

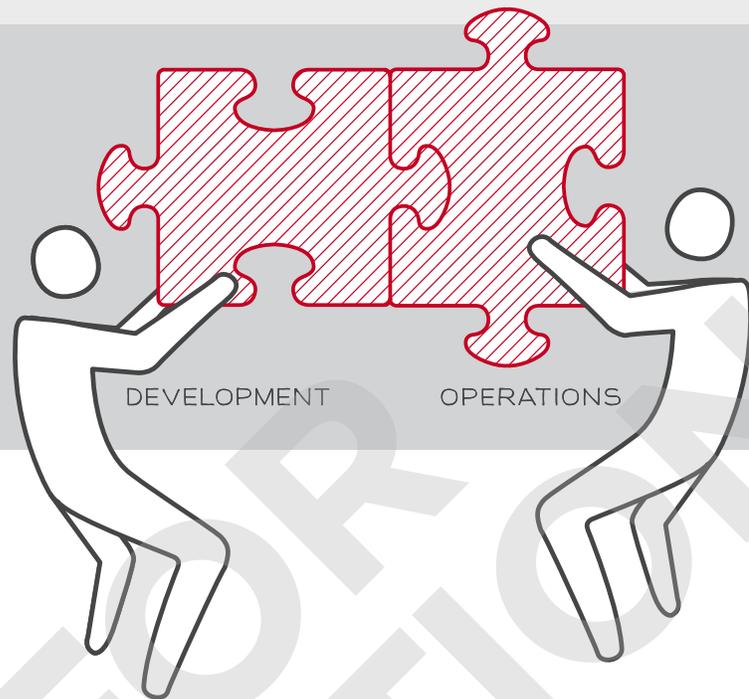
The DevOps Mindset:

Real-World Insights
from Tech Leaders



BROUGHT TO YOU BY

 **rackspace**
the #1 managed cloud company



INTRODUCTION

Without Cross-Functionality and Flexibility, DevOps Just Won't Work

As an IT professional, you know that reacting quickly to customer demand and achieving a faster speed to market is a top priority. This kind of agility requires a continual cycle of release and adjustment, and keeping up with it depends on how well your Development and Operations departments can collaborate. Achieving the necessary collaboration isn't always easy. But it is out of the necessity for it and the efforts required to achieve it that the DevOps movement has risen.

The speed-to-market benefits of DevOps are becoming widely known, and more companies are interested in adopting such a system as part of their own infrastructure - if they haven't done so already. You might think some new piece of hardware or software is what you need. Yet the agility that companies seek isn't a tangible product. "You can't buy DevOps in a box," explains Kevin Behr, author and Chief Science Officer at Praxis Flow.

Certain facets have to be in place for DevOps to function, and the number one factor is having a company culture in which it can thrive. To understand the importance of culture, consider how DevOps can function within an organization: "Basically, it is both a social system and a technical system - a Socio-Technical System," Behr elaborates, "and what DevOps does is bring the social more in balance with the technical than it has been in the past."

The culture that can foster the DevOps mindset has to exist throughout a company, not just within a single department, so people can collectively contribute all their skills to solving problems. "One of the key elements of making DevOps successful," Behr emphasizes, "is actually making sure that it is part of an organizational plan. Organizations that want to be successful with DevOps typically transition away from managing tasks to managing

boundaries cross-functionally. Otherwise, DevOps can become a local optimization.”

If you want to transcend the status quo and reach new standards, you must be willing to do what it takes to stay competitive. “Flexibility is the key,” Behr says. “You need that flexibility not just because of choice, but because, now more than ever, the market is dictating it.”

So, How Do You “Do DevOps”?

We’ve established that DevOps won’t lead to any significant results if the right circumstances aren’t in place. So how do you achieve the right circumstances? You must remove silos that prevent your teams from optimizing their collective tech strength. You need to clarify your goals and limitations, and also strike a balance between technology and employee interaction.

