Making Matrixed Organizations Successful with DevOps

Tactics for Transformation in a Less Than Optimal Organization

IT REVOLUTION
DevOps Enterprise Forum 2017

Making Matrixed Organizations Successful with DevOps

Tactics for Transformation in a Less Than Optimal Organization

IT REVOLUTION
DevOps Enterprise Forum 2017
Portland, Oregon

25 NW 23rd Pl
Suite 6314
Portland, OR 97210

Making Matrixed Organizations Successful with DevOps:
Tactics for Transformation in a Less Than Optimal Organization

First Edition
Produced in the United States of America
10 9 8 7 6 5 4 3 2 1

Cover design and interior by Devon Smith

For further information about IT Revolution, these and other publications,
special discounts for bulk book purchases, or for information on booking authors
for an event, please visit our website at www.ITRevolution.com.

PREFACE

In April of this year, we at IT Revolution had the pleasure of hosting technology leaders and experts from across the DevOps Enterprise community at the DevOps Enterprise Forum event in Portland, Oregon. The Forum's ongoing goal is to create written guidance to overcome the top obstacles facing the DevOps Enterprise community.

Each year at the Forum, the topics covered have included organizational culture and change management, architecture and technical practices, metrics, integrating and achieving information security and compliance objectives, creating business cases for automated testing, organizational design, and many more.

For the first two years, we organized the participants into large teams that worked on a small number of broad topics. However, this year, we shifted our approach in two ways—first, we invited a core group of past participants to propose topics they would like to work on and second, we asked them to narrow their topics so that they could have "nearly shippable" artifacts by the end of the second day. The result was more teams working on more problems with more written guidance.

After the Forum concluded, the groups spent the next eight weeks working together to complete and refine the work they started together. The results can be found in this year's collection of Forum papers. I hope you will agree that the smaller teams and reduced scope of the guidance benefits both the teams as well as the reader.

IT Revolution is proud to share the outcomes of the hard work, dedication, and collaboration of the amazing group of people from the 2017 DevOps Enterprise Forum. Our hope is that you will gain valuable insight into DevOps as a practice.

—Gene Kim
June 2017
Portland, Oregon

INTRODUCTION

At the 2016 DOES conference, the paper *Thinking Environments* was published by IT Revolution.[1] In *Thinking Environments*, the authors describe four organizational models, and the benefits and drawbacks of each model with respect to DevOps. They recommend moving away from a matrixed organization to overcome the drawbacks for DevOps teams. Figure 1 illustrates organizational optimizations for DevOps.

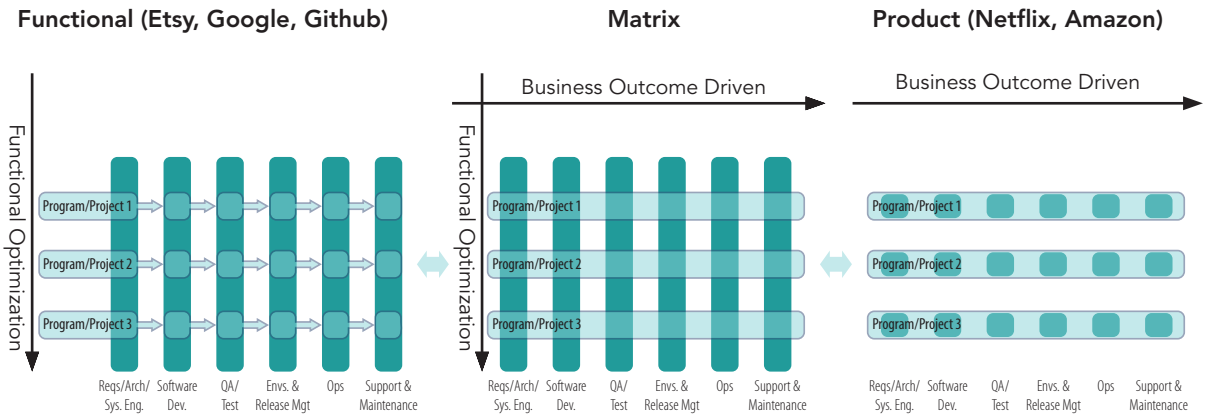

Figure 1: Organization Optimizations for DevOps

In some cases, especially large traditional organizations, there are good reasons for not moving away from a matrixed organization. For example:

