DevOps: Winners and Losers

An insightful guide to understand winning strategies for DevOps success

ConnectALL

ABOUT AUTHOR

Lance Knight is the SVP and GM of Operations for ConnectALL. His responsibilities include sales, sales operations, customer success, and technical support. Previously, he held SVP/VP roles at LeadingAgile, Tasktop Technologies, and Accept Software, specializing in field operations, sales development, and customer success. Lance started his IT career with a large aerospace manufacturer where he learned about Lean Manufacturing and Systems Thinking. He's a published author of books and white papers on leadership, software development, and software sales. @lancedknight

TABLE OF CONTENTS

Introduction

In today's highly dynamic and competitive business environment, winners and losers are differentiated by their ability to be more agile and efficient, and by the way they interact with customers and users. You either evolve or die — and building a vv is critical to staying ahead in the digital transformation game.

This book lays out the changing forces that are creating the urgency for greater efficiencies between development and operations. The book reviews some winning and losing DevOps strategies and shares insights on how and where automation and tooling fits into your DevOps roadmap.

At the end of this e-book, you will:

- Understand the changing forces driving DevOps
- Gain a high-level understanding of Lean Principles and Systems Thinking
- Learn how to defend the need for a budget around DevOps tools
- Acquire knowledge to create a DevOps roadmap

The importance of DevOps concepts

The landscape of software development is changing. Businesses today are in a constant state of change due to both external and internal forces, including rapid development and deployment of new technologies.

Time to market and reaction time to vulnerabilities are driving technology changes. Never before have companies needed to change rapidly and yet be more secure. There are more and more devices every day that need access to information. Reacting to risks rapidly is key to your company's ability to continue doing business. These capabilities are not product features. They are strategic business imperatives that are integral to a company's success.

These forces cause every company to be a software development company. Employment of software developers is projected to grow 24 percent from 2016 to 2026, much faster than the average for all other occupations.

The main reason for the growth in both applications developers and systems developers is a large increase in the demand for computer software.

Example: Banking industry.

The banking industry has had to adapt faster than ever before. Banking choices and customer loyalty are now technology driven. Falling behind is a going out of business strategy. My grandparents were with one bank their whole life and technology did not play a role in that choice. Personally, I recently switched banks for a better mobile app. At the same time, access to information needs to be secure and response time around vulnerabilities is imperative.

Businesses that rely on software to deliver new products and services must provide customers with the latest technology enhancements and an engaging user experience.

To meet this need, many companies are aggressively investigating and implementing modern software development and delivery methods such as lean, agile, continuous integration, and continuous delivery. When planned and executed together, these concepts create the foundation of DevOps that supports faster, more frequent releases of higher quality software applications, by focusing on automation and cross-functional collaboration. Enterprises that embrace DevOps can increase market share and deliver innovative, or even market-disrupting, solutions, making DevOps a leading initiative for software development and delivery.

State of DevOps today

Digital transformation is driving agile transformations and the adoption of DevOps philosophies. Time to market and reaction time are key elements in business success. It has created a tooling revolution and an explosion of DevOps tools in the software development and delivery world.

For many IT professionals, the evolving and growing number of tools, technologies, and practices which are associated with DevOps can be overwhelming. It can cause confusion and paralyze analysis, preventing effective adoption of DevOps, and resulting in costly failures due to misled initiatives. To overcome this, and prevent being losers, companies need a simplified approach and a strategy that helps them understand where they are, where they want to go and how they can best get there.

For DevOps to be successful, companies must solve the hand off issues between teams, and import the tools that users need to deploy with. The need is to break down the traditional barriers in the software delivery process caused by development and operations teams that have different objectives. Development wants to implement change as fast as possible, and operations wants to avoid any kind of change to keep things stable.

The goal of production software is to serve the product and security needs of the customer. DevOps is a collaboration between development, operations, and other teams with the recognition that everyone is tasked with achieving common business goals.

More than 80% of our customers worldwide have reported that they have adopted some principles and practices of DevOps. We've also noted that some initial efforts have not been successful mainly due to cultural and behavioral difficulties, hence they have to reset their DevOps initiatives. In essence, the adoption of DevOps has become more wide-spread, yet every implementation is unique. We believe that larger enterprises will continue to embrace DevOps by implementing it at scale, company-wide.

Losers: What defines failing strategies

Most of the 80% of companies that implement DevOps have not been successful in their initiatives due to cultural and behavioral differences or focusing on deploying specific tools rather than creating and executing a plan to improve the bigger system.

