Atlassian
JIRA Service Desk

How to Enhance IT Support with DevOps

(PART 1)

About the Author

John Custy

President,
JPC Group

John Custy, President of JPC Group, is a 35-year IT veteran, and service management authority. John has helped organizations improve service delivery, become more strategic, and increase value. He's an internationally known speaker and educator challenging the IT world to think differently.

John's achievements include:

- The Ron Muns Lifetime Achievement Award
- The Cherwell IT Industry Legends Award
- ITIL Service Manager
- ITIL Expert
- ISO 20000 Consultant
- KCS Consultant
- Kepner-Tregoe certified
- DevOps Certified Instructor

He's also a member of the HDI Faculty and the HDI International Certification Standards Committee.

Table of Contents

High-performing IT Organizations

What do they all have in common?

They're more agile

Teams are innovating faster than ever, and deploying 30x more frequently—with 200x shorter lead times.

They're more reliable

By streamlining their process and fortifying infrastructure, high performing teams have 60x fewer failures and recover 168x faster.

They're winning

Teams practicing DevOps typically exceed profitability, market share and productivity goals.

Source: Puppet Labs 2014 and 2015 State of DevOps report, based on responses from more than 20,000 technical professionals.

atlassian.com/service-desk

Today's Situation

Rising business expectations

We live in an interconnected and rapidly changing world. Today's businesses have multiple tools for productivity, communication and work—and new ones pop up every day. A recent survey by HDI showed that the consumerization of technology is changing expectations of IT.

The business wants innovation, new technologies, and frequent updates to support competitiveness and rapidly changing business requirements. At the same time, they want their services to be reliable and stable. But change is often the nemesis of stability. The more you change something, the more likely you'll break it. In fact, 80% of all incidents are due to changes (Gartner).

But who is expected to deliver more reliable services? Development teams, of course. Agile and lean methodologies changed how we develop software in the early '00s. Ever since, developers have been tasked with delivering features fast and frequently while IT operations and support must maintain infrastructure stability and availability. Customers want both. And therein lies the conflict.

To achieve this, we need to change the way we handle change.

IT Support Challenges

IT organizations are tasked with balancing business expectations while keeping the trains running. Within the IT organization, IT support teams face their own challenges.

- **Pressure to improve:** Continuing pressure to improve relevance of services to the business.
- **Balancing change with stability:** IT must balance rate of change with stability to ensure business value is provided.
- **Blind spots in information:** Scattered data or lack of information at all levels limits incident and problem understanding.
- **Sparse communication on changes:** Deployment failures have a high risk of downtime. And too often, changes are made that IT support doesn't know about (lack of communication).
- **Redundancy and rework:** Duplication of work, redundancies, knowledge gaps and hand-offs—resulting in higher IT and business costs.
- **Lack of alignment with other teams:** Support feedback often represents customers, and doesn't always get communicated to and from development, product owners and product managers.

Collaboration is the key to business success.

IT support teams are responsible for customer interactions and their success depends on the stability and quality of services. It's often the case that development and support teams have different goals, different methodologies, and work in different management systems. Over time, they become silo'd when there is a lack of collaboration, creating problems through miscommunication and a lack of knowledge sharing.

How do we break down these silos, improve collaboration and drive higher business value? IT support and development teams must work together, and with all other IT functions: development, operations, QA, support and testing. Everyone.

The DevOps movement was a response to this conflict by development groups and aims to bring these teams together. IT support needs to understand and adopt these agile, lean and kanban principles so that they can continue to support the IT and business services.

Let's dive in.

What Is DevOps?

DevOps advocates a collaborative working relationship between Development and IT Operations, where historically they've been separated.

"DevOps shifts the tradition of how IT is organized, how engineers interact. It brings a set of best practices that guides how engineers and IT works that is markedly different than a traditional set of principles. It's a culture of automating, measuring, and sharing in the name of increased efficiencies throughout the software development life cycle."

—Jason Hand, DevOps Evangelist at VictorOps, organizer of DevOps Days.

Development: Teams developing and maintaining software products and services. We'll focus on internal developers maintaining the systems and tools employees use to be productive, such as centralized purchasing, CRMs and more.

Operations: Teams delivering, managing and supporting products and services. We'll focus on IT teams supporting internal systems that employees use.

Companies like Netflix, Amazon and Google apply DevOps practices to release faster and maintain a high quality of services. For example, Netflix does around 100 releases per day—and we're not talking movies.

When you cultivate a collaborative culture between development and support teams, you shorten the feedback loop necessary to drive business agility. The functions of a collaborative culture can be further explained by the following widely accepted DevOps principles—The Three Ways.

Chapter 4

DevOps Principles

Patrick Debois, creator of the term DevOps, says in his blog, "DevOps is not about a technology, DevOps is about a business problem."

The best way to understand DevOps is distilling it into 3 principals popularized by the Phoenix Project:

The First Way: Systems Thinking

"The First Way emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department. DevOps transcends departments and showcases the overall value to the customer."