- Skills development
- Career development
- Separation of duties
- Optimization of resources across products/programs
- Potentially more clear ownership of technical debt compared to the product focused model (no one owns technical debt in the product focused model)
- Aligns with business funding and/or regulations

In other cases, organizations are structured to reflect customer organizations or the organization is early in their DevOps transformation and there's insufficient support for reorganization.

[1] Download the full paper here: http://itrevolution.com/book/thinking-environments/.

This paper is focused on the problem of organizations that can't or won't move away from a matrixed organization. In particular, we focus on those matrixed organizations in which:

- The functional (vertical) organization is responsible for defining and measuring the process;
- And, the programs/projects (horizontal) are responsible for program execution.

Audience

This paper is written for team leads and specialists in matrix-based organizations where a significant part of the work is done in programs and projects, thus creating conflicts with task priorities and work visibility, as well as leading to misaligned metrics on the individual and team level.

Problem Statement

As defined in *Thinking Environments* and expanded by this team, there are drawbacks to a matrixed organization. Our goal is to provide tactics for mitigating those drawbacks where possible. The table below shows organizational attributes, the drawbacks of a matrixed organization with respect to desired DevOps attributes, and some tactics your matrixed organization can use to mitigate the drawbacks.

Table 1: Organization Attributes

	Drawbacks	Tactics to Address the Drawback
Systems Thinking	Locally optimized process. Functional silos focus on their processes and optimizing their stream of work versus focusing on the overall flow and collaborating to overcoming bottlenecks.	Form a cross-functional leadership team (Joint Chiefs) on programs/projects. Visibility of the value stream. Eliminate hand-offs or minimize their impacts.
Skill Development	Functionally, there is no funding to sharpen the saw across the functions. The structure encourages the functions to focus on processes and training that only help their silo instead of improving the system as a whole, e.g., the test organization doesn't invest in learning coding skills needed for test automation.	Cross-functional team updates. Create processes optimized for a program to have alignment for the value stream delivery.

Organization Attributes, cont.	Drawbacks	Tactics to Address the Drawback
Process	Process is owned by siloed functions, which can drive localized optimization versus global optimization. Programs don't have the authority or ability to change the process to optimize flow.	Cross-functional team updates. Form Joint Chiefs. Update metrics. Creates processes optimized for a program to have alignment for the value stream delivery. Visibility of the value stream.
Accountability	Individuals can receive conflicting direction from their 2+ bosses. This of course can be mitigated by designating one or the other supervisor as the "direct" reporting relationship, and the other as the "dotted line."	Updated metrics to reflect end-to-end value. Single prioritized backlog for each individual.
Flow and Queueing	Will likely require queues, hence delays and management overhead to prioritize.	Make value stream visible. Single prioritized backlog for each individual.
People	Depending on the degree of affinity with the product portfolio, technical professional development expectations might not be set or measured; or, on the contrary, expectations for development outside the functional area might not be met. People may experience a tension between the requirements of function and product.	Encourage more empathy of individuals across functional boundaries. Visibility of the value stream.

DEFINITIONS

Value Stream: The value stream is all of the processes organizations go through to deliver value. For software organizations, this is the total activity that organizations engage in to go from a business idea to delivering a solution that meets a business need. If organizations are going to improve this process, the first step is understanding their current process and making it visible so everyone has a common understanding of the biggest inefficiencies. This visualization (often called a value stream map) should include that flow of value through the organization and metrics that highlight the inefficiencies in the process. A simple example of the scope of a software delivery value stream map is represented in Figure 2.

