

EMBRACE DEVOPS

YOUR GUIDE TO THE DEVOPS LIFESTYLE

DevOps is both a cultural and a professional movement. Learn how Lean principles, a diverse workforce, and a good roadmap all help to build better products with greater speed.



TABLE OF CONTENTS

Foundations of DevOps	PAGE 3
DevOps and the Lean enterprise	PAGE 5
Forging connections among people with diverse skills	PAGE 6
Establishing roadmaps and themes	PAGE 7
Additional resources	PAGE 9

FOUNDATIONS OF DEVOPS

Debates about what DevOps is have been going on for several years. However, rather than being a static entity, with a single definition, DevOps may be closer to a practice, with some underlying principles that remain constant, and with forms and applications that vary according to the experiences of the practitioners.

A DEVOPS PRACTICE IS EASY TO RECOGNIZE

An example of such a practice is kungfu. The name kungfu does not denote fighting. Instead, kung can be translated as work or achievement or merit. The word fu means man, but can be taken in the sense of human. Translated, kungfu means excellence achieved through long practice of one's skills.

Even though there are many schools of kungfu, someone who practices it is easy to recognize. The same is true of DevOps. Although there are many schools of DevOps, there are some underlying principles that comprise the foundation for all of them.

DEVOPS IS BOTH A CULTURAL AND A PROFESSIONAL MOVEMENT

A definition that describes DevOps as a whole is that it is a

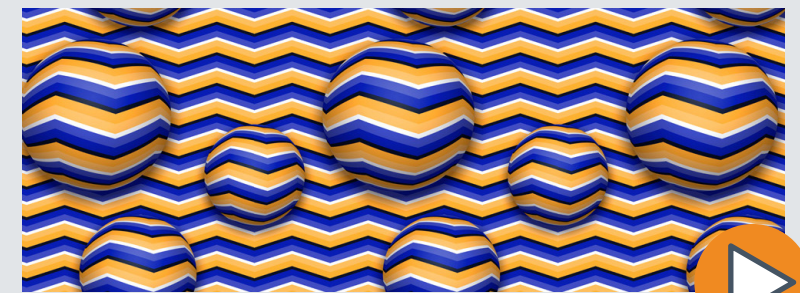
cultural and professional movement, focused on how we build and operate high velocity organizations, born from the experience of its practitioners. Breaking this definition down into its components:

- DevOps is a cultural and professional movement, just like heavy metal or hip hop is a culture, or otaku. It's also a professional movement, just as lead guitarist, MC, or animator are professions within their respective cultures.
- DevOps is about building high-velocity organizations. Everyone who practices DevOps is doing it to create these types of companies.
- DevOps is born from the experiences of its practitioners. Although many people assume that the original DevOps practitioners were web innovators, that's not necessarily true. What does matter is that DevOps practitioners are always honing their skills and looking for ways to improve.

CHEFCONF KEYNOTE: [Chef Style DevOps Kungfu](#)

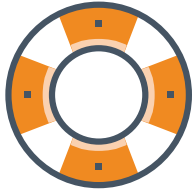


BLOG POST: [The DevOps Engineer is an Optical Illusion](#)



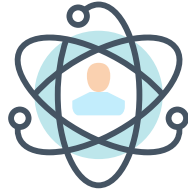
THE FIRST PRINCIPLE OF A DEVOPS PRACTICE

There are several principles that mark someone's practice as a DevOps practice. The first principle is that: **DevOps practitioners design products for the safety, contentment, knowledge and freedom of their peers and their customers.**



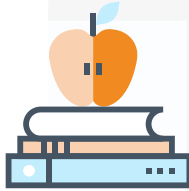
Safety

Safety can mean human safety, it can mean safety of information. It can also have a broader meaning, which is the ability of individuals to act without fear of unintended consequences. DevOps builds organizations where people can express themselves without getting hurt.



