ATLASSIAN

CALMS Framework

Assess your ability and measure progress on your DevOps journey.

IAN BUCHANAN

Principal Solutions Engineer

CALMS is a framework that assesses a company's ability to adopt DevOps processes, as well as a way of measuring success during a DevOps transformation. The acronym was coined by Jez Humble, co-author of “The DevOps Handbook,” and stands for Culture, Automation, Lean, Measurement, and Sharing.

Culture

DevOps isn’t simply a process, or a different approach to

ATLASSIAN

All the tooling and automation in the world are useless unless development and IT/Ops professionals work together. Because DevOps doesn't solve tooling problems. It solves human problems.

Think of DevOps as an evolution of agile teams – the difference is now operations is by default included. Forming product-oriented teams to replace function-based teams is a step in the right direction. Include development, QA, product management, design, operations, project management, and any other skill set the long-running product requires. Rather than having one team do everything or hiring "DevOps professionals", it's more important to have product-based teams that can seamlessly work together.

Few things foster collaboration like sharing a common goal and having a plan to reach it together. At some companies, switching suddenly to product-based teams is too much, too soon. So take smaller steps. Development teams can – and should – invite appropriate members of the operations team to join sprint planning sessions, daily stand-ups, and sprint demos. Operations teams can invite key developers to their meetings. It's an agile and organic way to get on the pulse of each other's work, ideas, and struggles.

RELATED MATERIAL

Learn about the benefits of DevOps

See article →

RELATED MATERIAL

Build a DevOps culture

Read more →

Challenges and even emergencies are effective tests of DevOps culture. Do developers, operations, and customer advocates swarm on a problem and resolve it as a team? Do incident post-mortems focus on improving outcomes for next incident instead of pointing fingers? If the answer is "yes," that's a good indication that your organization is embracing a DevOps culture.

The most successful companies are on board with DevOps culture across every department, and at all levels of the org chart. At such broad scale, the term "DevOps" is often too narrow and the term is no longer needed. Such companies have open channels of communication, and talk regularly. They assume keeping customers happy

ATLASSIAN

Automation

Automation helps eliminate repetitive manual work, yields repeatable processes, and creates reliable systems.

Build, test, deploy, and provisioning automation are typical starting points for teams who don't have them in place already. And hey: what better reason for developers, testers, and operators to work together than building systems to benefit everyone?

Teams new to automation usually start with continuous delivery: the practice of running each code change through a gauntlet of automated tests – often facilitated by cloud-based infrastructure – then packaging up builds and promoting them through environments using automated deployments.

Why? Computers execute tests more rigorously and faithfully than humans. These tests catch bugs and security flaws sooner. And automated deployments alert IT/Ops about drift between environments, which reduces surprises when it's time to release.

Another major contribution of DevOps is "configuration as code." Developers strive to create modular, composable applications because they are more reliable and maintainable. That same thinking can be extended to the infrastructure that hosts them, whether it lives in the cloud or on the company's own network.

"Configuration as code" and "continuous delivery" aren't the only types of automation seen in the DevOps world, but they're worth special mention because they help break down the wall between development and operations. And when DevOps uses automated deploys to send thoroughly tested code to identically provisioned environments, "works on my machine!" becomes irrelevant.

Lean

When we hear "lean" in the context of software, we usually think about eliminating low-value activities and moving quickly – being scrappy and agile. Even more relevant for DevOps are the concepts of continuous improvement and embracing failure – which lay the foundation of an experimental mindset.

users of your product.

We have agile development to thank for making continuous improvement a mainstream idea. Early adopters of the agile methodology proved that a simple product in the hands of customers today is more valuable than a perfect product in the hands of customers six months from now. If the product is improved continuously, customers will stick around.

And guess what: failure is inevitable. So you might as well set up your team to absorb it, recover, and learn from it (some call this “being anti-fragile”). At Atlassian, we believe that if you’re not failing once in a while, you’re not trying hard enough.

