CloudBees®

Real Approaches for Calculating the Business Value of DevOps
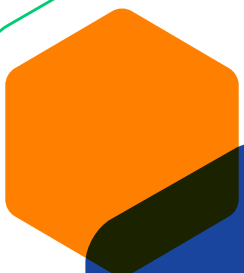
Contents

Executive Summary

Over the course of the past decade, companies and organizations of every size, from every part of the world and in every industry, have increasingly adopted the tools, practices and processes that collectively make up DevOps. The reasons for this widespread adoption are numerous and fairly well-known: increased speed to market, better collaboration across the various functions that touch and operate the software development and deployment lifecycle, increased quality, less manual and duplicative work, better alignment with agile principles previously adopted, more efficient utilization of resources...the list goes on and on.

Clearly, organizations that have adopted DevOps have not been shy about telling their stories - a simple Google search brings back any number of case studies, articles, presentation videos and the like telling story after story of DevOps transformation. These stories present a compelling picture of a larger transformation:

» How the adoption of DevOps has led to major changes in the way organizations operate
» How organizations bring their products to market
» How they engage with customers and the public, and even
» How they view themselves as a business

Despite all of the evidence of real change at organizations worldwide, many companies and even the DevOps community at large have had trouble describing the impact these transformational changes in process and practice have had in economic and business terms. Read through the success stories from that Google search mentioned previously and you will find no shortage of qualitative descriptions of before and after scenarios. You will also find plenty of examples of improvements in operational metrics. You might even find some testimonials from people in business roles like product managers or executives not even associated with IT - talking in qualitative terms about how DevOps transformations have made their products better, enabled them to be more responsive to the market or made them nimble enough to stay steps ahead of the competition. But what you will rarely find is how any of these operational or qualitative benefits translate into dollars or real evidence that connects DevOps improvements to companies' most crucial strategic goals. This is where most companies struggle, attempting to show return on their DevOps practices in hard dollars.

There have been many thought leadership pieces published that attempt to fill this gap and describe the business benefits of DevOps at a more aggregate level. While well-intentioned and carrying merit, these pieces of analysis usually fail to connect DevOps improvements to measurable business impact. Instead, their conclusions provide broad projections to be applied the same way to every organization, regardless of the vast differences that exist in size, situation, geography and industry (among others). Lacking any better options to quantify business impact, many organizations end up relying on these broad based guesstimates to estimate the business impacts of their own DevOps practices on their organization. The results are typically underwhelming. Executives and those in business roles rightly want to see real connection to their organization and impacts measured using internal company data.

At CloudBees, we believe strongly that organizations can and should develop these impact analyses for themselves - they are more accurate and more impactful because they are more representative of the specific organization. Luckily, creating these kinds of real business value analyses is much more achievable than most DevOps practitioners and leaders typically realize.

In this whitepaper, we will demonstrate just that. You will learn how to make the case for your DevOps initiatives by:

- » Detailing the problems with relying on the most common approaches to quantifying the business benefits of DevOps
- » Introducing the fundamentals of how to express business value, and identifying the three core metrics you need to address
- » Outlining an approach to measure the business benefits of DevOps within your organization

State of the DevOps Business Case

At CloudBees, we've worked with over 300 organizations to help them create, review or update business cases, business justifications and business impact analysis models and materials. At DevOps World and our regional customer conferences we hold well-attended workshops guiding DevOps executives, managers and practitioners in the basics of putting together a real, quantifiable business impact analysis of their DevOps initiatives. We have drawn two very important inferences from the high level of interest in these workshops:

1. Many members of the DevOps community recognize the importance of showing the business impact of DevOps
2. A high percentage of those attempting to create a business case on their own struggle with putting it together

What are the reasons behind this struggle? Why do so many within the DevOps community have difficulty quantifying positive business impact? There are a host of reasons that seem to surface when we ask these questions:

» They do not have any baseline data
» They feel it is too hard to measure the business impact of DevOps
» The benefits of DevOps are inherently technical and operational and not directly tied to the business
» The savings generated are "soft savings" not "hard savings"
» DevOps is not as important as other areas of IT (e.g., security)
» Business audiences don't understand DevOps well enough to recognize how it impacts the broader business

But when we dig a little deeper, we find these stated reasons are secondary, at best, and more like smokescreens, at worst. For example, a lack of baseline data isn't a reason to give up on conducting a business impact assessment; rather it's a reason to make good faith estimates and show the projected improvement. If business audiences don't understand DevOps or if it is seen as inherently technical, it's time to find out from them what increased speed to market and release of new features and improvements mean to their slice of the business!