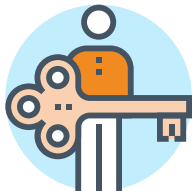
Contentment

Contentment means being satisfied with the things you have. Constant happiness is not a realistic goal for a DevOps practice. You will always have bad days. Contentment is a realistic goal. If you feel good about the people you work with, the systems you're building, and the outcomes you expect, you can achieve contentment. Even on bad days, you'll have a community of people you can rely on to help you get through.



Knowledge

Access to knowledge is a leading indicator of social progress. The goal isn't to minimize the amount of knowledge you need. The goal is to make sure people can access the knowledge they need when they need it. For example, can your engineers see the revenue the company generates and why? If they can't, how can they even begin to understand what products and features will improve the business?



Freedom

Freedom is the power to act or speak or think without hindrance or restraint. DevOps empowers people to act. DevOps means that you trust the people in your organization, no matter what their job, and give them the freedom to do what they know best.



People over products

A final marker of DevOps practitioners is that they put people over products and companies. DevOps practitioners, when they talk about what they're doing and why they're doing it, talk about people before they talk about the company or the products the company makes. DevOps practice prioritizes the human beings who are doing the work.

“ DEVOPS PRACTITIONERS DESIGN PRODUCTS FOR THE SAFETY, CONTENTMENT, KNOWLEDGE AND FREEDOM OF THEIR PEERS AND THEIR CUSTOMERS. ”

DEVOPS AND THE LEAN ENTERPRISE

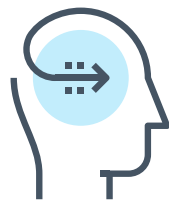
DevOps and Lean principles are complementary. While DevOps was born in the world of software, it embraces many aspects of Lean, which began as a way of streamlining manufacturing processes. [Learn more about Agile Lean DevOps here.](#)

SIX BASIC LEAN PRINCIPLES YOU CAN USE TODAY

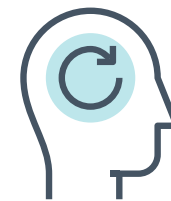
Here are some Lean principles that you will want to incorporate into your DevOps practice, if you're not already doing them.



Eliminate waste. In the spirit of constant improvement, always look for ways to eliminate waste. (In Lean, waste is called muda, which is a Japanese word. Many Lean concepts began at Toyota.) Constantly evaluate your process and ask yourself, "What am I doing that I don't need to do?"



Prefer pull systems to push systems. A pull system is driven by customer need as opposed to a push system, which uses top-down planning. Align your process and resources with market demand.



Practice kaizen, or continuous improvement. DevOps practitioners are always examining their process and trying to make it better.



Practice kaikaku. Kaikaku means radical change. There are times when continuous improvement (kaizen) isn't enough and you need disruption.



Work in small batches. Rather than spending months developing a product and then releasing it, with no input from your customers, try releasing in small iterations, where each iteration represents a minimum viable product. With each iteration, you can incorporate feedback from your audience so that you know you are satisfying market demand.



Embrace failure. Both DevOps and Lean look at failure not as an exceptional event but as a normal one. When you release iteratively and are constantly experimenting and incorporating feedback into your product, failure is to be expected and is an opportunity to learn about what works and what doesn't.

FORGING CONNECTIONS AMONG PEOPLE WITH DIVERSE SKILLS

DevOps is about transforming the way our companies run and part of that transformation is understanding that our companies are about people over products. In any company, you need human beings who do many different things. You need CEOs, you need sales reps, you need software developers, you need marketers, you need system administrators, the list goes on and on. All of these people are necessary and all of them bring their own talents and experiences to the table.

GET TO KNOW THOSE YOU WORK WITH

Not only do you need people with different job skills but you need people from different backgrounds, from different parts of the globe, and people who've done odd things in their lives. All these different perspectives feed back into each other and the more diverse the pool of people, the better the outcome for the company.

To practice DevOps, you also need to form bonds with those people in our company who are different from us. You can do this in very straightforward ways. Take someone out to lunch who doesn't do what you do. Ask them about what they do. Do you know what your sales reps actually do? Do you know why they do what they do? Let's get even more basic. Do you even know the names of your sales reps?

