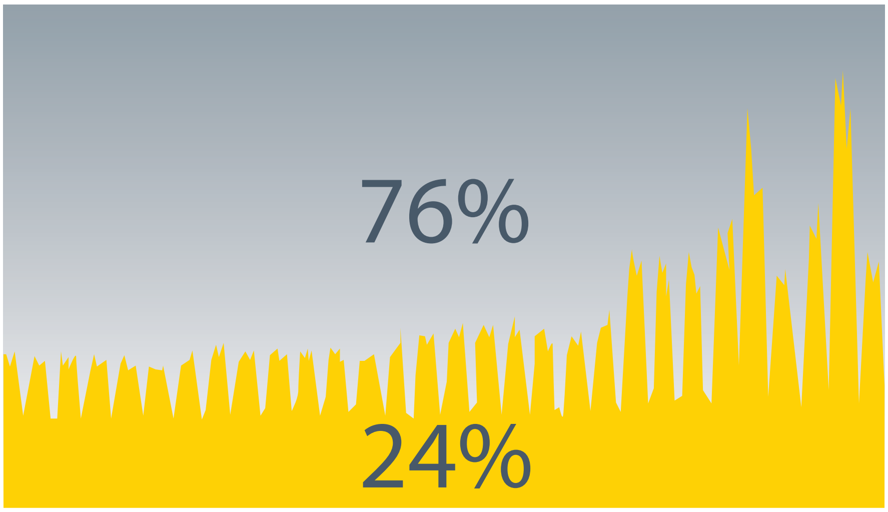
A fully automated and standardized process allows development and operations teams to adjust quickly to new environments and implement changes rapidly when needed.

The gains associated with increased flexibility are generated by reducing the total cost of ownership (TCO) of running the application in production. One of the significant sources of cost in the traditional datacenter comes fluctuating usage patterns. The usage pattern for amazon.com as an example, suggests that during one specific month, up to 76% of the computing resources could have been wasted if the environment had not been elastic and flexible.

Non flexible environment.

Up to 76%

of computing resources wasted



Monthly web traffic pattern (November) on Amazon.com and server resources capacity [21]

IDC Findings

72% TCO reduction

moving to cloud

Cloud technology adoption and the ability to scale based on actual fluctuating load allows companies to optimize their infrastructure spend. Continuous Delivery also enables an organization to more easily accommodate changing application runtime environments and seamlessly deploy the application to multiple types of environment on demand. This means that an IT organization is enabled to take full advantage of the potential offered by cloud technologies, and of the highly competitive cloud providers market.

IDC research found that 11 companies moving from on-premises production infrastructures to AWS cloud, which offers a high level of flexibility, reduced their application TCO by an average of 72% [12]. IT spending is generally considered by analysts to correspond to about 5% of the company's revenue [10].

Based on this assumption the gains from increasing production environment flexibility (GFX) by adopting private, public or hybrid cloud technologies in applications generating $100M in revenue can therefore reach an average of $3.6M. These gains can further increase as release flexibility and deployment automation reach higher maturity.

The gains from flexibility in IT environment can therefore be measured as:

GFX = [Application Annual Revenue] x 0.036 (72% Avg TCO reduction x5% cost of IT as percentage of revenue)

A Scorecard for Measuring ROI of Continuous Delivery Investment

Revenue gains from accelerated time to market of new functionality (GTM)

GTM = ______________ (Revenue per year) x 115/100 (Revenue Increase Estimate) = $ □

Gains from cost reduction of application failures resulting from increased quality (GQL)

GQL = ______________ [Failures per Year] x 28.3 [Avg minutes to recover difference] minutes x $ ______________ [Revenue per minute] = $ □

Gains from enhanced IT team productivity and cost reduction of IT headcount waste (GHC)

GHC (IT Ops) = ______________ [IT Avg Salary] x ______________ [IT Staff headcount] x 0.16 (7.2/45 saving hours per week/total hours) = $ □

GHC (Developers) = ______________ [Developer Avg Salary] x ______________ [Dev Staff] x 0.18 (60% Time reduction by implementing continuous delivery x75% Time spent on root cause analysis x40% Time developers spend on problem resolution) = $ □

Gain from flexibility in the IT environment (GFX)

GFX = ______________ [Application Annual Revenue] x 0.036 (72% Avg TCO reduction x5% cost of IT as percentage of revenue) = $ □

