WIND™

AN INTEL COMPANY

DevOps in the Internet of Things

Six Reasons It Matters and How to Get There

WHEN IT MATTERS, IT RUNS ON WIND RIVER

EXCUTIVE SUMMARY

It's increasingly apparent that the development of software for the Internet of Things (IoT) and the management of those systems once they are in operation cannot be separated—making IoT software an area ripe for "DevOps." More than a buzzword, DevOps has the potential to help accelerate system development, ensure system quality, and optimize system reliability in the field.

DevOps is already being used in the IoT enterprise systems where the business logic resides. This paper sets forth six sound reasons why DevOps should, and likely will, become standard practice in the development and deployment of software for gateways and edge devices as well, and outlines a technology infrastructure that can help organizations implement IoT DevOps more quickly and easily.

TABLE OF CONTENTS

THE EVOLUTION OF DEVOPS

The integration of development and operations, dubbed "DevOps," is a hot topic in IT circles—so hot, in fact, that a standard definition, let alone formalized DevOps structures and best practices, is yet to emerge. One source characterizes DevOps as a "culture, movement, or practice" emphasizing collaboration between developers and IT operations teams with the goal of creating an "environment where building, testing, and releasing software can happen rapidly, frequently, and more reliably."

Note that this definition refers to "culture" rather than organization. While it's true that DevOps may ultimately require an organizational change, it first requires a cultural change to break down silos that separate those who build software and systems from those who implement and operate them. (The evolution of DevOps may be likened to that of agile development, which began as a movement with the "Agile Manifesto" and is still thought of more as a set of principles than a process.)

For developers of embedded systems, the concept of DevOps may at first seem foreign. Historically, the development team built the software (or a device) and handed it off to another team for release and support. But in IoT that model is a recipe for something far short of success, if not failure. The reliable performance of an IoT solution requires a constant feedback loop, regular monitoring, speedy issue resolution, and frequent upgrading. Internet connectivity creates the opportunity for constant infusion of innovation into the system, without waiting for the next "big bang" release. It's a process of continuous learning that necessarily requires developers and operators to collaborate closely every day.

Agile development is the forerunner to DevOps. In agile development, designers, testers, developers, and integrators merge into cross-functional teams that have end-to-end responsibility for specific functions or subsystems (see Figure 1). That responsibility includes delivery, which ideally can be automatic once the software passes internal testing and quality assurance. Automation of delivery makes possible the concept of continuous deployment, which increasingly goes hand in hand with agile methodologies.

If your organization practices agile development and continuous deployment, you are on the evolutionary path to DevOps. You may even be practicing DevOps in an ad hoc fashion without realizing it. The next step is to formalize a DevOps organizational process and structure, supported by technology that integrates the building, testing, deployment, and management of IoT applications on a single platform with a high level of automation. But before we discuss how you do it, let's be clear about why you should.