“The key thing to be thinking of here is the mindset angle of DevOps: the collaboration, the measurement, the sharing,” says Behr. “By balancing the social and technical side of things, you can actually get much faster feedback loops in your learning process and in your development and operational processes. You can actually learn, advance and iterate much quicker. But if you don’t develop those sides equally, what you get is automation without any thought or a lot of collaboration without any real traction toward achieving your company’s goals.”

Rackspace contacted several technology leaders with DevOps experience to talk about the trials of adoption and the successes of implementation. This ebook showcases their perspectives, examines challenges and achievements, and offers suggestions these experts recommend for other organizations looking to adopt the DevOps mindset.



Jim Kimball

Chief Technology Officer

HedgServ

Location:

New York/Ireland

Company snapshot:

HedgeServ is a global, independent fund administrator.

Size of organization:

App. 700

KPIs tied to DevOps:

"None. Our DevOps initiative is not about KPIs. It is about how we work."

Q&A with Jim Kimball

Why did you decide on DevOps? What was the catalyst to get you started down the DevOps path?

We had been doing monthly releases since we started our business. As we grew, we kept doing monthly releases even though they started to involve more and more ceremony with more and more people. Finally, during one release we had issues that got us all in the room together and our CEO just said enough was enough. He kind of woke us up to the fact that we had outgrown our process. We were forced to look at it fresh.

What are the essential components of DevOps?

For us, it is an understanding of the different perspectives of all the people involved. I come from a development background, and I admit I didn't really look "over the wall" to understand what was going on with our tech guys. It turns out they were scrambling to deal with whatever everyone else in the organization threw at them. And in the process they were forced to learn how to adapt in a world where they had little to no control.

When something broke, it was so easy to point it out after the fact, and I was as critical as anyone, asking things like, "How could this happen?" without understanding that it was unfair to ask this of the people being swamped by it.



"If senior management has a desire to work differently and an understanding of the challenges of the people doing the work, DevOps can happen."

One aspect of embracing DevOps is we learned that we had created the system that caused this, and so we were then in a position to start to correct it.

How long did it take you to get from beginning the transition to the point when you were doing DevOps successfully?

I think the fundamental shift toward DevOps started when we got away from focusing on individual team goals and elevated our conversation to organizational goals and let the teams drive toward them, probably a little over a year ago. I suspect that could be called the beginning of our DevOps journey.

And now we have learned to communicate these goals in the form of intents, with leaders of cross-functional teams able to determine how to achieve those intents. It has been amazing to watch this happen. We still have a long way to go. We are still experiencing frustration from people wanting and trying to do good but not knowing how. When that goes away, I would say we are doing DevOps successfully.

Are you seeing an improvement in cycle time, go-to-market speeds, customer satisfaction? If so, how are you measuring and tracking the improvement?

We are exploring ways in which we can measure our improvements, but I'm not really satisfied with the measures I've seen and how they might apply to us. My current observations of success are drawn from the behavior I see in the people in our technology group, how their attitude is changing. For example, the programmers have a blog where they describe how failures occurred for all to learn from. We are just now wrapping up our new server build, which was a combined effort between programmers and IT. While that in itself is awesome, the even cooler thing is the fact that a whole new avenue of conversation has opened up.

We recently had a case where one of our libraries was causing trouble when we had multiple versions of our main application installed on the same server. The programmer investigating reached out to the IT team to ask about migrating this library into the base build instead of being packaged with the app. That solution path would never have been explored a year ago.

Did you run into roadblocks when you began your DevOps implementation? If so, how did you navigate around them?

I would love to say that the journey has been smooth, but there are roadblocks everywhere. You want to just point out to someone, "Hey, look at this—here is a better way to work that will make your life more rewarding" and they just jump at it, but it doesn't work that way.

I have been very fortunate to have a team of leaders who are willing to try things with me and learn with me. Together we have worked with our teams to ease people along the path to where we want to go.

Are repeatability and scalability important to your organization? How does doing DevOps help you achieve those things?

Repeatability and scalability are our business. The capabilities that DevOps gives us directly enable this by encouraging collaboration within the organization to come up with better solutions.

The library example I gave above would have been solved by the developers a year ago. It's not like the problem wouldn't have been solved prior to our DevOps experiences, but it would have been solved within the team silo and the solution would have been fragile.

