Progress Chef

Embrace DevOps

Your Guide to the DevOps Lifestyle

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

EBOOK

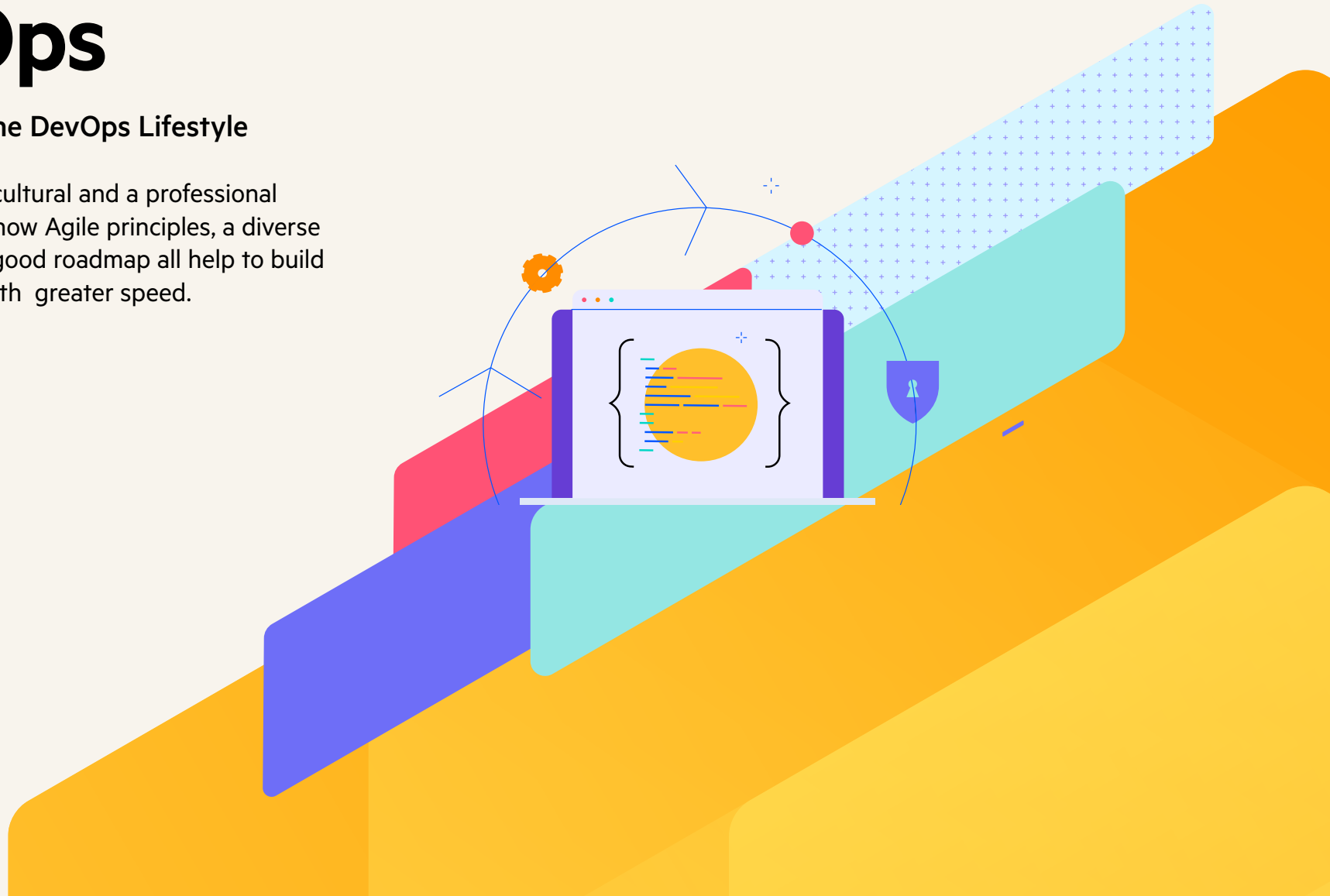


Table of Contents

Foundation of DevOps

Debates about what DevOps is have been going on for several years. However, rather than being a static concept, with a single definition, DevOps may be closer to a practice, with some underlying principles that remain constant, and tools and processes that vary based on an individual organization's needs and practitioner skill sets.