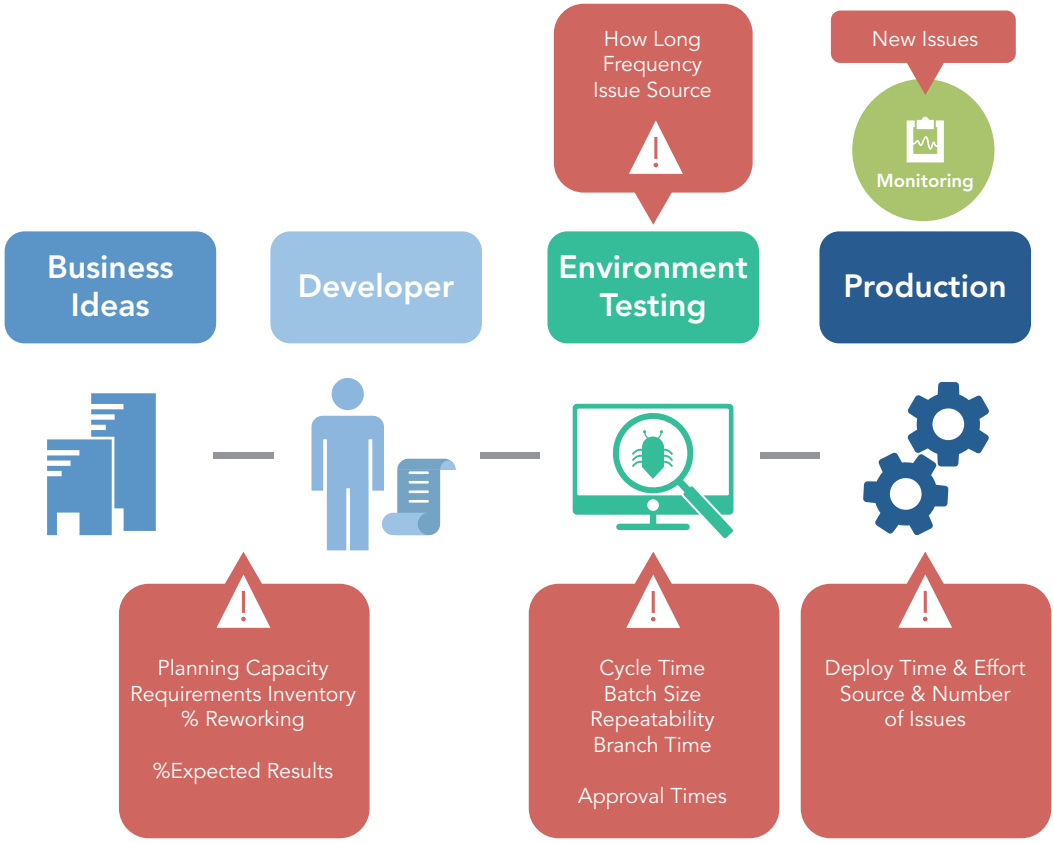

Figure 2: Software Delivery Value Stream Map

Delivery pipeline: For software organizations, the implementation of the value stream is typically accomplished with a delivery pipeline. This delivery pipeline is the specific process

and tooling an organization uses to go from a business idea to a developer writing code that they need to check in, test, and then deploy into production. Every organization has a value stream, but only organizations with processes under control have repeatable, standardized delivery pipelines. (If your process is ad-hoc, your delivery pipelines are always one-offs). People that are trying to improve the speed and quality of their value streams while maintaining all aspects of quality work toward standardizing and automating their delivery pipelines and putting all aspects of this automation under version control. The automation enables faster speed, supports smaller batch sizes, and ensures consistency by eliminating human errors. Putting all aspects of this under version control supports efficient triage by allowing everyone to know exactly who changed what and when. A simplified example of a deployment pipeline is represented in Figure 3, with different testing stages and quality gates.

