THE ENTERPRISERS PROJECT

HOW TO EXPLAIN DEVOPS
IN PLAIN ENGLISH

HOW TO EXPLAIN DEVOPS IN PLAIN ENGLISH

What is DevOps? 6 definitions and analogies

We asked DevOps experts how they explain DevOps in the shortest, simplest words so that anyone could understand its value, regardless of their technical background. See below for a few punchy definitions and some helpful analogies to help you tell your DevOps story.

1. DevOps is a cultural movement

"DevOps is a cultural movement where both key stakeholder groups (software developers and IT operations) agree that software is not really adding any value until it is used by somebody – customers, clients, employees, etc," says Eveline Oehrlich, chief research analyst at DevOps Institute. "Due to this, both teams are ensuring together that software is delivered with speed and quality."

2. DevOps empowers developers

"It's commonly stated that DevOps allows for faster delivery to production by implementing and leveraging automated processes. To me, it's far more fundamental," says Jai Schniepp, director of secure DevOps platforms at Liberty Mutual. "DevOps empowers developers to own, run, and manage the end-to-end delivery of an application or piece of software. It eliminates the confusion around ownership and drives teams toward an automated, developer-managed infrastructure."

3. DevOps is a collaborative approach to building and delivering software

"Simply stated, DevOps is an approach to building and delivering computer software in which everyone works together," says Gur Steif, president, digital business automation at BMC.

> "DevOps empowers developers to own, run, and manage the end-to-end delivery of an application.

4. DevOps is like an assembly line

"I would compare DevOps to a manufacturing assembly line," says Steif. "The idea is to design and build all the parts up front, in a way that allows all the pieces to fit together. For an assembly line to work, components must be designed to combine seamlessly. The folks who design and build engines must think about the chassis and motor mounts. The folks who build brakes must think about the rims and tires, and so forth. That's how it needs to be in software. The developers who write the business logic or the user interface must think about the database that stores customer information, the security that protects user data, and how it all operates when the service is exposed to what may be millions of users."

"Getting people to collaborate and think about work being done by others rather than focusing on their individual task(s) is the biggest obstacle to overcome. If you manage to achieve that, you stand an excellent chance of achieving digital transformation," Steif adds.

5. DevOps is a recipe – combining people, process, and automation

Jayne Groll, CEO at DevOps Institute, has a great cooking analogy for explaining DevOps: "DevOps is a recipe that relies on ingredients from three major categories – people, process, and automation," she says. "Most of the ingredients can be adapted from other well-known practices and sources such as Lean, Agile, SRE, CI/CD, ITIL, leadership, culture, and tools. The secret behind DevOps is how these ingredients are blended and in the right proportions (like any good recipe) in order to increase flow and value to the customer."

6. DevOps teams are like NASCAR racing crews

"When I talk about desired outcomes for a DevOps initiative, I mention NASCAR or F1 racing," says Chris Short, principal technical marketing manager, cloud platforms, Red Hat, and publisher of the DevOps'ish newsletter. "Crew chiefs for these race teams have one mission: Finish in the best place possible with the resources they have available while overcoming the adversity thrown at them. Race teams don't think from start to finish; they flip the table to look at the race from the finish line to the beginning. They set a goal, a stretch goal, and then begin working backward from those goals to determine how to get there. Work is delegated to team members during race week to push towards the set of objectives that get to the desired outcome."

> Race teams don't think from start to finish; they flip the table to look at the race from the finish line to the beginning.

"Race teams practice pit stops all week before the race. They are in weight training and cardio programs during the week to keep them physically ready for the grueling conditions of race day. They continually collaborate to address any issue that might come up. Similarly, software teams should practice releases often. If the safety systems are in place and practice runs have been going well, release to production happens more frequently. Speed makes things safer in this mindset," says Short.

"It's not about doing the 'right' thing," he adds, "it's about addressing as many of the things that could keep you from the desired outcome as possible. Collaborate and adjust based on the real-time feedback you observe. Expect anomalies and work to improve quality so the impact of those anomalies on the goal is minimized. These are the expectations for everyone in a DevOps world."