In the end, we find the *actual* reasons for avoiding the building of a business case look more like this:

» **Engineers deal in absolutes by necessity.** To a developer, either features work or they don't; code is error-free or it isn't. This mentality results in technical stakeholders allowing "perfect" to be the enemy of the effective. Estimates, projections and nuance are all perfectly acceptable attributes of an effective business case.

» **DevOps engineers underestimate business audiences.** They are part of an important community that deals with problems that often require difficult problem-solving and a deep understanding of relevant technologies. While clearly laudable, this often leads engineers to underestimate the ability of non-technical audiences to understand the improvements and benefits that DevOps bring to the business.

» **Technical stakeholders and the community they belong to have their own language.** This technical language is really their native professional tongue. When asked to translate from this native technical language into business-speak, many technical personas feel unable or unwilling to do so.

» **Very often, technical team members have little interest in the core business objectives, measures, markets and strategies of their organizations.** This is a fascinating observation made over years and in completing hundreds of business value assessments. We've talked to countless numbers of DevOps managers and practitioners who live and breathe DevOps, but don't really have an interest in financial services, insurance, manufacturing, software - or whatever business their organization is in. As a result, they fail to see the impact to operational metrics that define efficient software development and delivery.

» **Many engineers and technologists become timid when stepping out of the comfort zone of the technical community.** As a result, their presentations to the business side of the organization can lack the boldness sometimes required to definitively demonstrate the connection between DevOps improvement and the core business.

» **DevOps leaders are unprepared for the skepticism and pushback that can come from business and executive audiences.** DevOps leaders (who are often tasked with conducting business impact evaluations) are used to the respect they've earned as experts in an important and complex domain. Very often, these leaders are unprepared for the pushback that can come from business and executive audiences when the subject matter turns to the subject of business impact - especially when spending money is a part of the discussion. DevOps leaders whose opinions and positions typically carry weight in a technical arena get blindsided by opposition in business arenas that they are not experienced in handling.

All of these obstacles - both perceived and actual - can and should be overcome. We will turn to that subject later in this paper. However, let's first examine the most common outcome when trying to tie DevOps value to the business: Reliance on outside sources to create studies for them.

It's Not You: The Problem With Outside Studies

Recognizing the need to tie DevOps results to business metrics, many experts have stepped in to try and solve this issue. There are numerous publicly available studies from leading technology analyst firms, DevOps consultancies and DevOps tools vendors seeking to provide a framework for understanding the business benefits of DevOps initiatives. There are numerous publicly available studies from leading technology analyst firms, DevOps consultancies and DevOps tools vendors seeking to provide a framework for understanding the business benefits of DevOps initiatives. Sometimes these reports look at the initiatives overall, other times they look at specific pieces of the whole (for example, studies focused on the impact of automated testing, feature flags, etc.). Given the absence of more specific analyses and the challenge experienced by organizations attempting to put their own together, these efforts definitely serve a purpose and are, for the most part, very well-intentioned. At CloudBees, we've published studies based on our work with hundreds of customers to try and provide an estimated impact of various continuous integration and continuous delivery improvements.

The problem lies in using and relying on these kinds of broad and general studies is that they are not a substitute for building a business case that speaks to the specifics of your particular organization. DevOps is a multi-faceted domain; one with many different potential component parts. It can be used to improve and impact businesses in any vertical. While many of the high-level improvements of DevOps (build time, release frequency, time between new feature deployment) may be thematically similar across different kinds of organizations, the realities of different verticals, different markets, different organization sizes, different geographies, different goals and objectives and even the personalities of different individuals can mean a measurable difference in how a thematic benefit is realized within any particular organization. In short, there is really no substitute for an organization conducting a business impact assessment specific to themselves.

The Fundamentals of the DevOps Business Case

When we talk about creating a DevOps business case, it is helpful to first understand some basic concepts about how business cases are put together. An effective business case cannot avoid quantified evidence and/or projections, nor can it contain only quantitative material. Rather, a complete and comprehensive business case will contain both quantitative and qualitative content that, taken together, tell a coherent story about a current state in need of improvement. It also shows how the proposed course of action delivers that improvement and provides projections of the qualitative and quantitative aspects of the improvement. Within the quantitative content, there needs to be at least a prominent portion that quantifies business impact in pure economic terms. Again, this economic model is only one quantitative component of the business case. It should include significant qualitative material.

