The DevOps ‘Secret Sauce’:  
9 Experts Share Their Insights on How to Succeed with DevOps
Table of Contents

1 Introduction

2 It's All About the People
   2 - Julia Wester
   3 - Helen Beal
   4 - Jayne Groll

5 It Takes the Right Processes
   5 - Marc Hornbeek
   6 - Aruna Ravichandran
   7 - Rob Hirschfeld

9 It Requires a Cultural Transformation
   9 - Steve Brown
  10 - Michael Fulton
  11 - Rob England

12 Summary
What makes some DevOps implementation succeed while others fail?

That question is always on the minds of IT managers who are just starting their DevOps journey. And for good reason.

When DevOps is done right, it results in benefits like faster deployment, higher-quality code, reduced time spent maintaining applications, faster time-to-market, and even revenue increases. But simply deploying DevOps tools or paying lip service to DevOps principles is not enough to yield meaningful change.

In fact, according to an Atlassian survey on DevOps Maturity, 35% of the companies utilizing DevOps approaches didn’t see the benefits they had hoped to achieve with the approach. While the good news is that 65% of organizations were realizing those benefits, new adopters want to know how they can avoid being part of the other 35%.

To find out, we contacted some of the most knowledgeable DevOps experts in the field. We asked them all the following question:

“Think of the best DevOps adoption project you’ve ever seen or heard of. What, in your opinion, was the “secret sauce” that made it successful?”

Their answers yielded a wealth of insight that we’ve collected in the following eBook. And while their responses diverged as widely as their personalities, a few common themes emerged.
The best DevOps adoption project I have been a part of definitely included a large technology component: building a robust CI and deployment process as well as an easy way to provision servers/environments. But, the adoption project also included the people side of the equation. We began building relationships between Ops and Dev Engineers. We had representatives of each group attend stand-ups for the other. We began fostering alignment through an understanding of the priorities of the other team which allowed us to anticipate what was needed from each other BEFORE it was needed. Through these interactions and deepening relationships, people began to understand the challenges of the other. That, in turn, generated empathy. That empathy helped us to make better decisions that considered the needs of the whole.

“Technology is definitely a necessary part of a DevOps transformation, but when you exclusively focus on technology you don’t really transform your organization... only its processes.”

Technology is definitely a necessary part of a DevOps transformation, but when you exclusively focus on technology you don’t really transform your organization... only its processes.
It’s solid leadership - the kind where the vision is understood and patiently communicated, where the leader has the humility to get into the trenches with the people to help change the behaviours, they understand the pace of change and keep enthusiasm and momentum for the stages of evolution even in the face of setbacks. It’s when they are quick to celebrate success and show gratitude for improvements and effort.

It’s solid leadership.
Culture doesn’t transform, people transform when inspired to do so. We know how to implement technology, but instilling a different way of thinking in people can be daunting. Techniques such as transformational leadership, immersive learning environments, internal DevOps Days, shared training experiences and value stream mapping exercises can help people connect to each other and to adapt to a new way of working. Creating a common vocabulary across principles and practices improves communication and collaboration. Cross-skilling and upskilling give professionals a clear learning path that can lead to innovative disruption. Helping people cross the cultural divide is likely the single most critical success factor for a DevOps transformation.
Successful DevOps projects always start with a thoughtful DevOps assessment. The assessment carefully picks the best starting application, delivers a clear understanding of the current state of flow against the 27 critical success factors for continuous delivery. The assessment includes a value mapping of continuous flow future state. With help of an experienced consultant, the assessment determines a program of improvements to people, process, and technology to accomplish the first level of continuous flow.
Two Things:

1) Identifying what their end goal was (metrics that matter) and working on meeting those goals. One of my blog articles, Metrics Only Matter When They Matter, goes into further detail.

2) Having a single customer champion (cultural transformation).

"Identifying what was their end goal was (metrics that matter) and working on meeting those goals."
The best DevOps victories are born from the cauldron of hard knocks. For me, those struggles have been during the growth phase of open source projects. We started our first generation of the physical provisioner Digital Rebar (at the time called Crowbar) as a single vendor project adjacent to OpenStack.