STRONG CONNECTIONS MAKE BETTER PRODUCTS

Aside from making your company a better place to work, there is another very practical reason to broaden your network. Someday you're going to have a big idea that you know will really move the needle on your business.

To make that idea happen, you're going to need feedback and consensus from people who work in many different areas of your company. All the bonds you've formed company wide will come into play.

As you circulate your plan it transmutes and becomes not just your plan but the sales team's plan, and the business development teams plan and the operation team's plan. Everyone has a sense of ownership and is invested in seeing the plan succeed.

It's possible to see a plan fail because your bonds are not inclusive enough. Build consensus for projects by prioritizing the people in your company.

WEBINAR: [Fueling the Love of Chef at Adobe](#)

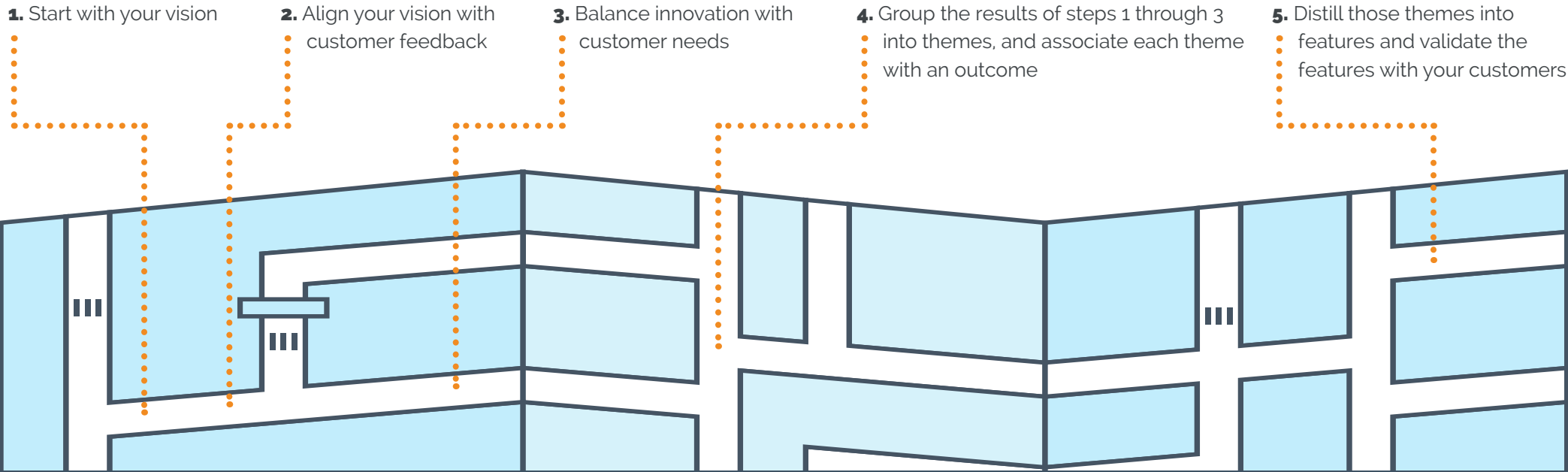


ESTABLISHING ROADMAPS AND THEMES

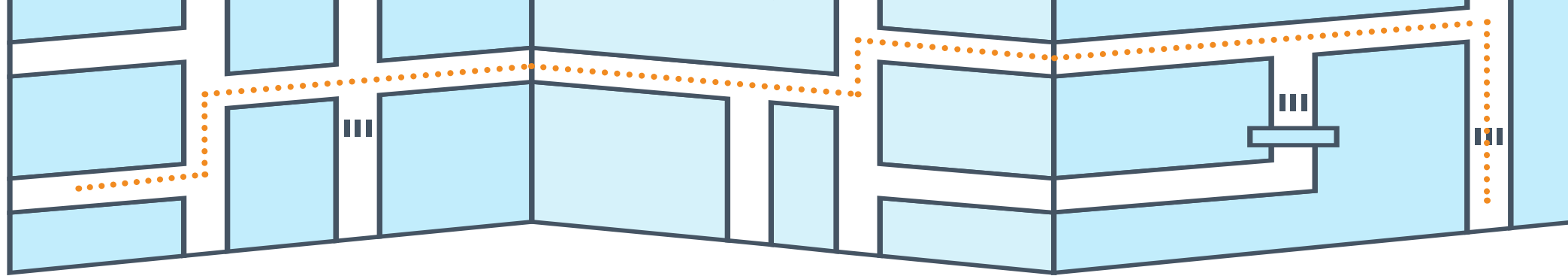
To begin, every product needs a strong value proposition. In other words, it should be a product that people will love. Liking a product isn't enough. Start by focusing on what customers need, not on what they want. A single customer might be adamant about needing a feature but if none of your other customers ask for it, it's probably a feature that only the particular customer wants. If you have fifty customers asking for a feature, it's something your customers need.

DEFINE YOUR ROADMAP

A roadmap can codify your thoughts about how best to discover what customers need and will love. To create a roadmap:



For example, a theme on the Chef roadmap was ecosystem development, and its outcome was that companies other than Chef should sell Chef. Various people had ideas on how to achieve that outcome, and those ideas became the features. Next, a team validated those features with customers. If the features didn't resonate, then the team would come up with different features that could still fulfill the outcome. The team would then do another iteration with customers.



VALIDATE AND REFINE YOUR ROADMAP

In general, after you validate the roadmap with customers, your themes should hold, your outcomes may or may not hold, and the features will shift all the time.

Be suspicious if, working backwards, your features don't change but your outcomes and your themes are no longer true. You're prioritizing features over the actual goals of your roadmap.