It is helpful to note here that a retrospective business case contains the same components, but much of the qualitative and quantitative material would be citing actual evidence of improvement rather than projections. That said, even retrospective business cases typically contain material that project future benefits, as well.

Many organizations have internal templates or examples that give a standard structure to business case documents. Creators of DevOps business cases would be wise to use these standards wherever they exist, whether they are mandatory standards or not. One of the biggest challenges in putting together a business case for a complex technical program like DevOps is making the material easy to digest and understand. Where standard document templates exist, using them means that the intended audience will already know how to read and use the document, which enables them to focus on the concepts and content included in the business case. Deviating from the standard format gives the reader/reviewer an additional job to understand the structure of your brand new document. That's definitely a taller task.

Whether your organization has a standard template or not the most crucial consideration in deciding what to include and what points to emphasize is the intended and expected audience of readers and reviewers. Typical personas that tend to be a part of a DevOps business case audience include:

» **IT management.** The IT team within most organizations is treated almost exclusively as a cost center. As such, managers of IT tend to be hyper-focused on cost-cutting and efficiency. The kinds of data points and anecdotes most likely to draw their attention are operational metrics (e.g. deployment frequency or error rate), the time duration of tasks, interoperability of technical components or processes and simplification or standardization.

Their focus is usually on improvements that can be measured at a task level, such as how much time a particular step in a process takes before and after the introduction of a new tool.

» **Development management.** Development organizations are charged with bringing as much innovation and work product to the market as possible in the shortest amount of time. Finding and hiring additional resources is notoriously difficult, so upleveling the output of existing teams is a constant aspiration of most development managers. Accordingly, the measurements that drive this group are those that speak to productivity: assessments of bandwidth, throughput, quality/rework and percentages of time spent on innovation vs other tasks. These measurements are typically expressed at the team level.

» **Business/executive management.** Unsurprisingly, members of this part of the audience are interested in higher level impacts that go beyond just the operations and productivity of the technical teams. For executives, a quick look at an organization's annual report or most recent all-hands slide deck will almost always spell out the strategic objectives and priorities they are driving, and they are always keen to understand how any initiative helps push these forward. Non-executive business managers may have a more narrow scope (a product, region, business unit rather than the entire enterprise, etc.), but they will usually have similar goals and objectives related to things like revenue or customer satisfaction. The audience wants to see connection between the DevOps initiative and true business metrics and outcomes.

Incorporating The Three Pillars of DevOps Business Value

With these three key audiences in mind, the creator of a DevOps business case should always include components that speak to the three pillars of DevOps business value.

They are:

1. Operational efficiency
2. Resource optimization
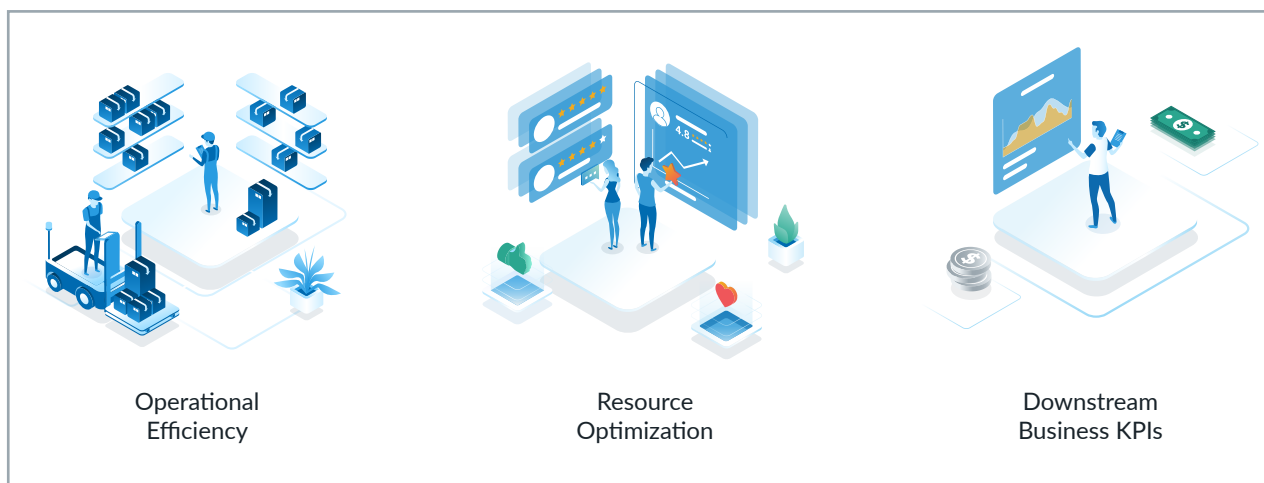3. Downstream business KPIs

Operational Efficiency   Resource Optimization   Downstream Business KPIs

Figure 1: The three pillars of DevOps business value

Pillar 1: Operational Efficiency (IT Management)

This category is going to be most relevant to the IT management persona and focuses on reducing the amount of time spent on providing platforms and spaces for work, as well as the the amount of time development and delivery work takes in both effort and duration. Here we would look to highlight qualitative and project quantitative improvements in automating manual tasks, reducing duplicative tasks, eliminating unnecessary steps in a process and other improvements of this nature. Note that all of these examples would be applicable at the task level, which is exactly where the IT management audience is looking for possible improvements.

Pillar 2: Resource Optimization (Development Management)

The resource optimization category speaks directly to the development management portion of the audience, but in actuality this category speaks to exactly the same improvements as those in the operational efficiency pillar. The difference lies in how these improvements are expressed. In the operational efficiency section, improvements would be expressed at the task level and measured in units of time and effort. In the resource optimization section, these improvements would be rolled up across an entire team or department and expressed in terms of percentage improvement, the slice of a team pie chart dedicated to innovation or saved time (hours or days) across a team.

Pillar 3: Downstream Business KPIs (Business/Executive Management)

Most DevOps business cases we see include some semblance of the first two pillars, but it is the rare case that includes this third pillar. And that's a shame, because this is the pillar that is dedicated to the business/executive management portion of the audience, and that is the portion of the audience that most frequently makes decisions on greenlighting initiatives and investment! This section focuses on projected or observed impact to specific business KPIs - the list of KPIs. The list need not be comprehensive, but should at least be presented as representative. We have some suggested approaches for this in the next section of this paper.

Pulling the Business Case Together: How to Approach It

In the work CloudBees has done helping organizations put together business cases in support of their DevOps initiatives, we've had a lot of opportunity to see how to successfully approach each of the three quantitative pillars.

The Overall Document

As noted previously, one of the biggest challenges in creating a successful DevOps business case is in being able to make a case for a program that is inherently technical and complex. You must distill that down into something digestible and understandable for a mixed technical and non-technical audience. Structuring the document such that a high-level reviewer can easily derive the big takeaways from the content - without getting lost in the details - is the best way to ensure that your case is fully understandable by the entire audience.

How do you achieve this, in your final deliverable? An effective approach would include two parts:

» **A deliverable that uses slideware to tell the main thrust of the qualitative narrative** and highlights the most important data points and conclusions of the qualitative analysis. This is true whether or not an actual live presentation to your audience is part of the review process. Even if the reviewers will only read the materials and follow up with Q&A, slideware is incredibly effective for this purpose because it forces the writer to be organized, precise, to think about sequencing and have a very sharp pair of editorial scissors to keep the content on each slide concise.

» **Beyond the slideware narrative, you should also have a very detailed quantitative document.** You may also want to include a technical appendix for items such as complex architectural diagrams and busy process maps. These items can make big picture comprehension more difficult, but also enable technical-savvy readers to drill down more deeply, if desired.

The Three Pillars: Quantitative Sections

Content for each of these three pillars should be a part of any quantitative narrative with its own set of best practices. As you put together your business case, keep in mind these principles for making and displaying calculations for the pillars. We will go into some detail, below, for each.

Pillar 1: Operational Efficiency (IT Management)

For this section, it's important to be as granular as possible. Recall that the IT management audience to whom this portion is directed is usually most interested in task-level improvements, such as time spent on a particular task. Getting granular improves the effectiveness of this section in several important ways:

» You meet the needs and expectations of IT management, who are always a key portion of the business case audience.

» You display thoughtfulness and expertise around the process that shows you've done your homework. That lends credibility to the entire case.

» You make it easier to show changes in task-level behavior over time. Many objections to operational efficiency calculations center around reviewers not believing big improvement is realistic. When you, instead, show positive, incremental improvement to granular tasks over time, it is harder to object to larger improvements that, in turn, show larger benefits.