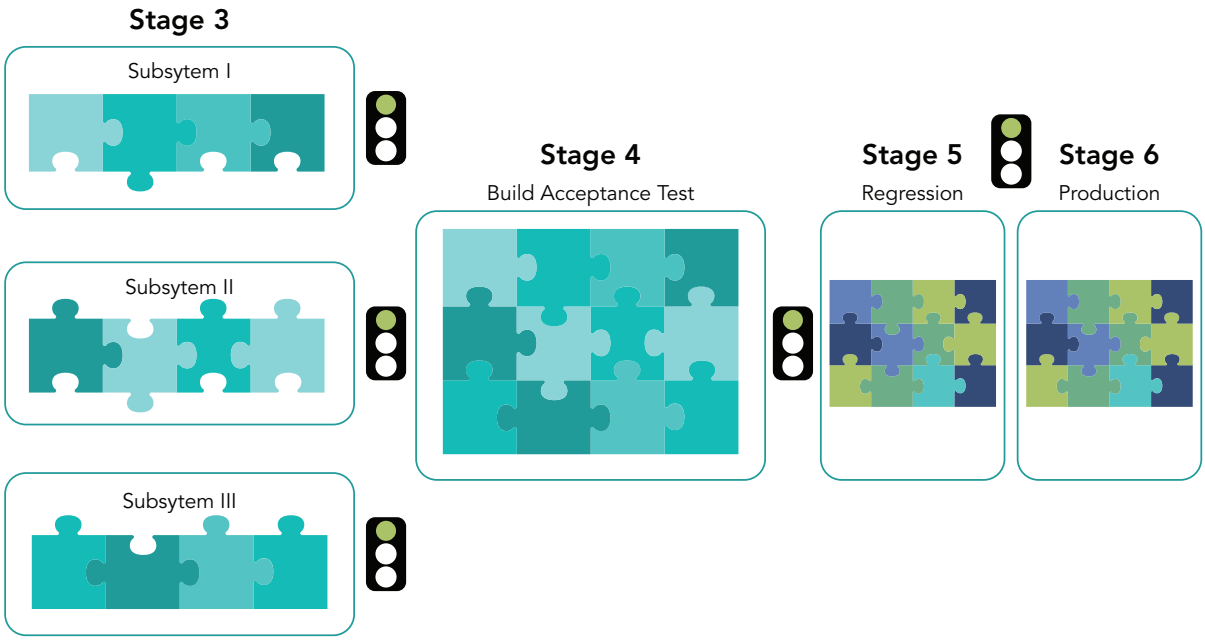

Figure 3: Full System Deployment Pipeline

TACTICS

1. Optimizing Processes for Flow Through the Organization

Cross-Functional Process Updates

- Creates processes optimized for a program to have an alignment for the value stream delivery.
- What it solves: Process is owned by siloed functions, which can drive localized optimization versus global optimization.

  Within a matrixed organization each functional structure defines and owns the processes for the functional area they lead. This creates processes that are often optimized for the functional area but not necessarily optimized for regular delivery of business value. To deliver value, the defined processes must be followed and are often auditable. By creating a cross-functional team comprised of leaders across the vertical structure, the organization shifts its focus from optimizing each individual functional area to focus on optimizing the whole and increasing the flow of value. This approach creates shared ownership and accountability for the delivery stream and reinforces desire for continuous improvements across the functional areas.
- What it is: Leads from each of the functional areas come together to create a cross-functional team (CFT) with a common goal of streamlining processes to improve lead and cycle time. This requires a systems thinking mentality in which the CFT looks across the entire value stream from concept or idea to delivery. In doing so, this team looks across the end-to-end flow of the processes they have created, and streamlines those processes to optimize value and flow. Therefore, the CFT will create common goals to optimize their processes for improved delivery. The functional leads are often responsible for ensuring the professional development of their members. Having a common goal across the functional areas will help shape the professional development opportunities that reinforce the shared vision. This in turn builds trust and common direction within a team that may have once had disparate and competing interests.