I bet money that a solution developed the old way would have worked for three years until one of the assumptions it made bit us. Getting more robust solutions to all of the little problems is how DevOps helps us with both scalability and reliability.

What DevOps myths ("we're too big," "we're not in Silicon Valley," "what we do isn't scalable") have you personally dispelled in your transition to DevOps?

Perhaps the biggest one for us is that we are a Windows shop, and there has traditionally been a bias toward Linux as the one true DevOps platform. I can say with certainty that is not the case.

What would you recommend to another company interested in beginning a DevOps implementation?

I would say it is okay to not know the answers. I feel one of the biggest challenges in IT can sometimes be working with groups within and outside your organization who expect certainty. Even if you don't know the path, learn the goal and begin the journey.



James Kenigsberg

Chief Technology Officer

2U, Inc.

Location:

Landover, Maryland

Company snapshot:

2U, Inc., specializes in online university programs offered through a cloud-based software-as-a-service (SaaS) platform.

Size of organization:

App. 575

Annual revenue:

"\$83 million for year ended 12/31/2013."

KPIs tied to DevOps:

"The number and speed of deployments; number of contributors across the board who can work and contribute to our source code."

Q&A with James Kenigsberg

What's the first step toward beginning a DevOps collaboration?

The first step is opening up the lines of communication between your teams. Remember, the term DevOps is a blend of developers and operations, bringing two silos together. So to achieve true DevOps collaboration, you need your employees to really think and act as one, not just be merged together in name only.

By pushing communication from the start, everyone gets a better feel for others' needs and how they do their jobs. Then they take those things into account while doing their own jobs — working not just for themselves.

What are the essential components of DevOps?

- **Automation:** If you're doing something more than twice, you should think about converting it into an automated task. Automation ensures your process is repeatable and reliable; it standardizes the execution of the task to the best way every time, without any risk of deviations from peer-reviewed code to improve the process for the whole team.
- **Transparency:** Transparency gets employees from these groups to take an intimate look into what the others are doing, improving communications and business processes for all, not just for those in your department.
- **Talent:** You want employees who put business needs, efficiencies, and automation before their historical knowledge of how hardware works, and don't draw a line in the sand between IT and developers. Before digging in to solve a problem, they talk to co-workers elsewhere who may have solved the problem in the past. Above all, they are working with the goal of the entire company in mind, not just the tasks at hand.

What kind of culture existed in your organization before you transitioned to DevOps? Did it change? If so, how did you facilitate that change?

From the beginning, our company culture relied heavily on great communication. We had a few employees wearing many hats, but everyone collaborated together extremely well so that there was never a disconnect. I think this was paramount when we began our DevOps shift because we already had a sense of transparency. Our initial employees were able to help breed the DevOps culture as our staff grew because they had been thriving in it already for some time.

"To achieve true DevOps collaboration, you need your employees to really think and act as one, not just be merged together in name only."

Is there a specific approach that will make the transition to DevOps less painful?

The team that trains together stays together — your teams should constantly cross-train across various departments. Your developers should know how QA departments work, and your IT operations group should be aware of the needs of the developers. Employees in general must be aware of everyone else's needs so they can work in silos but still have the transparency and communication necessary to be productive.

What DevOps myths ("we're too big," "we're not in Silicon Valley," "what we do isn't scalable") have you personally dispelled in your transition to DevOps?

We aren't located in Silicon Valley. Our company headquarters is in Landover, Maryland, but our tech team is in New York City. So we don't have that "Silicon Valley" feel that everyone talks about, yet we have still been able to attract top-tier talent and incredible workers because they believe in our company mission and what we are doing at 2U.

You don't have to be in Silicon Valley to embrace the DevOps culture; anyone can do it from anywhere if you have the right people. In fact, many new Silicon Valley firms these days have a bad habit of building shiny technology because it is shiny, which goes against the DevOps principles of communication and transparency. It shows that you aren't listening to the needs of the business, or being open with the business about your development goals.

What would you recommend to another company interested in beginning a DevOps implementation?