Operational efficiency task-level example

CI to CD Handoffs	Before	After
Number of changes/promotions per month	504	504
Number of minutes per CI/CD handoff	20	5
Number of hours handing off per month	168	42
Handoff time cost per year (Assumes a very conservative hourly rate of US $53.00; the actual cost for a developer, including benefits, is much higher.)	$106,848.00	$26,712.00

Table 1: Example of per-task improvement

Note that the per-task improvement in Table 1 is actually quite modest (15 minutes), but the volume of instances means that these small drops start to fill up a sizable bucket over the course of a year. If you think about an operational efficiency model that includes 15 or 20 such line items, an improvement of this size adds up to a $1.2 - $1.6 million impact. That kind of a figure almost always turns heads.

Pillar 2: Resource Optimization (Development Management)

As noted earlier, the resource optimization section really covers the exact same ground as the operational efficiency section, except that the way in which the data is framed and presented is changed to better suit its intended audience - in this case, development management. As a consequence, the most important thing to remember is to roll up improvements to a team or department level, and to express savings in the number of hours or days saved across the team. You can then add up all of the team-level improvements to show movement in a pie chart that shows the slice of the pie dedicated to innovation getting bigger (or conversely the size of the slice dedicated to administrative work getting smaller).

CI to CD Handoffs	Before	After
Number of developers	40	40
Total development time per month	7,040	7,040
Number of hours handing off per month	168	42
Percentage of time spent handing off	2.39%	0.60%
Handoff time cost per year (Assumes a very conservative hourly rate of US $53.00; the actual cost for a developer, including benefits, is much higher.)	$106,848.00	$26,712.00

Table 2: Example of percentage of time saved for a task

Resource optimization example

Here we see the exact same task level benefit as we used in the operational efficiency section, just turned on its head to be expressed as a percentage of time saved across a team. Again, if you think of this as one of 15-20 similarly impactful line item improvements, the percentage of time spent on administrative tasks would drop by an absolute figure of 26-36 percent. Think of the increased innovation development teams could bring forth with that kind of reclaimed time!

Pillar 3: Downstream Business KPIs (Business/Executive Management)

This is the section that is almost always missing from business cases our customers share with us when soliciting our feedback. Given the tendencies of DevOps managers noted in our section on the state of the DevOps business case, it's really not a surprise. These effects, impacts and benefits occur by definition outside the purview of DevOps and most DevOps managers lack curiosity about, expertise with or visibility into the downstream implications. In the end, though, the logic is simple and anyone should be able to connect the dots to downstream business impact as long as they solicit input from the right kinds of resources.

The logic of the downstream impact of DevOps improvement on business KPIs goes like this:

» Increased efficiency necessarily leads to increased throughput
» Increased throughput means more. More features, more fixes, more capabilities
» For any application, delivering more stuff in less time means the impact of the application itself becomes greater

Most applications that development teams build have an intended purpose and have an impact they are trying to make. Think about an e-commerce application, for example, and it is fairly intuitive that its purpose is to drive revenue through online sales. If improvements and new capabilities for this application are introduced more quickly, the amount of revenue driven through the application would be expected to increase.

Downstream business KPIs example

Let's look at a simple example from real work we did in collaboration with a customer in the auto insurance industry. Here we are looking at an application meant to improve the customer claims experience. One of the KPIs this application should impact is customer retention - that is, the percentage of policies that get renewed in a given year. This is a very important metric in insurance, because the cost of acquiring a new customer is far greater than the cost of simply processing a renewal. This is true for many other industries, as well. The hypothesis is that if the claims experience is a good one, policyholders are more likely to renew.