Joint Chiefs

- Form a cross-functional leadership team on programs/projects.
- What it solves: Lack of systems thinking; queuing; locally optimized process.

- What it is: Form a (scrum) team that owns a backlog with system-wide tasks, such as:
  - System architecture
  - Environments architecture (driving to common environments for all teams)
  - Defining capabilities/features, allocating features to teams, prioritizing feature development
  - Continuously monitoring process/value stream for optimization
  - Document the value stream, define the deployment pipeline, and prioritize the improvements
  - Hold monthly checkpoints to review progress of improving the delivery pipeline with the team so they see progress and understand the improvements that are coming next
  - Sample team members: technical functional managers, chief system architect, chief software architect, test architect, environments architect, chief engineer

Update Metrics to Reflect End-to-End Process

- What it solves: Locally optimized process. Organizations have metrics they use to measure the process, programs, functional areas, and individuals. Updating these metrics to support the goal of improving the value stream workflow improves the ability to transform to a DevOps culture while remaining in a matrixed organization. Current metrics, such as number of lines of code produced for a development team, directly hinder the overall goal of improving the value stream. Either transform this metric to include lines of code created per feature and for automated testing (for example) update the metric to include the required focus on the end-to-end value stream, or get rid of a lines-of-code metric altogether and bid based on feature size for design, code, test, and deployment. Add metrics for measuring work in progress (WIP) and rework, which are functionally agnostic. Another metric, defects found in QA/Test, needs to be changed to push the finding of defects earlier in the cycle and to reduce the defects found so late in the cycle.[2]

  Another key metric generally used for Operations is uptime. This metric directly impacts the ability to improve the value stream. This metric needs to be updated to include uptime with delivery of business function. Each of the metrics are updated to allow the functional areas to manage their teams and measure appropriately but also improve the throughput of the value stream.
- What it is: Each functional area needs to review the metrics they have in context to the overall value stream. This should be done by the cross-functional process to make

[2] For more recommendations on measuring process effectiveness, see *Measure Efficiency, Effectiveness, and Culture to Optimize DevOps Transformation* (https://itrevolution.com/book/measure-efficiency-effectiveness-culture-optimize-devops-transformations/).

sure the measurements for each area provide the value to the functional areas but also drive the right behaviors for the value stream for the DevOps transformation. The functional area leaders need to agree on these new metrics to ensure they support the overall business goals while supporting the improved flow for delivering business value. This will change the overall metrics for each of the functional teams as well as the metrics used to measure the teams and individuals within the functional areas. These new measurements will need to be agreed to by senior management as well to ensure people's reviews will be aligned with these new metrics.

Make the Value Stream Visible

- What it solves: Local optimization. It enables systems thinking because it spans functional silos and enables empathy. Without visibility there can be no understanding. It enables optimization of processes across silos and better joining up of the handoffs, thereby improving flow and reducing queuing. It enables people to start building skills outside their immediate silos, opening up the range of poly-skilled career paths in addition to those in the functional silo.
- What it is: Big organizations will often have several value streams. The most important point about visibility is that each person in the value stream and the people that manage them understand how their work contributes to the stream. They typically will focus on how to improve flow through their silo. Making the value stream visible will add focus on how they can improve flow for those downstream and how they can accelerate feedback to those upstream to improve the quality and executability of the work they receive. It's also important to make all work impacting the value stream visible. Initial mappings may just look at ideas to working software, but all four types of work impact the flow.

  One of the ways to maintain focus on the value stream is to hold regular collaboration workshops in which the leaders of each contributing vertical review the stream and agree and commit to improvements in their areas that will improve the overall stream.