In the context of DevOps, failure is not a punishable offense. Teams assume that things are bound to go pear-shaped at some point, so they build for fast detection and rapid recovery. Postmortems focus on where processes fell down and how to strengthen them – not on which team member messed up the code. Why? Because continuous improvement and failure go hand in hand.

Measurement

It’s hard to prove your continuous improvement efforts actually improve anything without data. Fortunately, there are loads of tools and technologies for measuring performance, like how much time users spend with your product, whether that blog post generated any sales, or how often critical alerts pop up in your logs.

Although you can measure just about anything, that doesn’t mean you have to (or should) measure everything. Take a page from agile development and start with the basics:

How long did it take to go from development to deployment?

How often do recurring bugs or failures happen?

How long does it take to recover after a system failure?

How many people are using your product right now?

feature usage, customer journeys, and service level agreements (SLAs). The information you get comes in handy when it's time for road mapping and spec'ing out your next big move.

All this juicy data will help your team make decisions, but it's even more powerful when shared with other teams – especially teams in other departments. For example, your marketing team wants shiny new features they can sell. But meanwhile, you're seeing high customer churn because the product is awash in technical debt. Providing user data that supports your roadmap – even if it's light on features and heavy on fixes – makes it easier to build consensus and get buy-in from stakeholders.

Sharing

As much as we wished that there was a magic wand to transform all teams into high-performing DevOps teams, DevOps transformations require a blend of practices, cultural philosophies, and tools. But like you've read, the benefits to breaking down the Development and Operations siloes are well worth it – increased trust, faster software releases, more reliable deployments, and a better feedback loop between teams and customers.

Embracing DevOps is no small task. Yet given the right mindset, effort, and tools, an organization can undergo a DevOps transformation that yields significant benefits.

The long-standing friction between development and operations teams is largely due to a lack of common ground. We believe that sharing responsibility and success goes a long way toward bridging that divide. Developers can win instant goodwill by helping to carry one of operations' biggest burdens: the pager (a figurative construct these days). DevOps is big on the idea that the same people who build an application should be involved in shipping and running it.

In conclusion...

Out of this idea comes the phrase, "you built it, you run it," which fosters a hands-on approach accross teams. This doesn't mean that you hire developers and simply expect them to be excellent operators as well. It means that developers and operators pair

ATLASSIAN

review at all.

Teams that embrace DevOps often have a rotating role whereby developers address issues caught by end users while, at the same, troubleshooting production problems. This person responds to urgent customer-reported issues, creating patches when necessary, and works through the backlog of customer-reported defects. The “developer on support” learns a lot about how the application is used in the wild. And by being highly available to the operations team, the development teams build trust and mutual respect.

Atlassian created Open DevOps for teams that look to build the toolchain they want, with the tools they love, thanks to integrations with leading vendors and marketplace apps. **Try it now.**

IAN BUCHANAN

Ian Buchanan is a Principal Solutions Engineer for DevOps at Atlassian where he focuses on the emerging DevOps community and the application of Jira, Bitbucket, and Bamboo for better continuous integration and continuous delivery. While Ian Buchanan has broad and deep experience with both Java and .NET, he is best known as a champion of lean and agile practices in large enterprises.

During his career, he has successfully managed enterprise software development tools in all phases of their lifecycle, from cradle to grave. He has driven organization-wide process improvement with results of greater productivity, higher quality, and improved customer satisfaction. He has built multi-national agile teams that value self-direction and self-organization. When not speaking or coding, you are likely to find Ian indulging his passions in parsers, meta-programming, and domain-specific languages.