In a conversation with the product manager for the claims application, we were told that the product team identified 10 features they wanted to include in the next release and that the development team was able to commit to complete the top seven of those 10. That led the product manager to project an increase to customer retention of 0.2 percent:
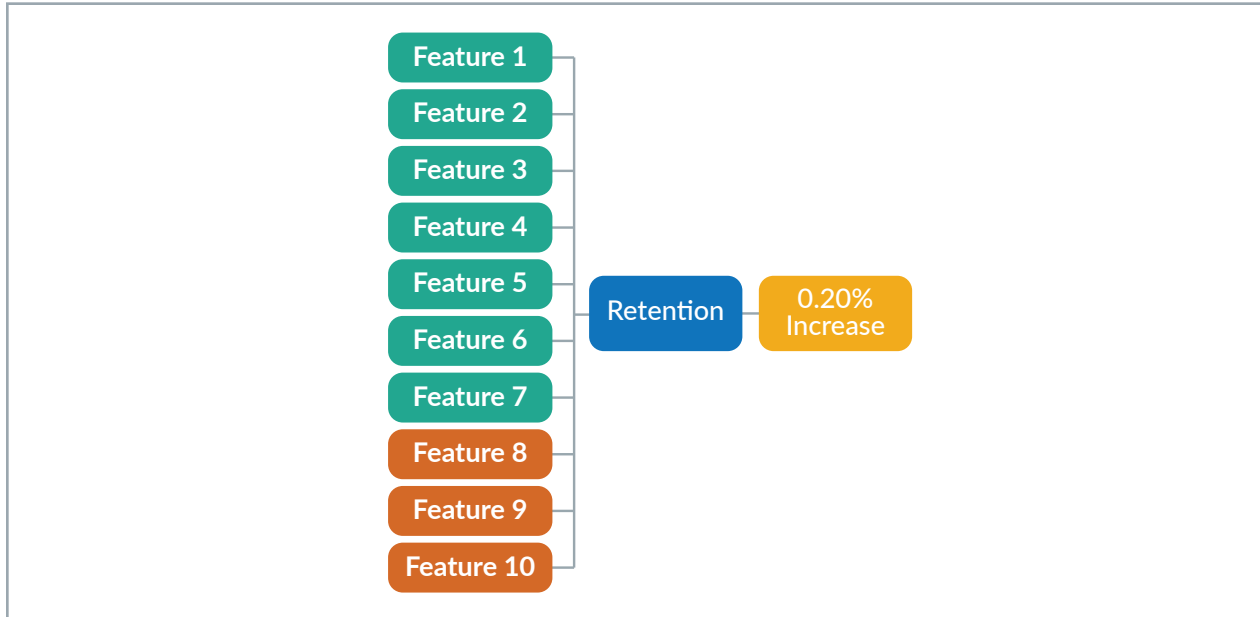

Figure 2: Number of features to be included in the next release

We then asked what the additional impact would be if development throughput was increased such that all 10 of the requested features could be completed instead of just seven. The response was that we might see an additional 0.05 percent increase, for a total increase of 0.25 percent:
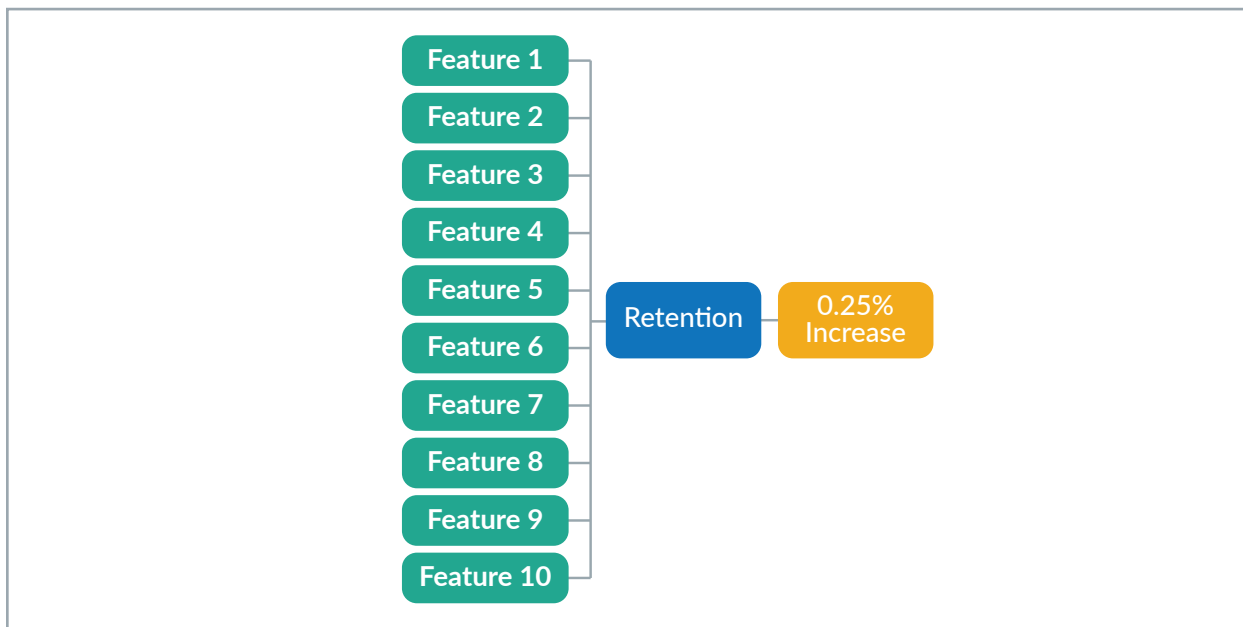