TOTAL GAINS [A] = $ □

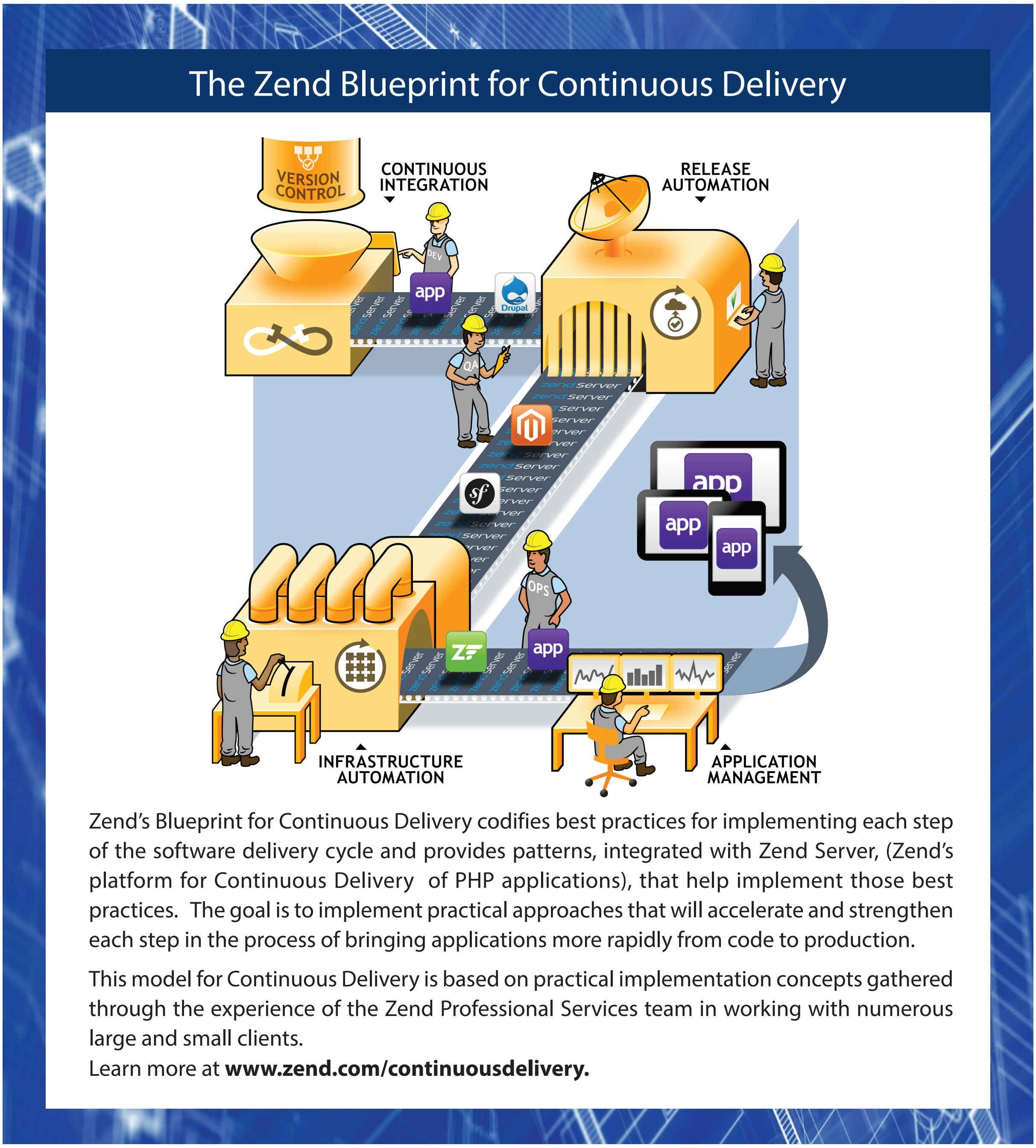
ROI % = (______________[A] - Cost of investment) x 100

Cost of investment

Summary

This paper has outlined a methodology for measuring the gains from an investment in Continuous Delivery, based on known benchmarks from other organizations. We have provided a framework for IT executives to evaluate the potential revenue gains and cost reductions realized from a mature automated software delivery process and apply those parameters to their unique application, IT staff and environment.

Gains to evaluate include direct revenue increases from accelerated time to market, reduction in IT and developer head count wastage, reduction in application failures due to increased application quality, and gains from flexibility in IT environment. When measured in an ROI calculation compared to costs of implementing more rapid application delivery practices, we at Zend believe (based on our numerous engagements with our clients), that most companies will see a compelling and direct near term return on investment.