So, DevOps losers need to get started again. DevOps initiatives fail when teams are not aligned. DevOps is more than just pushing code to production. It involves cross-silo cooperation and is not easy. In a large organization, everyone gets excited about DevOps transformation, but each person has a different understanding of what that means, so, with everybody pulling in a different direction, you end up with something that doesn't enable you to release code more frequently.

Communication is key to getting everyone on the same page early on. As Gary Gruver, an experienced software executive, stated in his book, Starting and Scaling DevOps in the Enterprise, starting the DevOps journey is like having five blind men build an elephant. Solution: He suggests that organizations use process maps to communicate with all their teams to give everyone a common understanding of what the issues are and introduce everybody at the same time to DevOps concepts from both front and back end.

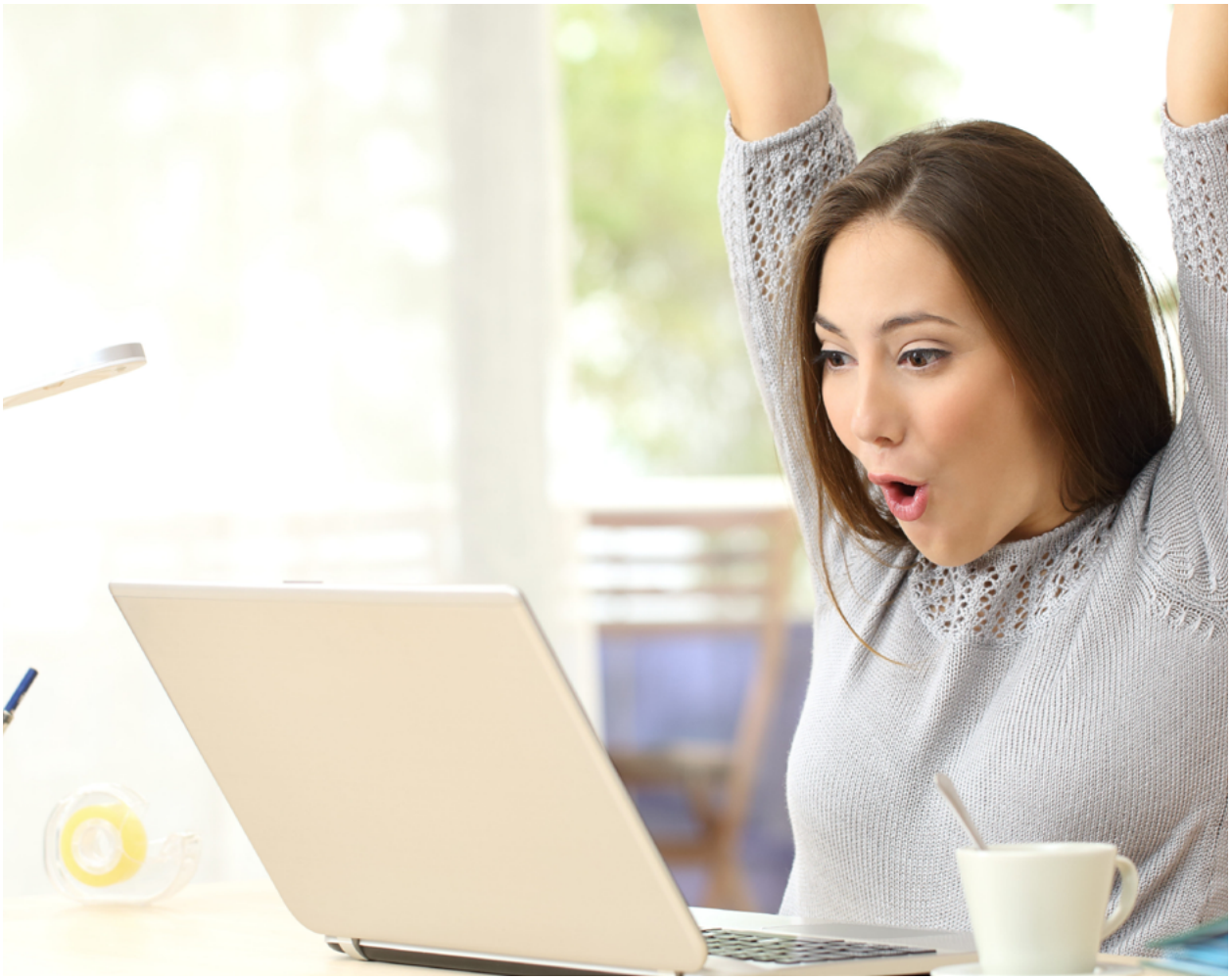
Winners: What DevOps winners do right

DevOps is not about tools only! It involves a set of practices that aim to apply the lean concepts of the manufacturing world to the software world. As Gene Kim, Founder of IT Revolution stated long ago — the heart of DevOps lies in these three core principles:

- Flow, which accelerates the delivery of work from development to operations and then to customers
- Feedback, which enables you to create ever safer systems of work
- Continual learning and experimentation, which fosters a high-trust culture and a scientific approach to organizational improvement on a daily basis

DevOps winners follow these principles, making small changes that provide quick feedback repeatedly. They have a common goal of decentralizing authority and control, tearing down infrastructure nightly, working as a single team, using metrics to improve and integrate systems, to remove waste, align on business goals, map their value streams, deploy code at any time, treat infrastructure as code, use lean and systems thinking, and make sure ops and dev sit with each other at every step.

DevOps winners break this down into three key areas: Culture, Infrastructure, and Process. Let's take a look at each of these in depth and the concepts that make them work flawlessly when implemented effectively.



Culture

DevOps winners change culture as well as process

DevOps winners focus on changing the differences in the cultures of the organization, breaking down the silos — operations want stable environments, while development wants to respond to customer requirements quickly. The most important part about the DevOps culture is understanding that it is not just about tools. It is a combination of people, culture, processes, and automation. Culture is the most important of these and continues to be a huge issue. The difference in objectives and mind set between development and operations demands that organizations break down the silos and bring everybody together from the top down, starting at the CTO or CIO. You can't force developers to do operations, but you can form one team with the same goals and objectives. This calls for not only a culture change but also a process change — from agile velocity for development cycles to automation to initiate continuous delivery, reducing risk, securing code, and bringing in more confidence in the entire pipeline.

DevOps winners don't depend on one hero

To make a DevOps transformation successful, enterprises need to get everyone on the same page — from individual contributors to the top management. It calls for a unified team that does not depend on one hero, therefore removing roadblocks and paving a path for easy flow of information. This begins with making sure that there is motivation, incentive, and most importantly the right mindset and culture for cross-team collaboration. We live in a service economy and everything today is about delivering top-quality applications to customers, emphasizing the crucial need to work together, which demands continuous delivery. Organizations should also help teams visualize an end-to-end workflow that gives them a breakdown of what they need to work on next, how much work is in progress, where the bottlenecks are, and more. This acts as a catalyst for a relationship-building environment as everyone knows who is working on what part of a service at a given time and how much each person is contributing to the overall success.

While smaller organizations or start-ups may find it a lot more organic to create an environment of experimentation and learning, it can be adopted by a larger organization with some careful steps. Gene Kim, in the books DevOps Handbook and The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win explains how one of the key principles of DevOps is fostering continual experimentation, taking risks and learning from them, and understanding that repetition and practice are prerequisite to mastery. This enables you to improve and also tread onto paths that are untraversed, giving team members opportunities to share their learning with stakeholders, both upstream and downstream, in a process. There is no checklist to what tools you should acquire or where you should begin — you make a plan as to where you want to go, experiment and learn on the go. In most DevOps implementations, things fall in place as you move ahead with resilience and additivity.

DevOps winners work as a single team

People are the main ingredient in a successful DevOps initiative. Having the right staff on the DevOps team will make or break the initiative — ideal scenario would be to embed operations in the development team. It is critical to designate people who are willing and able, and dedicated from the start to the DevOps initiative: willing, in that they understand the initiative requires potentially a completely new way of working, and able, in that they are capable of doing work in a new way — one that is different from the way they have been doing their current work.

Building resilient systems allows teams to collaborate better and work together, knowing how to respond to failure. When an enterprise begins DevOps, it should scrap non-scalable and traditional change management approaches to tackling failure. DevOps is a response to increasing process complexities and the best way to handle complexities is to build resilience. Instead of trying to cut out failure, enterprises have to learn to become resilient to failure. This should begin with the top leadership and trickle down.

While DevOps and agile aren't the same thing, agile development definitely plays an important role in DevOps deployment. So what kind of agile practices do you put in place? Kanban is something that organizations lean toward when it comes to the operations side of the DevOps equation. Kanban is a scheduling system from lean manufacturing which can be applied to all kinds of teams and processes. Lean is a set of principles that comes from the world of manufacturing. It encourages teams to focus on improving flow in the system, and thereby improving the continuous delivery of work, facilitating incremental product releases with chunks of new functionality or defect fixes, and enabling the visualization of the entire value stream. Reviewing the entire operations might help your organization chose what's best — there is a possibility that you may move the entire DevOps process to a Kanban basis, or development will continue with scrum while operations moves to Kanban.

