GitHub

Seven DevOps tips for faster app development

DevOps is about speed: faster application development, faster updates and continuous development, and faster shipments, all of which lead to shortened systems development lifecycles. Recent research shows that high-performing DevOps teams recover from downtime 96 times faster, have a five times lower change failure rate, and deploy code 46 times more frequently. In short, successful DevOps teams can deploy code and recover in a matter of hours, not weeks.

It's no small wonder organizations of just about all sizes and across all vertical markets and global geographies have embraced DevOps. Another survey found that nearly three-quarters of organizations surveyed globally have already adopted DevOps. However, all this well-founded enthusiasm and DevOps potential is tempered by real challenges. This same study also discovered that eliminating process bottlenecks to speed up releases shot up to the top of the list of DevOps challenges. Streamlining collaboration and selecting the right suite of automation and other tools followed in order of leading challenges.

This paper lists the top seven DevOps tips that high performing teams are following to maximize their DevOps efforts and investments.

1 DevOps culture matters, especially when built from the grassroots.

The author of the annual State of DevOps Survey, now in its seventh year, wrote, "The most important themes for us have always been empowering teams to do their best work, overcoming the cultural divide between dev and ops."[2] Extensive analysis of historical data from this survey shows that DevOps success is derived most often by starting with small, grassroots efforts. Next, developers need a lightweight platform with which they can securely share early successes and proven practices with other teams. This sharing then spreads to multiple teams within a single department, and ultimately to other departments. Said another way, don't try overdoing it with DevOps efforts from the outset. This is precisely the practice followed by the world's largest business software maker and a global leader in logistics.

2 Embed security deeply into DevOps right from the start.

This means automating security configurations to the greatest extent possible. As DevOps organizations evolve from their grassroots beginnings, security policy becomes an intrinsic part of operations and not something IT scrambles to prove to meet audit requirements. As the world's leading software development platform, GitHub, provides solutions that are GDPR-compliant, encrypt all data in transit, and support industry-leading control considerations with the Cloud Security Alliance CSA-CAIQ Assessment. Organizations also have the option to host code on their own infrastructures using the self-hosted product GitHub Enterprise Server, built to support strict security, auditing, and compliance requirements. This solution is governed by existing local information security controls, including firewalls, VPNs, IAM and monitoring systems. This on-premises solution can aid in the effort of avoiding many of the regulatory compliance issues common with cloud-based solutions. Thus the focus can be turned away from keeping away the auditors toward keeping the data of the organization and of its customers secure. And most importantly, teams and developers can build more secure code faster.

DevOps organizations and teams need to shift focus from the details of audits and compliance to developing secure code as an efficient, holistic process.

3 To the extent possible, automate.

The conversation about automation in DevOps usually settles on automation of systems configuration, workflows and systems provisioning. This infrastructure automation tackles the common challenge of keeping developers' work on pace and not running ahead of operations capabilities, which is directly related to the ability to deploy software. Automation serves up another key feature, namely the creation of more broadly available self-service in later stages of development, which of course leads to greater efficiency. Using an array of templates, GitHub allows users to automatically configure workflows in order to keep the status of project board cards in sync with associated issues and pull cards. Users can automate actions based on triggering events to eliminate some timeconsuming manual tasks in managing a project board. And users can copy a GitHub project board to reuse its customization for similar projects down the road.

4 Exploit the benefits of an open platform in providing transparency.

Collaboration, which is a main driver of efficiency in DevOps, lays at the heart of open source, wherein software is developed in a highly collaborative, collegial and public process. Open source revenues globally have grown dramatically in recent years as organizations of all types have aggressively build open source into their DevOps methodology. Web-based software platforms allow users to track and follow the actions of very large numbers of other developers, no matter where they are located or for whom they work. An open platform also gives developers and teams collaboration both within and outside their teams. In doing so they can not only leverage open source code, but also exploit security bug updates for building more secure code quicker. This openness yields the potential for optimal transparency, and this can radically improve collaboration and learning in the highly complex business of software development.

One study shows that in such a social development environment, users make a rich and mutually rewarding set of connections and social inferences from this networked activity, such as inferring someone else's technical goals and vision when editing code, or better guessing which projects have the best chances of success in the long run. Users then combine such inferences into more effective strategies for coordinating work and advancing their own technical skills.[3]

5 Integrate with the widest set of tools possible.

In the latest version of an annual survey on enterprise DevOps deployments, the top two challenges to more effective DevOps processes are people oriented (eliminating process bottlenecks and streamlining developer collaboration.) But a close third on this list is selecting and implementing application monitoring, management, and automation tools. The ability to integrate DevOps efforts with the widest array of tools possible is a cornerstone of DevOps success. Whether it is to promote continuous integration or code review or code editing, it is essential that developers can find just the right tools for every step of the development process, and then begin to use them on the spot. GitHub offers access to literally hundreds of tools designed to help development teams better communicate, to automate their work, and to develop better software. There are times when only a custom-developed tool is needed for a specific task. GitHub enables users to create such customized tools with greater access to data using GitHub GraphQL API, which is the same API the company used to build GitHub.

GitHub is how people build software. Trusted by over 40 million developers and half of the Fortune 500, GitHub helps DevOps teams of every size collaborate securely—and deliver better customer experiences, faster.

6 Maximize open source efficiency with innersource.

A fast-growing number of organizations are adopting and using innersource, a development methodology harnessing best practices from large-scale open source projects. These kinds of highly complex, multi-development team efforts require coordination across hundreds if not thousands of developers and teams. Thus, innersource is proving to be a formidable weapon to combat the many challenges posed by large-scale DevOps projects, and a great way to boost efficiency through code reuse. Companies can access and leverage open source best practices behind the innersource firewall. GitHub is home to the single largest open source community in the world. Developers and teams on the GitHub platform can view, modify and distribute a project for any purpose they choose, as enforced by open source licenses. And GitHub offers a broad series of self-help guides for finding users for your project; building welcoming communities; maintenance and governance; and open source metrics, to name a few.

7 Promote continuous feedback, integration, change management, and deployment.

Discovering common practices and industry standards in DevOps is one thing. The more important matter is to integrate them and embed them deeply into a process flow that crosses data and information silos from development through deployment and then to operations. To this end, continuous integration builds on test automation to ensure that as new code is added to an application, continuous integration focuses on quality at the application level as new or modified code is integrated. Change management brings operations in the picture to allow input on what other systems could be impacted and what consequences or opportunities a change might expose at a broader level. Incremental deployment strategies often fall victim to operational constraints that can limit or stop continuous deployment in its tracks. This is a people issue, not a technology problem. And the remedy is the inclusion of all stakeholders in application development, operations and support. This kind of continuous intradepartmental input will significantly boost the likelihood that a great application will also meet user needs and expectations.

GitHub is how people build software. Trusted by over 40 million developers and half of the Fortune 500, GitHub helps DevOps teams of every size collaborate securely— and deliver better customer experiences, faster. To start your free trial or learn more about GitHub's on-premises and SaaS solutions, visit **https://github.com/business.**

[1] "2017 State of DevOps Report," Puppet, 2017.

[2] "2018 State of DevOps Report: Practical guidance for your DevOps evolution," Puppet blog, September 2018.

[3] "Social coding in GitHub: Transparency and Collaboration in an Open Software Repository," jsntsay.com, 2012.

GitHub Enterprise