BLOG POST

Why DevOps is for everyone, not just a single team →

"DevOps should be embraced by everyone. It needs support at a leadership level as executives drive culture for an organization."

DevOps Kungfu, Adam Jacob ChefConf 2015 →

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:

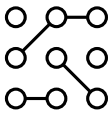
- DevOps is a cultural and professional movement, just like heavy metal or hip hop is a culture, or otaku. It's also a professional movement, just as lead guitarist, actor, or animator are professions within their respective cultures.
- DevOps is about building high-velocity organizations. Everyone who 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.

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.



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.

Knowledge

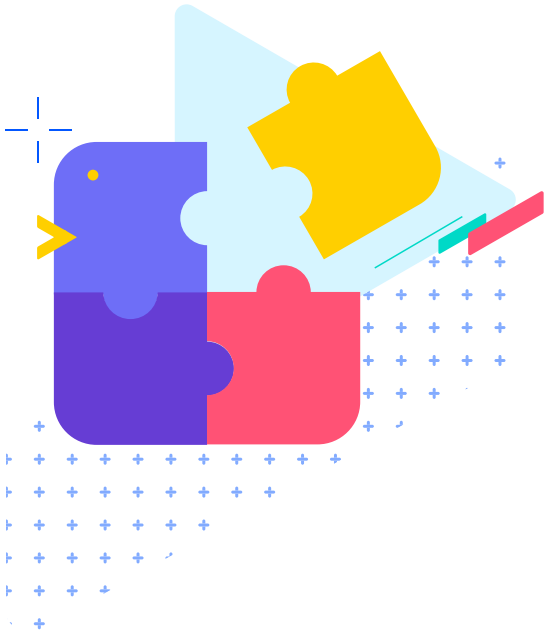
Access to knowledge is a leading indicator of social progress. The goal isn't to minimize the amount of knowledge you need. The goal is to make sure people can access the knowledge they need when they need it. For example, can your engineers see the revenue the company generates and why? If they can't,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

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.

"DevOps Practitioners Design Products For The Safety, Contentment, Knowledge And Freedom Of Their Peers And Their Customers."

Devops and the Agile Enterprise



DevOps and Agile principles are complementary. While DevOps was born in the world of software, it embraces many aspects of agile, which began as a way of streamlining software development processes.

Some Of The Basic Agile Principles You Can Use Today

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

Customer Satisfaction

The highest priority in Agile software development is to enable the continuous delivery of working software while maintaining customer satisfaction.

Continuous Learning And Improvement

Both DevOps and Agile practitioners are always examining process, learning, and improving inefficiencies while simultaneously redefining processes based on new requirements.

Continuous Delivery

Agile development cycles complement DevOps to ensure frequent and sustainable software delivery.

Self-organizing Teams

Agile practitioners take decisions as a group, key contributors drive the decision-making process rather than a singular manager or management team.

Technical And Design Excellence

Agile principles support shorter dev cycles and frequent releases without compromising quality. It prioritizes collaboration between developers and product teams.

Welcome Changing Requirements

Agile principles support changing requirements at every stage of the development cycle.

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 these people are necessary, and all bring their own talents and experiences to the table.

CASE STUDY

Tesco Streamlines Patch Management with Self-Checkout for App Teams

Read the Case Study →

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

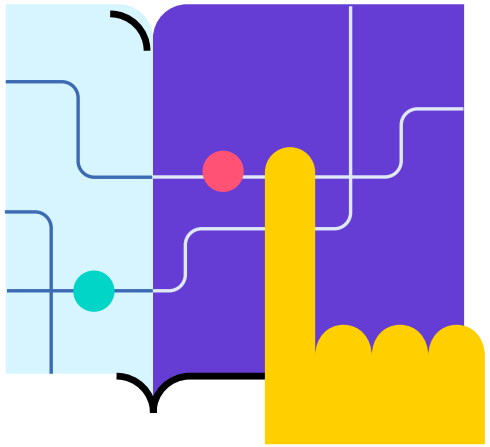
Aside from making your company a better place to work, there is another very practical reason to broaden your network. Someday you're going to have a big idea that you know will really move the needle on your business.