– Gene Kim

Development builds, and then ships to Operations—fortifying customer value. The First Way is all about shifting our thinking to the entire system from the customer's perspective, versus the goals of one team or department.

Some examples of Systems Thinking are:

- Never passing a known defect to downstream work centers
- Never allowing local optimization to create global degradation
- Always seeking to increase flow
- Always seeking to achieve profound understanding of the system

The Second Way: Feedback Loops

"The goal of almost any process improvement initiative is to shorten and amplify feedback loops so necessary corrections can be continually made,"

- Gene Kim

By improving feedback between development, operations, support and customers, all teams understand their impact to the overall goal.

Clarity and collaboration between teams breaks barriers, improves productivity and reduces conflict. When teams work together from the same feedback loop, communication breakdowns are a thing of the past.

The Third Way: Continual Experimentation and Learning

"Creating a culture that fosters at two things: continual experimentation, which requires taking risks and learning from success and failure; and understanding that repetition and practice is the prerequisite to mastery,"

– Gene Kim

This might be allocating time for experiments or creating rituals that reward your team for risk-taking and success.
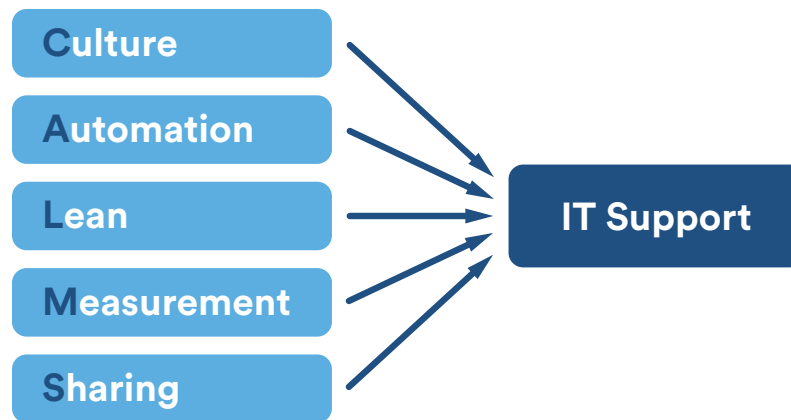
Source: http://itrevolution.com/the-three-ways-principles-underpinning-devops/

Keep CALM(S) and Apply DevOps to IT Support

So, how do we put DevOps into practice for IT support?

Luckily, there's a framework called CALMS, coined by Damon Edwards and John Willis at DevOpsDays Mountainview 2010.

Culture
Automation
Lean → **IT Support**
Measurement
Sharing

Let's apply the CALMS model to IT support.

Culture

Culture is all about daily or weekly rituals that influence how a team works.

Shift from silos to constant communication

DevOps is about agility and speed. Some rituals you can apply to your team include:

- Daily standups
- Hack days
- Chatrooms
- Scrum, sprint planning, and retrospectives
- Kanban framework

Focus on results, not just activities

Don't obsess on the number of changes, but the results of those changes.

Understand behaviors

Culture is a result of habits—you need to understand how your teams work together, and what you can improve.

Automation

Automation works when it supports the people and processes in place.

Go from trapped knowledge to shared knowledge

- Known errors should be tracked at the time of deployment and communicated to support.
- Support should link articles to incidents and problems, so everyone understands what's causing the issues.

Shift from orphaned tickets to linked tickets

- All teams should be responsible for ticket linking and documentation- including the service desk, admins, developers, and operations, so everyone knows what's going on.
- Link incidents, problems and requests to changes and releases.

Don't just deploy, but support

Continuous integration, continuous delivery, and continuous deployment are core to the First Way of DevOps—thinking about the performance of the overall system. Your team needs to be prepared to support, operate, and ensure their ability to restore services. They need to know what's being deployed and when, so they can give the right answers to the customer.

Help with testing

Automated testing is key to a DevOps environment. On top of that, support teams hold data about various customer environments that should be continuously fed back to testing.

Keep up with monitoring data

Support should know about any proactive monitoring and provide requirements for monitoring, with events linked back to incidents, problems, and changes.

Lean

Eliminate waste

Make your processes as simple as possible. What are you doing that doesn't add value? Eliminate duplicate work or unnecessary administration.

Always improve

Continuous improvement in IT support is about constantly iterating on your processes and services to deliver higher value services to the business.

Failure is normal

One principle of lean and of the Third Way is understanding that if you're not failing, you're not getting stuff done. "What was learned" should be the most important thing your team reflects on. Create a ritual of post-incident or post-mortem reports based on lessons learned that are shared across teams.

Metrics

Keep track of metrics that matter

How long does it take to recover from failure?

Measure 'Mean Time To Repair' (MTTR) and whether it increases or decreases over time.

How many repeated issues do you have?

Measure repetitive incidents and how often they occur in a period.

Are you meeting your targets?

Measure service level targets in your service level agreements and operational level agreements in the context of deployments. What's the impact of deployments on achieving your goals?

What are your costs?