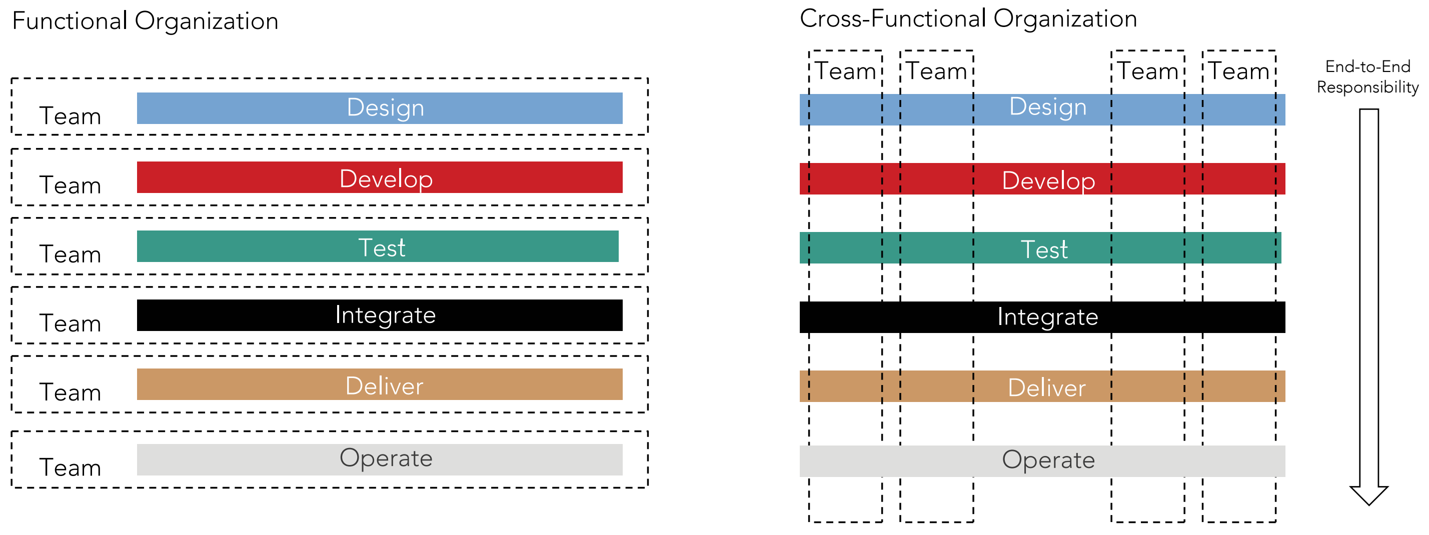

Figure 1. Functional vs. cross-functional organization

SIX REASONS DEVOPS MATTERS IN IOT

DevOps has already gained a foothold at the enterprise level. Witness the social media networks or video or music streaming services that are constantly pushing new content to users. Or consider the mobile device makers and application providers that make new releases of operating systems and other apps available for download on a regular basis. Now, the next frontier for DevOps is IoT, specifically the edge devices (the "things" in the Internet of Things) that perform the IoT system operations and feed data back to the enterprise. Here's why:

- The evolution effect: DevOps is simply bound to happen, in some shape of form, as the next step in the evolution from agile development to continuous deployment. Organizations that embrace DevOps and formalize it through integrated, cross-functional teams are likely to have a decided advantage over those that do not.
- The spreading effect: If software on enterprise servers is being updated regularly or continuously, the systems that are connected to those servers and dependent on that software will likely require frequent updating as well. At some point, teams that develop the enterprise software will expect faster release cycles in the other parts of connected systems.
- The infrastructure in place: With systems now connected via the Internet and the cloud, it's possible to automatically deploy and regularly upgrade software in multiple field devices remotely.
- Software-defined "anything": Increasingly, it is the software deployed on a device (regardless of the hardware) that differentiates it and defines its functionality. That means that when functionality needs to be updated, it is more often going to entail a software update rather than an electrical or mechanical modification.
- New business models and revenue streams: The big promise of IoT is that it makes possible new business models and sources of revenue that couldn't otherwise exist. The ability to constantly deliver new software updates makes it possible to sell services that generate continuous revenues, rather than simply the one-time sale of the underlying product.
- Greater productivity and cost-efficiency: A process that accelerates development cycles without compromising quality through more effective collaboration is simply a better, faster, smarter way to work—with the potential to drive down operating costs.

The case for evolving DevOps from enterprise systems to IoT edge devices is fairly compelling. IoT application and device developers are under enormous pressure to deliver quality solutions and meet tight time-to-market demands. And because IoT systems may be expected to perform for many years, development and operations teams must work together to plan for their entire lifecycle, from design through end-of-life. Companies that can meet these challenges stand to gain a significant competitive advantage—and a transition to a formalized DevOps organizational structure would seem to be the answer for adapting to this new environment.

So why isn't it happening more quickly? Let's look at some of the obstacles and challenges to DevOps implementation.

OVERCOMING THE OBSTACLES

Change is rarely easy, and instituting DevOps is no exception. Organizations are likely to encounter several obstacles in the transition.

- Cultural and organizational change is difficult: A transition to DevOps entails overcoming years of ingrained cultural perceptions and behavior. Changes in reporting relationships, responsibilities, and accountabilities are bound to be a bit rocky. An organization must recognize the need to change and then build the processes and systems necessary to accomplish the DevOps vision.
- DevOps is inherently difficult at the device level: Unlike the enterprise software environment, where server hardware is fairly standardized, IoT systems can be very large, unique, and complex, with a wide variety of hardware platforms. Where enterprise systems have built-in redundancy, there is typically very little redundancy for software embedded in field devices. Reducing the risk of costly failure requires exhaustive production-level testing and quality assurance, which lengthens development cycles.
- Reliability is paramount: When software is continuously

deployed, it has to work as promised. IoT solutions have very high demands for reliability, quality, security, and safety. Correcting problems in deployed software can be extremely cumbersome and inefficient, and the business risk is high if the manufacturer has a contractual obligation to ensure performance as expected. Quality assurance is an essential ingredient of any DevOps model.

There are few tools that actually support the DevOps paradigm, which calls for agile code sprints, automated testing, fast and automated feedback loops, and collaborative teams with a high degree of autonomy and communication. What's needed is a clear path between development platforms and field systems, so that DevOps teams can monitor system health, detect potential issues, and act on them before they become problems.

HOW TECHNOLOGY CAN ENABLE DEVOPS

DevOps in IoT is not inevitable. It requires commitment, collaboration, communication, and a willingness to change. It can, however, be made easier with technology that integrates system development, testing and debugging, deployment, monitoring, and management on a single platform. Unlike conventional development tools designed to support functional organizations working in horizontal layers, an integrated approach would enable true cross-functional collaboration in a vertical model (refer to Figure 1). It would allow system developers and those responsible for operationalizing the software to work as a coordinated team in a centralized, cloud-based environment, thereby accelerating the delivery of applications with full quality assurance and enabling effective troubleshooting of systems in the field.