The real challenge came when we added external development partners to the project. Our lack of build automation went from an inconvenience to a series of days-long outages.

Our new development partner was six hours ahead of us and could make significant contributions to the code base before we'd had our first coffee. While this seems like a big win for velocity, the reality is it made our team grind to a complete halt. The project had no automated functional or integration tests so changes by the European team would often create issues that were exposed when we cranked our builds.

The result was chaos: our original team was constantly blocked by issues introduced in places where we were not working but still needed to resolve before we could get back to work. Often we'd fix issues just in time for the process to reset.
Today, we build the automated test infrastructure first in an open source project and work to maintain deep code coverage. The automated tests ensure that any change does not break behaviors or APIs before a person spends any time looking at the code. It also ensures that we minimize unexpected integration problems.

Most importantly, these gates provide fast accountability. Committers who introduce errors can self-correct before other team members are involved. It also creates incentives for expanding test coverage.

Lately, we've also been working to set up deeper integration tests that automatically build and exercise the code in production like environments.

The best DevOps victories are born from the cauldron of hard knocks.

These investments may seem like an excessive focus on testing; however, they have proven to significantly improve developer velocity for us. Our experienced developers spend much less time tracking down errors and regressions introduced accidentally. It also means that new contributors are more confident in their work as they come up to speed.
I think it's a combination of how a lot of elements work in concert together to help an organization transform from where they are today to where they need to go.

One is that the entire team, executives, IT, operations, and engineering at a minimum, are on the same page, that they collaborate. They understand where the organization is today where they want to go, and they're trying to find a way to do it together. Now we don't see that very often.

The other element is making sure that they understand the details of what their challenges are and that there's a little bit of science applied to the arts, and that there's a good plan drawn up and a program. And the third is the execution of that plan to make sure that the program that's designed to help them transform becomes an actionable executable process that is measured for success and re-measured and evaluated.

In addition, the culture and then the structuring of the transformation is absolutely critical to success.

I think you look at technologies and products after you look at structure and collaboration and process.

There's never a true reaching of success — it's always continued success. I think those are the key elements of it at any project.
The most important factor I have seen is the willingness to change and openness to learn.

“Willingness to change and openness to learn.”
There should never be a DevOps adoption project. DevOps is a way of working and a cultural transformation. A project has a start and an end, with deliverables, and the work is done internally to the project. The word adoption is also problematic because it has a finite implication.

Transforming the organisation has to be an agile iterative process without end. Therefore doing it as a project is a strong anti-pattern.

Transformation is an ongoing journey. The owner of the transformation needs to be a permanent function of the organisation to promote and advance it, which would normally end up sitting with a key takeholder for value delivery.

At my favourite client, they have established a Value Office, which owns the improvement of culture and practices to advance the delivery of value. They have established a "People Train" working alongside software Agile Release Trains.
The work must be conducted by everybody. Again it is an anti-pattern to have a project team trying to push transformation. It has to be pulled by everybody at their pace. We have to stop doing transformations to people. This was the failing of so many attempts at ITIL adoption.

There is no end state. Conventional organisational change managers get driven mad by the idea that we cannot define the target operating model. But anybody who claims to know what a group of people will work like in two years’ time is making it up. Every organisation is a complex system: we don't know what it will be like.

Conclusion

DevOps advocates often speak about the need for bringing the right people, process, and technology together in order to affect cultural transformation in an organization.

Based on the answers we got from experts in the field, technology seems to be the easiest of the three to get right. Several DevOps leaders noted that while technology is very important, it's really the other issues that separate DevOps failures from successes.

The “secret sauce” that makes some organizations succeed with their DevOps adoption seems to be a combination of people, process, and cultural transformation factors.

And as Rob England noted, “DevOps is an endless improvement journey, not a project.” Even if your initial adoption isn't as successful as you would like, you have endless opportunities to make incremental changes that will get you closer to your goal.