Include all Functions in Planning at the Earliest Point

- What it solves: Late, inconsistent, or missing requirements which cause delays, rework, and quality issues throughout the value stream. This often occurs when teams outside of development are not included in planning early in the delivery lifecycle. Operations is a common victim of this, but it is not uncommon with other teams as well (security, QA, etc.). In a matrix-based organization where functional silos exist by definition, it is easier for teams to inadvertently—or intentionally—not reach outside of their own walls when planning.

Attention should be paid to the impact of the planning process on specialist function teams. These teams are often labor capacity constrained. Managers cringe at the prospect of having their already fully utilized team members sit in on long planning sessions on the chance that their specialty area is touched upon. This often results in this sort of exchange:

Ops: "We didn't know about this."

Dev: "We invited you to our planning meeting last June."

Ops: "There is no way we could spare someone to sit in on your five-day offsite."

- What it is: Intentionally involve all functional areas in planning at the earliest point possible. If all requirements (functional, operational, security, etc.) are treated as equals in the planning process, the absence of certain requirements or the lack of input from critical functional areas should be noticed. Also, organizations who plan work in shorter increments have an easier time gathering requirements and ensuring their completeness and accuracy (and are able to recover quicker if not).

  To help avoid aggravating the capacity constraints of the functional specialist teams, try planning in shorter increments and allowing the functional specialist teams to interact with the planning process in predefined, time-boxed increments (e.g., order a "four-hour planning consultation).

2. Implementing Changes That Improve the Effectiveness of Individuals in Matrixed-Based Organizations

Eliminate Hand-Offs or Minimize Their Impacts

- What it solves: Reduces delays and errors introduced by problematic hand-offs between teams.

  Hand-offs are when work (and the context needed to complete it) has to move from one group of people working as an integrated unit to another group of people. Hand-offs include moving work from one part of the the value stream to the next (e.g., from Development to QA/Test) or a request from a team to a specialist function (e.g., from Release Team to Firewall Team). These hand-offs are usually governed by request queues which are prone to bottlenecks, blocking the flow of value through the value stream. Additionally, the context switching that comes with hand-offs leads to (1) increased errors due to context being difficult to transfer accurately between humans and (2) decreased capacity of teams as context switching results in a higher cognitive load.

  Eliminating or reducing the impact of hand-offs is key to improving flow and, in turn, improving time to market and quality. The fewer hand-offs we can have, the better. This is why so many examples of DevOps success stories feature small,

integrated, cross-functional teams where most, if not all, of the necessary work to deliver and operate a service stays within the boundaries of a single team. In a matrix-based organization we are going to have functional teams, by design. This means there will be hand-offs. To avoid the flow and quality issues that come with hand-offs, there needs to be active strategies put into place.

- What it is: Eliminates as many hand-offs as possible.

  When defining your end-to-end delivery pipeline, look for opportunities to either merge teams or move work to where it can be completed by as few teams as possible. Be sure to avoid creating teams that are too large, which creates hand-offs and context breaks within a team. For example, in the DevOps community there are companies that have had success merging Development and QA teams so that Dev and Test cycles can be completed at a rapid, seamless effort without hand-offs or breaks in context. Going further, some organizations adopt a "if you build it, you run it" mind-set and create cross-functional teams comprised of Development, QA, and App Ops individuals.

  Additionally, this will help minimize the impact of remaining hand-offs. Deploy strategies to avoid the bottlenecks and quality issues that naturally come from hand-offs. Specifically, focus on:

  - For moving work from one part of the value stream to the next, focus on converting information flow (Word documents, tickets, spreadsheets, conversations, etc.) into artifact flow (moving artifacts between tools). This dramatically reduces the amount of context switching and human-to-human information transfer loss. An automated delivery pipeline is a good example of this process. Once the code is checked in, it automatically kicks off the process of moving the code through each stage in the pipeline with automated testing for feedback, thus avoiding delays.
  - For requests from a team to a specialist function, focus on replacing request queues with self-service interfaces. Treat the hand-off point as an internal service provider with standardized parts provided behind a self-service interface or API. This enables the specialist functions to provide their value without blocking or slowing the flow of the delivery team who needs their services.