Align your tech department with your business needs. We have areas specifically aligned with the business (those responsible for learning tools are in Learning Systems; those who run the backend are on Business Systems).

Each area is fully loaded with a full stack of every role needed for a full software development lifecycle. Because employees are aligned to the business mission, it makes them not just people completing tasks, but turns them into a nucleus of your company to achieve your mission. It's not about the amount of code they write or tickets that they close; it becomes their goal to achieve the mission of the company.



Jeff Hackert

Engineering Manager

Riot Games

Location:

Santa Monica, CA (HQ)

Company snapshot:

Riot Games is a game studio that develops and produces multiplayer online games, including *League of Legends*.

Size of organization:

App. 2,000

KPIs tied to DevOps:

"We have KPIs for any endeavors, but we don't think about projects with that moniker. We think of DevOps as a culture topic more than as a class or designation or project."

Q&A with Jeff Hackert

Was there a particular point in time that you "started doing DevOps," or has that always been part of the culture there?

No, it has not always been part of the culture. When I joined the company a few years ago, we were 400 people. In the last two years, we have grown to nearly 2,000, so the growth has been substantial. That growth has been matched by growth of our infrastructure. So the original way the infrastructure had been growing before I had come on at Riot, Ops was cut off from developers and the rest of the organization, and there were a few intermediary organizations that became primarily responsible for the build of our software and a team that was responsible for deployment. They were not organized together in any meaningful way, and they were not communicating.

Was there a particular catalyst at Riot Games that got you started down the DevOps path, or how did you come to do things in a more collaborative way?

When I got to Riot, there was a culture of certain personal heroics in operations, but the cost was quite high and the resulting silos turned into some real challenges. The team that I started working with that really had a lot to do with this change was working with infrastructure; they were working specifically with Chef, but it was a small team of software developers working to implement technology that both developers and the operations team needed to participate, and there was no scenario in which the silos could remain and we could reach any kind of definition of success. So this idea of building up or bringing DevOps into the organization was just a natural extension of those activities.

How did you facilitate that change? How did you get people to come out of their silos and be more collaborative and communicative?

I will say that I don't know that, from my own side, I can claim responsibility for having achieved those things. What I can say is that I definitely brought that conversation out into the open and forced real reflection around their practices simply because I was standing in the middle between multiple parties and had the opportunity to force that conversation.

Among the people that I worked with, I actively confronted a number of assumptions that underlay how we were doing our day-to-day work. These conversations translated into slogans like, "No snowflakes." The idea of the no-snowflake environment is that software engineers and systems

"We had to flip the model, from the idea of personal heroics and siloed responsibilities to real, cross-functional collaboration."

engineers would not log onto a single box where there are a thousand boxes that all shared a set of attributes. We would not log onto a single box in production and alter its configuration, no matter how beneficial that change seemed to be. Instead, we would focus on cookbooks or recipes and the software expression of that node, or that group of nodes, as a way to resolve it.

Do you feel that you are “doing DevOps?” Would you describe the environment, the culture, there as a “DevOps shop?”

Yeah, after nearly two years of very active conversations, we no longer have what you would call a traditional Ops organization. By any sort of objective measurement of what DevOps is from a culture perspective, we lie in that model. But we do not refer to ourselves as doing or being DevOps.

The people who are doing DevOps often aren't saying that they're doing it. So that's definitely in line with what we've heard from other people.

Making DevOps a tool issue, or making DevOps a set of defined practices, will only get you where we have gotten with every other system like this, which is that we violate what we care about most. So if you're hiring smart people, we're breaking the value of those hires when we limit or constrain their creative intelligence in terms of problem solving.

What needed to happen for Riot to get where it needed to go, to be able to manage at the scale we're running at, which is pretty big, we would want not to constrain artificially the creative intelligence of the people working in that system. We want the purest expression of people's creative intelligence.