As you refine your roadmap, remember that identifying what customers need is just one step toward building products customers love. Think about including features that fulfill a variety of customer expectations. (Note that the following discussion of features is a simplification of the Kano model.)

Some features customers need are so basic, they're taken for granted. However, if they're not there, customers are very unhappy.

The next set of features to include are those that customers notice. They're happy if the features work and unhappy if they don't.

However, to make customers truly love your product, include

delighters. Customers don't expect to see these features, and are delighted when they are a part of the product.

Finally, remember that the best way to create a great roadmap is to get feedback from a wide variety of people within your company as well as a broad range of customers.

WEBINAR: [The DevOps Journey](#)



WHITE PAPER: [Continuous Automation for the Continuous Enterprise](#)

Businesses that embrace apps and digital business as the new customer interface will be the leaders in the coming years and will outperform their peers, industries, and markets.

To become a disruptor and not be one of the disrupted, businesses must deliver software at speed, with efficiency and low risk.



ADDITIONAL RESOURCES

USE CASES

Learn more about how these companies implemented DevOps to increase speed, improve efficiency, and decrease risk.

Transformation at Gannett

GANNETT



Fostering change at Hearst Business Media

HEARST

business media



FURTHER READING

DevOps and the bottom line

In this article, Dr. Nicole Forsgren argues that DevOps doesn't just positively impact the IT function, but also has a dramatic effect on the bottom line.

[Read the article >](#)

Examining tools with a DevOps lens

In this article, Jennifer Davis tackles one aspect of the challenge focusing on examining tools within the industry with a DevOps lens.

[Read the article >](#)

Open Source collaboration

In this article, Jennifer Davis digs into some of the technical aspects of working with tools that enable automation and improve our understanding, transparency, and collaboration.

[Read the article >](#)

LEARN MORE ABOUT CHEF



CHEF™

Create a modern software factory with Chef. Chef's comprehensive suite of automation products can manage it all--infrastructure, run-time environments, applications and even the DevOps workflow itself. With Chef, cross-team collaboration at scale becomes a reality. The most enduring and transformative companies use Chef to become fast, efficient, and innovative software-driven organizations.

www.chef.io >