Figure 3: Increasing the number of features to be included in the next release

That extra 0.05 percent may not seem like a lot, but the beauty of calculating the business benefit of impacting a downstream KPI is that a modest increase can translate to big economic impact.

Here's a quick calculation of the dollar impact of this improvement

Net written premium on 18,186,000 policies	$27,100,000,000.00
Loss ratio plus loss adjustment expense ratio - 2019	73.10%
Underwriting expense ratio - 2019	20.30%
Profit (dollars) after loss ratio and underwriting expense	$1,788,600,000
Average premium per policy	$1,490.16
Initial retention rate increase	0.20%
New retention rate increase	0.25%
Net retention rate improvement	0.05%
Impact of additional retention	**$894,300.00**

Table 3: Dollar impact of increasing the number of features

A couple of notes for those who may not know much about insurance: Loss plus loss adjustment expense (LAE) ratio shows the percentage of premiums paid out as claims plus the percentage of premiums spent processing those claims. Underwriting expense (UE) ratio is the percentage of total premiums spent on acquiring new policyholders - things like advertising and agent commissions factor in here. Broadly speaking, the percentage left over after taking out these ratios is profit.

What the table above shows is that a 0.05% increase in retention leads to keeping thousands of policies as renewals rather than losing those policies to other carriers and results in $894,300 in additional profits.

Now, $894,300 would serve to offset a pretty big chunk of the cost of any DevOps initiative by itself, but consider this: What we just showed above represents one KPI impacted by one application. At almost every organization there are dozens, if not hundreds, of applications - each of which has multiple business KPIs they are designed to impact. The visual representation of the opportunity for business impact probably looks more like this:
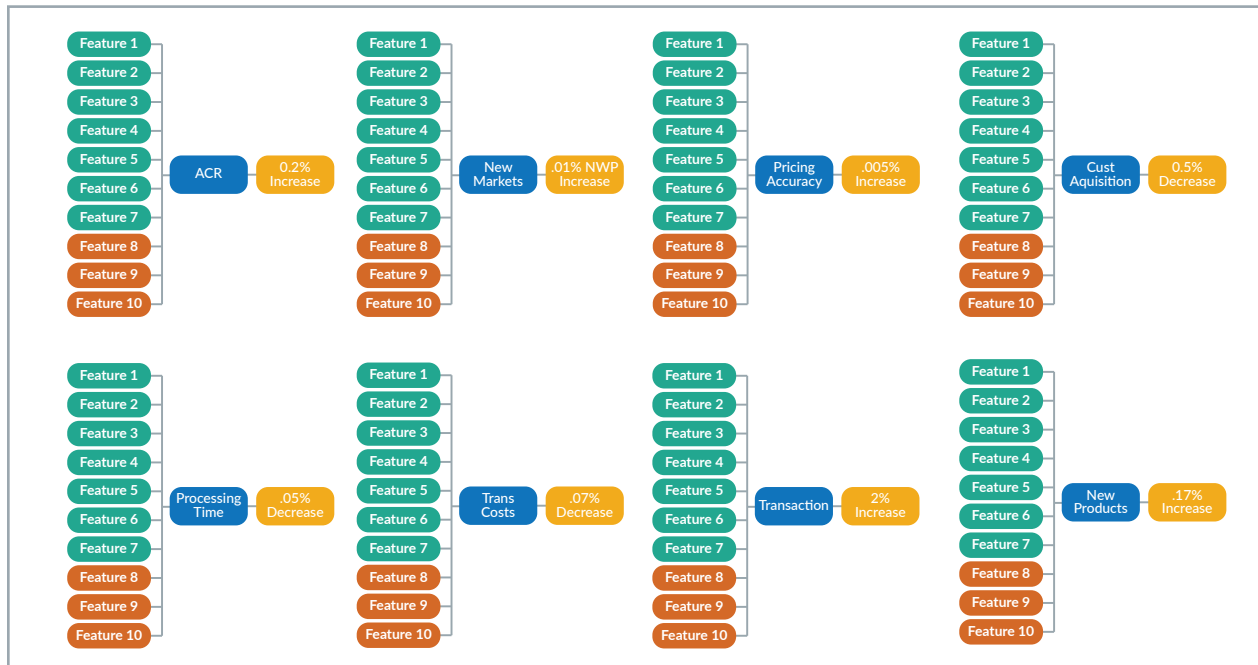

Figure 4: Visual representation for business impact potential

...and so forth.