Single Prioritized Backlog for Each Individual

- What it solves: Increased focus on systems thinking, clear accountability on direction, improved flow.
- What it is: In a matrix-based organization, people can get work assignments from both their line manager and their "dotted line" managers from various projects and

programs. These assignments often belong to separately managed (process-based) work streams, which creates a situation where an individual needs to follow multiple queues and switch between push-based and pull-based work items, as well as juggle conflicting priorities and deadlines. This doesn't just add to people's cognitive load, but also makes it difficult to provide visibility required by their managers.

Addressing work visibility can be one of the first steps to move toward a single prioritized backlog. Rather than working off different queues, an individual can, with the support of their line manager, create an individual backlog that is visible for all relevant stakeholders. Rules can be put in place to solve prioritization conflicts or a process can be agreed upon where stakeholders jointly set priorities. The line manager is responsible for solving situations where an individual gets overloaded with work.

One way to manage this single backlog is to use a kanban board, where assignments can be added as individual cards, with all relevant attributes (e.g., project code) added for improved tracking of work status across work streams. Extra attention needs to be given to using kanban for IT Operations work due to a significant amount of unplanned work (e.g., resolving incidents).

Encourage More Empathy of Individuals Across Functional Boundaries

- What it solves: Expectations for development outside the functional area might not be met.

  Having individuals become more knowledgeable across the entire value stream improves their ability to do their own focus area. This is not to say on large teams you want everyone to become generalists, but having, for example, developers understanding more about QA testing and the deployment and operations process helps them design and build the capabilities with a better design in the first place. By having the individuals in one functional area better understand the other functional areas they can better understand the challenges of the those individuals and can understand the impact of decisions they make.
- What it is: Individuals are encouraged to learn and expand their skills and knowledge across the full value stream. This makes individuals more productive within their own functional area while improving the flow through the value stream.

CONCLUSION

In large organizations it is not always possible to change the organization to support your DevOps initiatives. That does not mean you can't take advantage of the benefits that DevOps will provide. It just means you need to modify your approach and require the silos to take a more system wide perspective. This includes forming cross-functional teams that are responsible for leading the optimization of the flow of value through the organization and mapping the flow of value through the organization with metrics so it is clear to everyone what the biggest issues are that need to be addressed, and then start on a continuous improvement journey to address bottlenecks and issues.

Additional Reading

Atwell, Josh, Carmen DeArdo, Jeff Gallimore, Thomas A. Limoncelli. *Expanding Pockets of Greatness: Spreading DevOps Horizontally in Your Organization* (Portland, OR: IT Revolution Press, 2017).

DeGrandis, Dominica, and Kaimar Karu. *Using Kanban in IT Operations* (London, UK: Axelos, 2016).

Gruver, Gary. *Starting and Scaling DevOps in the Enterprise* (Philadelphia, PA: BookBaby 2016)

Gruver, Gary and Tommy Mouser. *Leading the Transformation: Applying Agile and DevOps Principles at Scale* (Portland, OR: IT Revolution Press, 2015).

Kim, Gene, Patrick Debois, John Willis, and Jez Humble. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* (Portland, OR: IT Revolution Press, 2016).

Contributors

Damon Edwards, Rundeck Inc., damon@rundeck.com, @damonedwards

Ben Grinnell, ben.Grinnell@NorthHighland.com, @ben_grinnell

Gary Gruver, Gruver Consulting LLC, gary@garygruver.com, @GruverGary

Suzette Johnson, Northrop Grumman, suzette.johnson5@gmail.com

Kaimar Karu, AXELOS, kaimar.karu@axelos.com, @kaimarkaru

Terri Potts, Raytheon Company, @terri_potts

Rosalind Radcliffe, IBM, rradclif@us.ibm.com, rosalindradcliffe@gmail.com, @RosalindRad