

Introduction

The increasing adoption of DevOps by most of the organizations across the globe clearly indicates its potential as a key enabler to achieving scale. Implementation of DevOps practices helps an organization deliver faster, better, high-quality and reliable software relying on the culture of cooperation and collaboration among all functions of an organization. It calls for fundamental cultural changes and modification of legacy programming practices. Here are the core DevOps best practices that help an organization achieve the goals of effective communication and collaboration, smoother operations and bug-free code..

Credit: <http://www.cigniti.com/blog/10-best-practices-for-successful-devops-implementation/>

1) Evaluate need to implement DevOps practice

Align your IT goals with Business goals. The need for implementation of DevOps should be business-driven. It should not be implemented just because it is the latest trend, but your development process for the business goals should demand this change.

2) Break org silos & encourage collaboration

DevOps practices demand to break down functional silos among various disciplines in IT. The philosophy of DevOps essentially is that development, operations, and other functions must work closely by cooperating and collaborating among themselves. Breaking down organizational silos improves communication among the teams enabling accessibility to information to everyone about what was done in the past, people involved and the associated results. It helps in better decision making, in turn, fetches better output and better ideas..

3) Put Customer satisfaction at the center

Organizations must keep adapting themselves to the ever-changing customer demand and deliver services / solutions that meet, rather exceed, customer expectations regarding time, functionality and performance. This is possible only by embracing the change in culture that stresses on team effort, transparent communications, and commitment to customer satisfaction, etc. Without the support of all the key business stakeholders, DevOps will not be successful. Right from defining the requirements, prototype development, unit/integration/regression testing, to deployment, everyone should be involved..

4) Start small and then scale up

Achieve DevOps approach for faster and smaller release cycles and then adopt at scale. Some quick successes consolidate the belief of various stakeholders in the new approach. Moving the IT culture away from silos need trust and acceptance in the new philosophy. Also, Organizations need to upskill current talent rather than hiring from outside. It enables the existing employees to achieve some early success which helps in improving their confidence on adopting DevOps.

5) Automate wherever possible

Automation enables faster execution throughout the SDLC, keeping up with the speed of DevOps. Automation can be employed and extended to code development, middleware configuration, database and networking changes, and to essential testing including regression testing and load testing. Automation saves time and efforts of developers, testers and operations personnel and, in turn, total costs.

6) Select compatible tools

The automation tools to be used in DevOps should be chosen depending upon how they react with another tool. It is recommended to choose a toolset which is compatible with your IT environment. Ensure that you adopt tools that are suitable to the rest of the toolchain that is existing. Tooling decisions should be taken wisely considering the overall tool compatibility for your organization. It is usually effective if the tools that you choose are from a single vendor because such tools must have been closely integrated with each other. Careful selection of tools reduce the conflicts that they possibly create between development and operations..

7) Define team & individual performance reviews

When the IT culture has to be collaborative, it requires an evaluation of team's as well as individual's performance in the team. Since cooperation and collaboration are at the core of DevOps, performance reviews for developers and operations personnel should mostly be based on their teams' ability to meeting their development and deployment goals.



8) Ensure real-time visibility into the project

For a cross-functional IT organization, it is important to have a project management tools that provide real-time visibility into a project or an application is required. It makes the project coordination among different functions easier. All the stakeholders need to understand in which phase the project is exactly in the development to the deployment process. Advanced project management tools have built-in automation that eases getting the information by displaying who and what are the crucial resources for the current tasks of the project..

9) Integrate and deliver continuously

Embracing DevOps without implementing Continuous integration and Continuous Delivery will be inefficient and unsuccessful. Continuous Integration is one of the key components of agile processes which enables developers to develop a software in small, regular steps by immediately detecting defects and providing feedback. Continuous delivery is an extension of continuous integration. Continuous Delivery approach ensures that every new or revised requirement is rapidly and safely deployed to production with quality by delivering each and every change to a production-like environment and making sure that the software / application functions as intended through rigorous test automation. It confirms that the software functions as intended through rigorous automated testing. Hence, Continuous Integration and Continuous Delivery should not be neglected for successful DevOps implementation..

10) Better results with monitoring & feedback

To know if the software or application is performing as desired while the environment is unwavering, continuous monitoring is essential. The Operations team has to ensure that the applications are performing at the optimal levels. They may work with the development team to build monitoring and analytics capabilities right into the applications being developed.

Finally, DevOps is a set of principles and practices that facilitates an organization to make their delivery of software / applications lean and efficient, while leveraging feedback from end-users that help to continuously improve. Feedback mechanism improves the processes of delivering an application.