That translates to millions of dollars or maybe hundreds of millions of dollars or even billions of dollars in impact. A good business case should never leave this section out.

You Can Do This: Getting Started

The reality is that while putting together a DevOps business case requires a good number of steps and principles, none of them are especially difficult to execute, even when business isn't your first language. In our work with hundreds of customer organizations seeking help with their business cases, it's always satisfying to see the lightbulb go on when a highly technical individual we're working with realizes that "Hey, I can really do this!" We feel confident that you can, too.

If you keep in mind all of the necessary components for an effective business case, you will absolutely put together a business case seen as timely, relevant and effective. To summarize, these components are:

- » Involve the right sources to help gather the necessary information for each component
- » Recognize the audience for your materials
- » Always, always remember to keep the main body of your document digestible and understandable for any audience

As you start to put together your case and your materials, the business consulting team at CloudBees is always available to take a look, provide feedback and even facilitate discussions within your organization to gather the necessary information to form the foundation of a great business case. Working with organizations to describe the value of DevOps is what we do - it's what we live and breathe everyday - and we'd love to share our insights with you.

Appendix A: Outline of a Business Case Framework

In this section, we've provided a list of possible line items to include in a business case across all three of the quantitative pillars. This is not meant to be complete or comprehensive, but is an extensive list of some of the most common and effective items we've seen across the hundreds of business cases we've worked on with CloudBees customers.

Operational Efficiency

Time and effort spent on:

- » End user support
- » DevOps system maintenance
- » Security patching
- » DevOps tool version upgrades
- » Monitoring DevOps container/machine performance
- » Provisioning and adjusting DevOps computing capacity
- » Managing user rights/roles/permissions
- » Creating new projects
- » Creating new pipelines
- » Providing high availability of DevOps resources
- » Troubleshooting DevOps infrastructure errors
- » Doing manual handoffs between automated steps
- » Creating and modifying workflows
- » Producing and reviewing operational reporting
- » Conducting required compliance activities
- » Supporting audits
- » Performing release administration
- » Managing release environments
- » Conducting rollbacks of deployment errors
- » Preparation of release windows
- » Managing releases to subsets of users
- » Executing feature toggles
- » Monitoring feature performance

Additionally, be sure to include the impact of the above on spend on cloud and/or on-premise compute resources. For example, a reduction in time for a process to run multiplied by however many times that process executes in a week, month or year may also translate into less money spent on compute resources.

Resource Optimization

» Administrative time spent by team/business unit/organization

» Manual work by team/business unit/organization

» Percentage of developer team time spent on development and/or innovation

» Deployment frequency

» Number of features delivered per sprint/release/day/week/month

» Total hours of manual work

» Total time saved by automation

» Output per team/business unit/organization

» Duration of team/project onboarding

» Total time spent on DevOps support tickets

» Average wait time for DevOps resources

Typical Downstream Business KPIs

Note: the best KPIs to include vary greatly by industry and even by individual organization. Some great resources to find the most important KPIs for your organization are annual reports and company update meeting slides. That said, here are some of the most common KPI categories:

» Revenue
» Customer satisfaction
» Transaction volume
» Transaction cost
» Market share
» New product performance
» Net Promoter Score
» Number of customers/accounts
» Number of advertising impressions
» Number of customer calls/contacts
» Customer acquisition cost
» Processing time

Learn More

Watch the Video
Connecting Software Delivery to Business Outcomes

Read the eBook
The Definitive Guide to Modern Software Delivery

Explore the Case Study
Morningstar Powers Market Momentum with CloudBees Platform

Learn More Today!
www.cloudbees.com

CloudBees, Inc.
4 North Second Street | Suite 1270
San Jose, CA 95113
United States
www.cloudbees.com
info@cloudbees.com

0122v00