The Zend Blueprint for Continuous Delivery

Zend's Blueprint for Continuous Delivery codifies best practices for implementing each step of the software delivery cycle and provides patterns, integrated with Zend Server, (Zend's platform for Continuous Delivery of PHP applications), that help implement those best practices. The goal is to implement practical approaches that will accelerate and strengthen each step in the process of bringing applications more rapidly from code to production.

This model for Continuous Delivery is based on practical implementation concepts gathered through the experience of the Zend Professional Services team in working with numerous large and small clients.
Learn more at **www.zend.com/continuousdelivery.**

Zend blueprint for continuous delivery

References and resources:

[1] DevOps Driving 20 Percent Faster Time-to-Market for New Services, Global IT Study Reveals, http://www.ca.com/us/news/press-releases/na/2013/devops-driving-20-percent-faster-timetomarket-for-new-services-global-it-study-reveals.aspx - from TechInsights Report: What Smart Businesses Know About DevOps, CA Technologies, September 2013

[2] Jon Jenkins, Amazon.com, "Velocity Culture", Velocity Conference June 2011

[3] The State of DevOps Report (by Puppet Labs), 2013, http://info.puppetlabs.com/2013-state-of-devops-report.html

[4] http://www.incomediary.com/20-websites-making-the-most-money

[5] IT Ops & DevOps Productivity Report 2013: Tools, Methodologies and People, RebelLabs

[6] 2013 Cost of Data Center Outages, Ponemon Institute Research Report, December 2013

[7] http://static.zend.com/topics/Zend-Developer-Pulse-report-Q2-2013-0523-EN.pdf

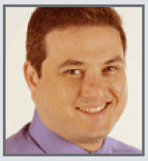
[8] http://static.zend.com/topics/zend-developer-pulse-survey-report-Q2-2012-0612-EN.pdf

[9] An ISV's Guide to Achieving Business Advantage through Faster Application Problem Resolution, BMC Software

[10] Michael Krigsman, "Worldwide cost of IT failure (revisited): $3 trillion" http://www.zdnet.com/blog/projectfailures/worldwide-cost-of-it-failure-revisited-3-trillion/15424

[11] Accenture Technology Vision 2014, From Digitally Disrupted to Digital Disrupter, www.accenture.com/technologyvision.

[12] "The Business Value of Amazon Web Services Accelerates Over Time", IDC Information and Data, December 2013



About the Author

Ido Benmoshe is VP, Global Support & Services at Zend, responsible for professional services, PHP training and certification programs, technical support and solution consulting. Ido has more than 17 years of experience in R&D management, technical services and product management. He participates regularly in industry events focusing on Application Lifecycle Management and IT Service management. The Zend Professional Services team leads implementations of agile methodologies, application design, optimization and Continuous Delivery solutions at global enterprises running mission-critical PHP applications. They focus on mentoring customers' development and operations teams to deliver faster releases of high-quality PHP applications with higher performance and availability.

About Zend

Zend partners with businesses to rapidly deliver modern apps across web, mobile and cloud. We helped establish the PHP language, which today powers over 240 million applications and web sites. Our flagship offering, Zend Server, is the leading Application Platform for developing, deploying and managing business-critical applications in PHP. More than 40,000 companies rely on Zend solutions, including NYSE Euronext, BNP Paribas, Bell Helicopter, France Telecom and other leading brands worldwide.

Corporate Headquarters: Zend Technologies, Inc. 19200 Stevens Creek Blvd. Cupertino, CA 95014, USA • **Tel** 1-408-253-8800 • **Fax** 1-408-253-8801
Central Europe: (Germany, Austria, Switzerland) Zend Technologies GmbH, St.-Martin-Str. 53, 81669 Munich, Germany • **Tel** +49-89-516199-0 • **Fax** +49-89-516199-20
International: Zend Technologies Ltd. 12 Abba Hillel Street, Ramat Gan, Israel 52506 • **Tel** 972-3-753-9500 • **Fax** 972-3-613-9671
France: Zend Technologies SARL, 105 rue Anatole France, 92300 Levallois-Perret, France • **Tel** +33-1-4855-0200 • **Fax** +33-1-4812-3132
Italy: Zend Technologies, Largo Richini 6, 20122 Milano, Italy • **Tel** +39-02-5821-5832 • **Fax** +39-02-5821-5400
Ireland: Zend Technologies, The Digital Court, Rainsford Street, Dublin 8, Ireland • **Tel** +353-1-6908019