SHARE THIS ARTICLE

NEXT TOPIC
Team Topologies →

ATLASSIAN

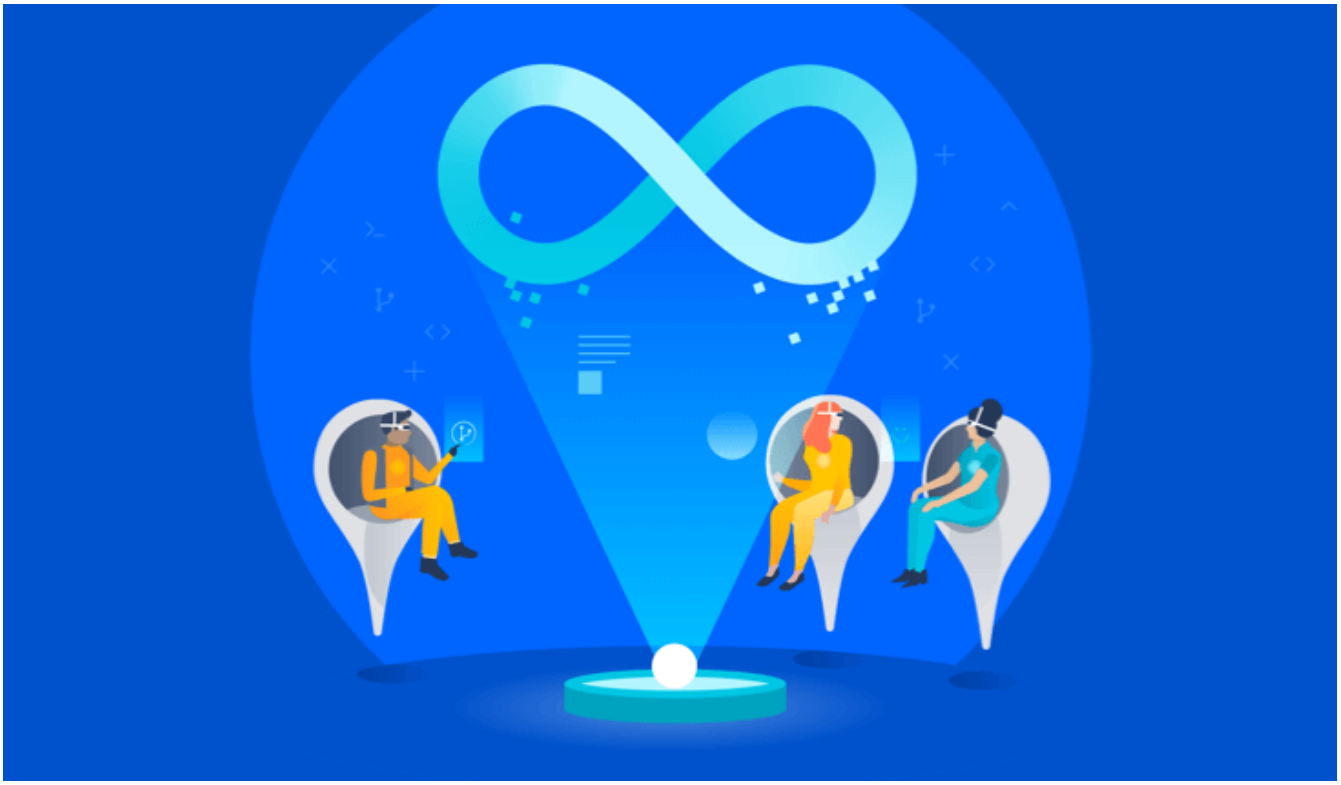
Recommended reading

Bookmark these resources to learn about types of DevOps teams, or for ongoing updates about DevOps at Atlassian.