What's your cost per incident, along with the total cost of support? Take this in context with downtime costs. In some cases, increasing support costs may be the best thing to do to minimize impact to the business.

Sharing

When it comes to sharing, ask yourself these questions:

Views

Does the organization understand what we're doing? Do we share that understanding?

Goals, priorities and pain

Do all teams understand each other's goals, hardships and priorities?

Process and workflow

Does your team's outputs of processes flow with other team's processes?

Knowledge

Is what your team knows shared with others? If it's in a system, is it accessible?

Communication

- How is your team collaborating with others?
- Are your managers rewriting communication?
- Are there unbarred channels for communicating?
- Are the right stakeholders kept in the loop of your projects, even if they're not in your immediate group?

Ownership

Do your IT groups—development, operations, testing, support and even the business—share ownership of successes, failures and learnings?

Toolsets

Do teams share tools and information? If not, do those tools integrate with each other?

Chapter 6

The Road to Success

Success is effective processes, not just *efficient* processes

- **Innovation:** Visions and goals are linked to business outcomes.
- **Sharing:** Categorizations and tools enable collaboration; ensure your tools work together.
- **Process:** Incidents, problems, and changes are part of a backlog. What we fix or don't fix is based on the organization's needs.
- **Improvement:** Continually improve service design, transition and operations.

Critical Success Factors

- **Common goals:** All teams understand one another's goals and agree to a set of the same goals.
- **Sense of urgency:** Everyone agrees to timelines and objectives.
- **Culture change:** A common vocabulary.
- **Learning:** People processes and automation.
- **Metrics:** Ensure that goals are set on outcomes.
- **Reinforce behaviors:** Reward and recognize teams.

Conclusion

Success Stories

Faster, better service is about culture, sharing and tools that enable your teams.

A collaborative culture is critical to success in DevOps. Aligning support and development can help your organization understand what services the business wants and needs to be productive.

Here are two companies that have brought their development and support teams together in JIRA Service Desk, powerfully simple service management software built on the JIRA platform.

Puppet Labs: Resolving tickets 67% faster
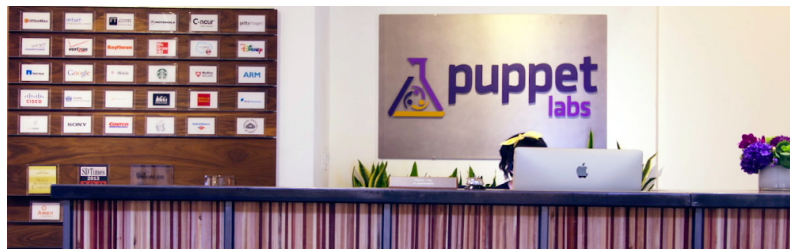
From small tech start-up to venture-backed company with more than 300 employees, Puppet Labs found itself at a crossroads. It needed to scale its support channels while maintaining service levels. The builder of IT automation software—on-premise and in the cloud—decided to turn to JIRA Service Desk for help.

"Since deploying JIRA Service Desk, we've had our resolved tickets increase by almost 67 percent, which is a sign that our help desk team is able to actually get more work done."

—Nick Cunningham, IT Manager

Today, all 300 employees are supported by JIRA Service Desk. Puppet Labs IT Manager Nick Cunningham notes,

"With JIRA Service Desk and the entire Atlassian stack, we've been able to increase user adoption by staying in the same system, streamlined our maintenance process, and reduced our total costs and labor for maintaining all those systems."

With Atlassian products, Puppet Labs was able to streamline its workflows, add critical efficiencies, and provide the help their customers needed quickly.

Skyscanner: The sky's the limit
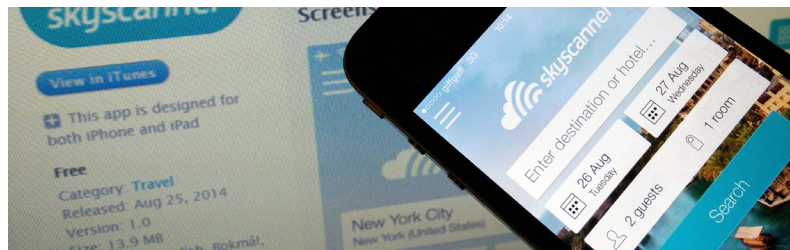
Millions of people book trips using Skyscanner's global search engine every day. With thousands of searches every second, it's no surprise that Skyscanner takes bugs in its software systems seriously.

To streamline the bug reports that came in, they needed a new type of service desk.

"We're building better software by fixing bugs twice as fast since implementing JIRA Service Desk."

—Michael Hall, Business Tools Lead

Today, their 700 employees in nine global offices are helping improve their global search engine with JIRA Service Desk.

"Our end-users actually get email updates, they know when problems are fixed and they understand their impact on product improvement."

With Atlassian, Skyscanner can now focus on individuals and interactions and are no longer limited by tools.

JIRA Service Desk brings service and development together.

Learn more about JIRA Service Desk