DevOps winners have a common goal

With DevOps, you can't expect dozens of releases every single day. It takes time. The important step is to make sure everyone in the organization agrees on the overall objective and scope — accelerate delivery and manage risk. If you are a large organization, your focus will be on discipline, checks, and balances, as you would need to consider the legacy technology in comparison to smaller companies that need not worry about all the nitty-gritty. Whether large or small, all organizations need to work toward producing a high-quality software product that can be delivered to the end user.

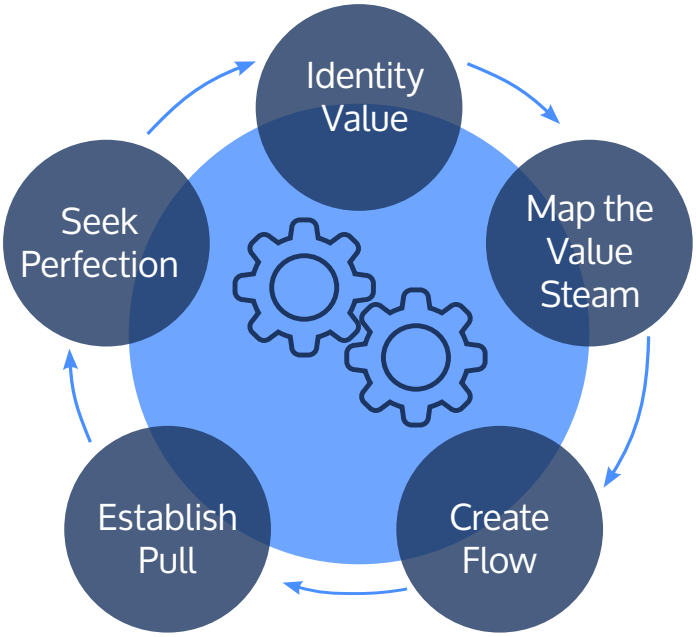
Process

DevOps winners use lean principles and systems thinking

Having a clear understanding of lean principles and systems thinking are essential to supporting any DevOps initiative. These principles allow you to see where waste and dependencies are impeding your velocity. Using techniques like value stream mapping and systems theory, you'll be able to remove waste in your software delivery process.

Systems Thinking

Systems Thinking is the application of Systems Theory to the DevOps process. This involves mapping the complete process from end-to-end, defining each component in the system and understanding the relationships between those components. Through this process, Systems Thinking allows you to find and eliminate bottlenecks and constraints in the process to improve overall process cycle time. Systems Thinking also uncovers opportunities to make the system self-regulating by applying measurement/improvement feedback loops to further reduce the overall process cycle time. Unless you look at the whole system as one entity, these opportunities for improvement are invisible.



Lean Principles

Lean principles focus on how to create value for the customer through systems by creating flow and pull processes, moving quality closer to the source, leading with humility, and respecting every individual.

One of the fundamental concepts in lean is the value stream. Value stream mapping has proven to be a good tool to not only get everyone together and understand problems, but also to identify areas of improvement, which is the main aim of DevOps. It allows you to assemble everyone involved with a workflow into the same room at the same time, to clarify their roles in the product delivery process and identify bottlenecks, friction points and handoff concerns. Value stream mapping reveals steps in development, test, release and operations support that waste time or are needlessly complicated.

It starts with the product person or team and moves through the development lifecycle, QA testing, release and operations, ultimately looking at how the IT team manages and monitors a product/feature after release. And it doesn't end there — it goes all the way through deployment till the customer sees value.

DevOps winners test before code

One of the concepts in DevOps is moving quality and test closer to the source in software development. This is accomplished by embedding quality and ops personnel in the development team and following protocols like TDD which involves envisioning an exhaustive test of the function, or actually writing test code before writing the functional software. This significantly reduces the number of iterations in the code/test process and the number of defects in the delivered code. It includes test in the delivered code package reducing the downstream test requirements and thus facilitating continuous integration and continuous delivery. Most importantly, this approach reduces the overall cycle time between definition and delivery of high quality, defect-free software.

DevOps winners use metrics to improve

You can't manage what you can't measure. It is essential to keep track of the progress that DevOps has brought to your delivery processes and tools, so you can make informed decisions and tweaks for more optimal results. It reduces level of risk, enables you to deliver changes frequently to the market, enhances customer feedback, and the reaction time to market changes exposing you to new security risks early on.

Measurement might always look a bit different depending on what your goals are. Since various members of an IT organization, and even members within a scrum team, have different priorities, it is indeed quite tricky to design a useful CI/CD dashboard that can make everyone happy. Lean metrics like cycle time and wait time is a good place to start but most solid metrics are always the ones the organization defines for itself based on the objectives of its unique business model.