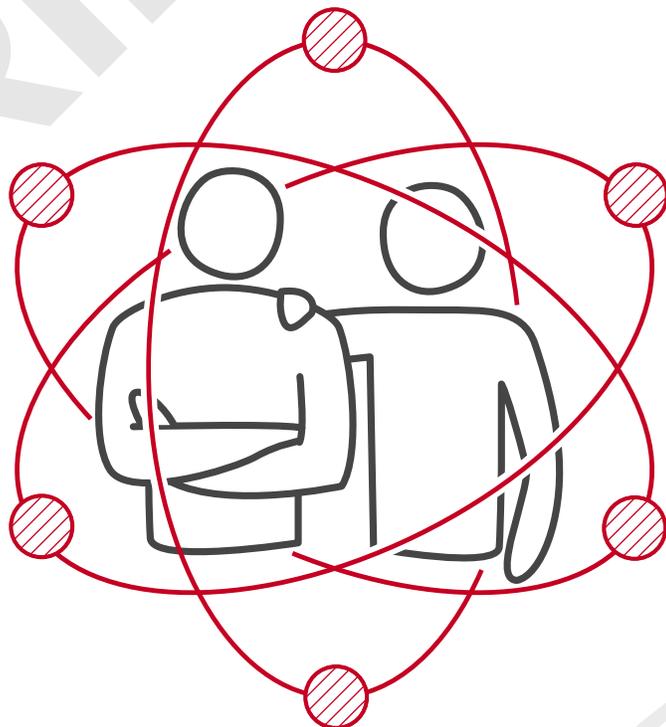
We want to honor and respect their abilities, not demean them by putting artificial labels on practices they should define and hold. However, we do

want to create a culture in which they're solving real problems. We had to flip the model, from the idea of personal heroics and siloed responsibilities to real, cross-functional collaboration.

What might you recommend to another company that is going through the same sort of growing pains that Riot Games did?

That they have faith in their hiring practices, and that they have real empathy and compassion for the people they've hired. When you execute on a really good hire, which should be all the time, every you bring in is bringing a set of experiences and a set of goals that they want to achieve. So you don't want to stand in their way. What you want to do is to guide or facilitate that creative output to solve the real problems that your business is facing. If we're talking about operations and infrastructure, the idea of DevOps unlocks that, but it is not a set of proscriptions or a recipe that will give you a defined best practices outcome.

Things like best practices will probably end up hurting you more than they will help you. Your best practice is the collective wisdom of the very smart people you've hired to solve your problems. And if what you have is a siloed or broken environment, you want to increase the communication, not impose new rules.





Bharat Krish

Vice President, Information Technology

HBO Latin America

Location:

Coral Gables, Florida

Company snapshot:

HBO Latin America is a cable television network serving Spanish-speaking audiences in Central and South America, and the Caribbean.

Size of organization:

App. 800

KPIs tied to DevOps:

"None specifically tied to DevOps."

Q&A with Bharat Krish

Do you describe yourself as "doing DevOps?"

We gather requirements, we design, we develop, we deploy and we score it. It's end-to-end product development. We have an internal software development organization and a digital products organization. We also employ development consulting companies outside in helping the effort, but right now we lead the effort.

And do these teams work seamlessly with each other, or do they each have their defined responsibilities with no crossover?

There's a considerable amount of overlap. It's a startup culture, and a lot of times people roll up their sleeves to get things done, so you might find my software developers who develop a product supporting the product as well. If they don't have involvement in support, they may not know the context behind what they're building—they wouldn't know how to enhance the product. So it's intentional that I encourage the culture of overlap.

Did that overlap exist before, or is this something you've encouraged? How did that culture come about?

It has been an evolution. The company has grown considerably. When a company starts out, everybody's doing everything, but over time we had built a number of teams, and now it's become more of an institution that we have to have them overlap, and it's part of our operating procedure. It's institutionalized now that if you build something you get to lead it, and you also get to support it.

It's a cultural aspect that I continue to enrich all the time. In a large organization, silos are built naturally. Part of my job is to break those silos on a constant basis and establish that overlap.

You talked about the developers being involved with the support of the product. Does this lead to better end products and faster improvement and therefore an increase in your customer satisfaction?