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 team's 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.

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:

1. **Start with your vision**
2. **Align your vision with customer feedback**
3. **Balance innovation with customer needs**
4. **Group the results of steps 1 through 3 into themes, and associate each theme with an outcome**
5. **Distill those themes into features and validate the features with your customers**

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.

ChefConf 21: Accelerate your Journey →

ChefConf 21 Evolving from Infrastructure as code to Policy as Code →

WHITEPAPER

Continuous Automation for the Continuous Enterprise

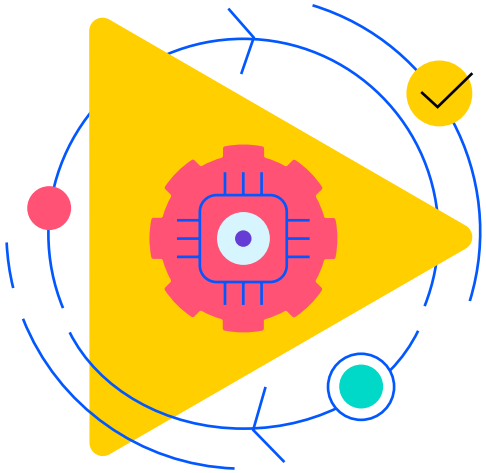
Businesses that embrace apps and digital business as the new customer interface will be the leaders in the coming years and will outperform their peers, industries, and markets.

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

Read Paper →

Current State Of DevOps

DevOps has evolved over the last decade. The adoption of cloud technologies, changes in architecture, platforms, and security regulations have all impacted DevOps culture. Agile methodology combined with DevOps principles has enabled a "shift-left" approach in software delivery. There are shorter and more frequent delivery cycles which are repeatable and sustainable.



Modern DevOps calls for even more efficient collaboration between teams and tools. Some of the fundamentals of modern DevOps are:

- Everything as code: The current IT architecture has adopted an "everything as code" approach. Infrastructure, configuration, compliance, and security can all be transformed into code.

- Automate everything: Codifying everything brings in standardization of processes and policies across the organization. The processes are simplified into repeatable functional code, and this makes it easier to automate the development and deployment processes while reducing human errors significantly.

- Validation: There are many tools used for continuous integration, delivery, and deployment, these are in addition to other tools used for maintaining infrastructure and compliance state. Without validation, there would be no way to understand if everything is functioning as needed and it will be difficult to ensure seamless delivery in a highly agile environment.

These fundamentals are crucial for a reliable and efficient "coded enterprise," and this can only be achieved through CI/CD pipelines. Every enterprise asset has its own development and deployment lifecycle which utilizes a defined set of pipeline actions.

ChefConf 21 Evolving from Infrastructure as code to Policy as Code →

ChefConf 21 : Automatically Deliver Chef: An Agile Workflow →

Additional Resources

Use Cases

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

Edgenuity

[A Lesson in Digital Transformation in the Midst of a Global Pandemic with Edgenuity →]

SAP

[Reshaping the DevOps Process →]

Further Reading

State of DevOps Report 2021
Created by Google and sponsored by leading organizations in the space, this report includes DORA's metrics for speed, efficiency and risk to use in our value selling method.

[Read the article →]

Top DevOps Trends That Will Dominate in 2022
Blog on security boulevard that predicts key DevOps trends for 2022 including edge computing, cloud-native infrastructure and teaming up with security.

[Read the article →]

Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations
Book by Gene Kim, Jez Humble, and Nicole Forsgren that details how to measure the performance of their teams, and what capabilities they should invest in to drive higher performance.

[Read the article →]

Learn More About Progress Chef: www.chef.io

facebook.com/getchefdotcom
twitter.com/chef
youtube.com/getchef
linkedin.com/company/chef-software
learn.chef.io
github.com/chef
twitch.tv/chefsoftware

Progress