Infrastructure

DevOps winners can deploy code at any time

Regardless of your end-goal, to achieve continuous delivery or continuous deployment, you need to automate your deployments and application release pipeline. This ensures the deployment of code at any time. Deploy Automation and Application Release Automation (ARA) are critical prerequisites for any continuous effort, and tools that enable these prevent deployment failures. Deploy Automation involves deployment modeling, which controls variability and reduces errors; enables accelerated and frequent product releases; enables consolidated access to tools, processes, data and people, with more visibility and improved collaboration between teams.

DevOps winners treat infrastructure as code

DevOps winners treat Infrastructure as Code (IaC). Sometimes referred to as programmable infrastructure, IaC is a means of automatically managing and provisioning infrastructure through writing code, avoiding any manual process. It involves describing servers in source files that you check into version control and apply automatically. Configuration management tools help you achieve this and lets you declare, in code, what your server should look like, and automatically apply changes to your servers.

Here are some key advantages of treating infrastructure as code:

- It leads to more reliable releases; automating the process of installing and configuring software reduces room for error
- It allows for a more repeatable release process
- Living documentation improves audits
- Environments are recreated, planned, tested just like code
- Avoids configuration drift
- Delivers technology roadmaps for the infrastructure

DevOps winners integrate systems to remove waste

In DevOps, integrating systems and enabling continuous delivery leads to easy incorporation of continuous feedback, improving the speed of product delivery, and enhancing collaboration. Automation and integration aids in removing some major wastes, adding value to the product, lowering costs and reducing lead time, increasing collaboration, resource utilization, unnecessary defects, waiting time reduction and talent utilization. DevOps winners amplify feedback to the extent that these wastes are identified early on and eliminated.

DevOps winners automate their pipeline

Automation is the backbone of successful continuous software delivery. While CI/CD is the automation of a single activity in the assembly line, DevOps is the pipeline of several pipelines. DevOps winners understand this difference and orchestrate the entire process as one large interconnected assembly line, and also automate DevOps tools. Here are some core components of good DevOps pipeline automation tools:

Continuous coding: Involves applying a code pattern that enables continuous delivery. The main intent of continuous coding is to find problems as early as possible and training your team to avoid them — reviewing and testing each change prior to integration or before it goes into a centralized version.

Automated testing: Using competent testing tools to fix the testing parts of an entire pipeline orchestration to make sure that teams are able to objectively evolve products without pause or breaks at every step of the delivery.

Consistent integration: Facilitating constant input integration, patch fixes, and subsequent deliveries, without compromising time to market.

Constant monitoring: Enabling steady communication all along the way among the teams to manage any exceptions.

Conclusion

Bringing DevOps into an organization is not an easy task and is definitely not something that can turn things around overnight or even in a few days. The world is full of examples of DevOps winners who have focused on consistency and not perfection, working patiently to achieve something that really matters and lasts for a longer period. Any new process creates risk and people are scared of change, but when you start small, see results and move forward in incremental steps, there is no fear of massive change. This fosters trust and generates success and therefore a recognition that implementing DevOps across an enterprise will benefit everyone.

For a successful DevOps transformation, an enterprise should focus on aligning the culture between teams. The operations team needs to adopt customer success principles and let go of their command and control heritage. They should stop acting like process enforcers and become process enablers. At the same time companies need to build automation in the pipeline, automating different levels of testing and code analysis, automating infrastructure requests, ensuring production build has completed the entire testing harness and automating the integration of defects back into one backlog.

One major takeaway from DevOps winners is that they recognize and address vulnerabilities with speed through the use of high-performance teams. To do this, DevOps must be supported from the top of the organization.

The outcomes:

200x faster deployment

60% fewer failures

Deploy **30x** more frequently

Experience **60%** fewer service disruptions due to change failures

Recover **168x** faster when they do experience failure

About ConnectALL

ConnectALL® powers businesses in achieving higher agility and increased velocity. Teams from software development and delivery, IT and business units across large and small enterprises worldwide use ConnectALL's integration platform to unify people, processes, applications and tools from multiple ALM and DevOps providers, such as Atlassian, Micro Focus, Microsoft, IBM, Salesforce, BMC, ServiceNow, and more. Designed to break down barriers to continuous delivery, ConnectALL helps companies rapidly create business value by bringing software innovation to market faster and increasing productivity through cross-team collaboration.

For more information, visit

www.connectall.com

+1 800 913 7457

sales@connectall.com