The nature of our business is that the products are evolving so fast, we see it as building a quality product that meets the needs of the brand. In the case of HBO, a premium entertainment brand, it needs to have a premium-quality experience, whether it's user experience or graphical representation or content. We're consistently striving for a quality that will be measured on how the audience lives the brand.

"In a large organization, silos are built naturally. Part of my job is to break those silos on a constant basis."

Do you feel that your developers, because you give them that ownership, are able to live the brand and really care about it?

We want everybody to be successful, and that's a baseline, but mistakes are encouraged. I always tell my team we're not running an emergency room here, so don't be afraid of making mistakes. It takes multiple iterations to build something that has the quality of our brand written all over it, and it also takes multiple iterations to build something that's simple.

It's in the DNA of most of my team members that they are willing to take that step and propose ideas and not have the fear of people thinking that the ideas may be stupid or not encouraged. When I think of a DevOps culture, for me it's building a foundation where development can happen in a more agile fashion, so I focus on the foundation more than anything else.

How do you focus on the foundation? Is that through your hiring practices, through training?

Multiple things. Just a week ago, we did a workshop with all my managers and part of that workshop was to reiterate the culture that we need to build and maintain within the organization. I look at it as five things: Keep things simple. Focus on managing people, supporting them, directing and training them. Be humble. Respect everyone. Have integrity.

By creating those core values, the expectation is, we will have a lot more people talking to each other about new ideas, not afraid to make mistakes and being honest about the work. In this sense it's going to reflect in a great product, in tackling a lot of the competition in our industry or how consumer behaviors change, being able to think out of the box and come up with solutions.

That's the culture we're promoting, and it's working. There are times we do have things that we need to fix, and new employees bring a lot of baggage with them that needs to be reset. It's a work in progress, but overall from my experience working with multiple companies, I've cracked the nutshell on trying to be more innovative because of the cultural values that we've implemented.

What would you recommend to another company who is interested in beginning a DevOps implementation?

Do not look at software development or support as a playbook that you can bring wherever you go. You have to first understand the culture of the company, the people, the customers you serve, and it also depends on the

region. Latin America is very different from the U.S., which is very different from Europe, which is very different from Asia.

All that needs to be understood and also understand the senior management, what are the strategic revenue opportunities they're thinking about, and based on that, hire the right people, build the right culture, and track the right metrics.



SUMMING UP

Creating a Path Toward Continual Collaboration

As with the tech experts portrayed here, you'll have your own reasons for moving from a siloed IT department to an agile DevOps model. You might recognize you need to achieve greater efficiency or want to reach new levels of productivity. Whatever the motivation, your company's culture will be the deciding factor in whether DevOps will be successful for you.

To do DevOps, you don't need to create a new department, but you do need to take a look at the culture on your team and throughout your organization. Is your company maximizing your IT professionals' collective skills and expertise? If not, is that because your company's established IT conventions are limiting productivity and therefore growth? These are tough questions and answering them will take time and commitment. You have to answer these questions, though, to start the conversation and to address the issues at stake.

The agility that's necessary for continual software deployment must be based upon a company infrastructure that encourages IT departments to see themselves as part of a larger team. And you'll never build that infrastructure without creating the path toward the right mindset.

ACKNOWLEDGEMENTS

Rackspace would like to thank all contributors for lending their valuable time and insights to this project:



Kevin Behr

Chief Science Officer
Praxis Flow
Co-Author of *The Phoenix Project*



Jim Kimball

Chief Technology Officer
HedgeServ



James Kenigsberg

Chief Technology Officer
2U, Inc.



Jeff Hackert

Engineering Manager
Riot Games



Bharat Krish

VP, Information Technology
HBO Latin America

This eBook is for informational purposes only; all statements and information are provided "AS IS" and are presented WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

Rackspace is either a registered service mark or service mark of Rackspace US, Inc. in the United States and other countries.

Third-party trademarks and tradenames appearing in this document are the property of their respective owners. Such third-party trademarks have been printed in caps or initial caps and are used for referential purposes only. We do not intend our use or display of other companies' tradenames, trademarks, or service marks to imply a relationship with, or endorsement or sponsorship of us by, these other companies.

SHARE WITH YOUR COMMUNITY