DevOps community

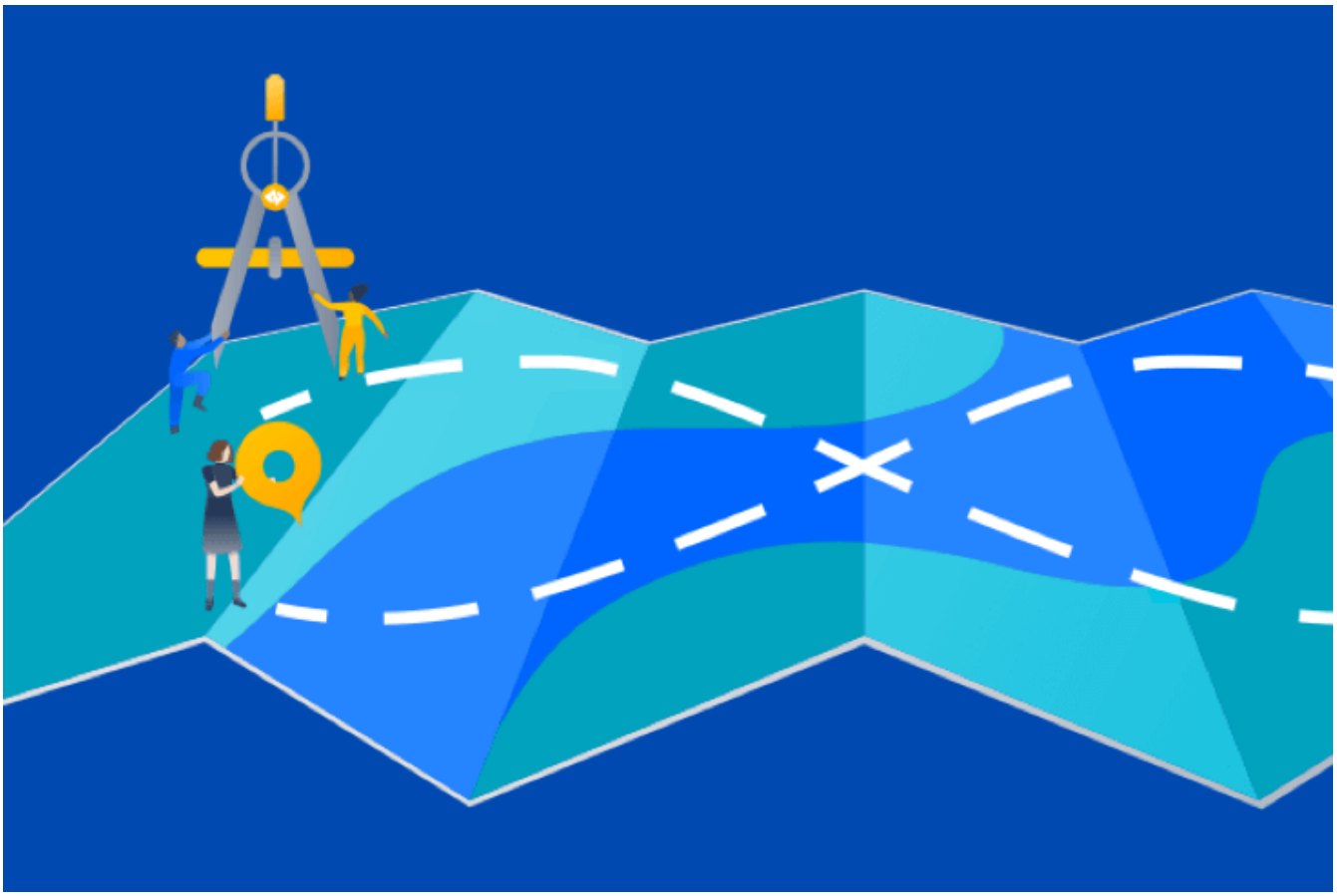
Learn more →

ATLASSIAN



DevOps learning path

Learn more →



ATLASSIAN

Sign up for our DevOps newsletter

Email address

Sign up

ATLASSIAN

Jira

Jira Align

Jira Service Management

Confluence

Trello

Bitbucket

See all products

RESOURCES

Technical support

Purchasing & licensing

Atlassian Community

Knowledge base

Marketplace

My account

Create support ticket

ATLASSIAN

Training & certification

Documentation

Developer resources

Enterprise services

See all resources

Privacy Policy Terms Impressum English ▼