Wind River® Helix™ Cloud is an example of such a solution. It's designed to help teams build and deploy IoT systems more quickly and manage them more effectively, leveraging the cloud for anytime, anywhere access and enhanced collaboration.

It's important to think of DevOps not simply as the merger of development and operations, but as the intersection of development, quality assurance (QA), and operations (see Figure 2)—QA being the essential step that ensures a system will work properly before going into operation. Helix Cloud maps to this three-stage DevOps model through the integration of three core components:

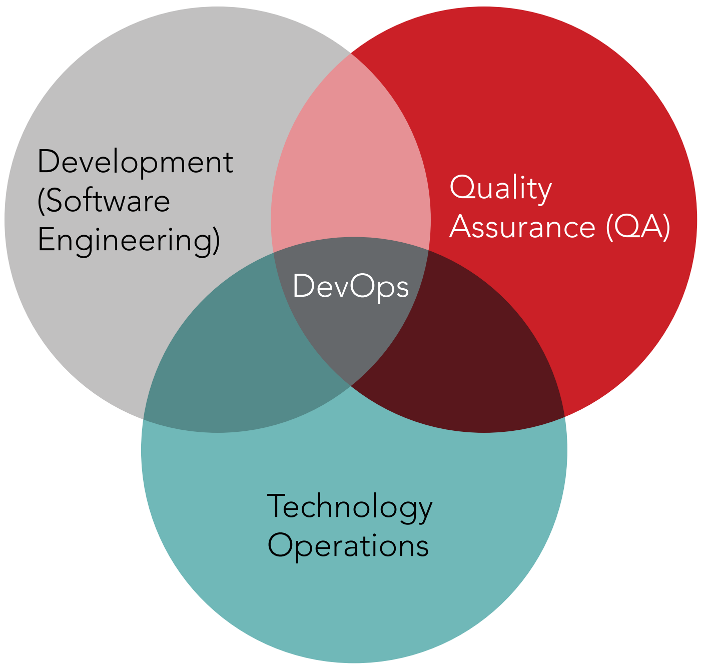

Figure 2. DevOps is the intersection of development, QA, and operations

- Wind River Helix App Cloud serves the development side of DevOps. It equips application developers with ready-to-use tools and software development kits (SDKs) for any hardware variant. App Cloud makes it possible to easily build applications independent of device operating system and hardware complexity. By providing developers with the appropriate tools and target systems, App Cloud helps mitigate integration issues between application and platform software and cut down on team handovers. As a cloud platform, it also allows anytime, anywhere access to tools and enables large, geographically dispersed development teams to collaborate across borders and time zones.
- Wind River Helix Lab Cloud addresses the testing and quality assurance aspect of DevOps. It allows instant access to virtual hardware of whole systems at representative scale, enabling teams to use full-system simulation for testing and QA of complex and large-scale IoT systems. A simple login provides any engineer with access to the cloud-based virtual lab. Using on-demand simulation software as a complement to hardware, teams can automate testing in entirely new ways and create

any number of virtual target systems for parallel testing. This significantly shortens the cycle between application development and system testing. With Lab Cloud, teams can stage and test systems at representative scale in a pre-production environment so they can move into production with confidence in system continuity.

- Wind River Helix Device Cloud is the platform for managing and operating devices from the cloud—the "bridge" between development and operations. It enables operators to safely and securely update, monitor, service, and manage devices in the field. Device Cloud automatically collects and integrates data from hundreds or thousands of disparate devices, machines, and systems, enabling operators to track device status and content, share data among engineers, and proactively determine when updates are needed.

Collectively, the Helix Cloud suite enables organizations to transition into a DevOps team structure. Specifically, it:

- Facilitates a new paradigm of "always connected" collaboration that isn't restricted by geography
- Helps break down the silos that separate those who develop software and systems from those who deploy and operate them, enabling teams to work cross-functionally in a collaborative spirit and making the release of software fast, reliable, and automated
- Allows testing of software at scale before deployment as well as proactive management of field devices, which together enable continuous quality assurance

CONCLUSION

Connected IoT edge devices have a lifecycle beyond "deploy, break and fix, and retire." Connectivity creates the opportunity to continuously infuse incremental innovation across the system lifecycle. DevOps puts organizations in the best position to capitalize on that opportunity.

As the lines between software creation and operation begin to disappear, so too must the organizational lines that separate system developers from system operators. The DevOps concept has been proven at the enterprise level and is evolving toward gateways and edge devices as a means to meet the unique challenges of developing and managing IoT systems. While there are obstacles to overcome, both cultural and practical, a technology platform that integrates system development, testing and QA, deployment, and management can provide the necessary infrastructure for implementing DevOps, empowering cross-functional teams to accelerate system development, ensure system quality, and optimize system reliability in the field.