TO THE NEW

JAN 2016

Top Devops Practices and Principles to Improve IT Efficiency

1. What is DevOps?

DevOps, the word is wider than it feels to the ears. So, what is DevOps?

It is not just Development and Operations working together towards a common goal; it is a combination of everything, it is a practice, it is a thought. In other terms, it's the adoption of efficient agile methodology in Ops team. It is a process of embracing a mindset in which Ops team uses the methodology of development team to carry out their operations. It is even better when it comes to implementing DevOps practices as a daily driver.

So, DevOps is the movement to bring Development and Operations closer – communicate and collaborate across Dev and Ops. Development here is inclusive of development and testing. All need to work towards a common goal. Inputs from one part of the system feed into other part of the system. DevOps is thinking long-term rather than short term. It brings in the thinnest lines between the various teams and combines together all the best practices available like automation, agile etc. DevOps mindset also focuses on automating the daily tasks, which the people have been doing for so long leading to the overall productivity of the organization. It is a way of thinking that is beneficial to other departments such as sales, analytics, finance and HR.

"DevOps is agile principles followed beyond the code."

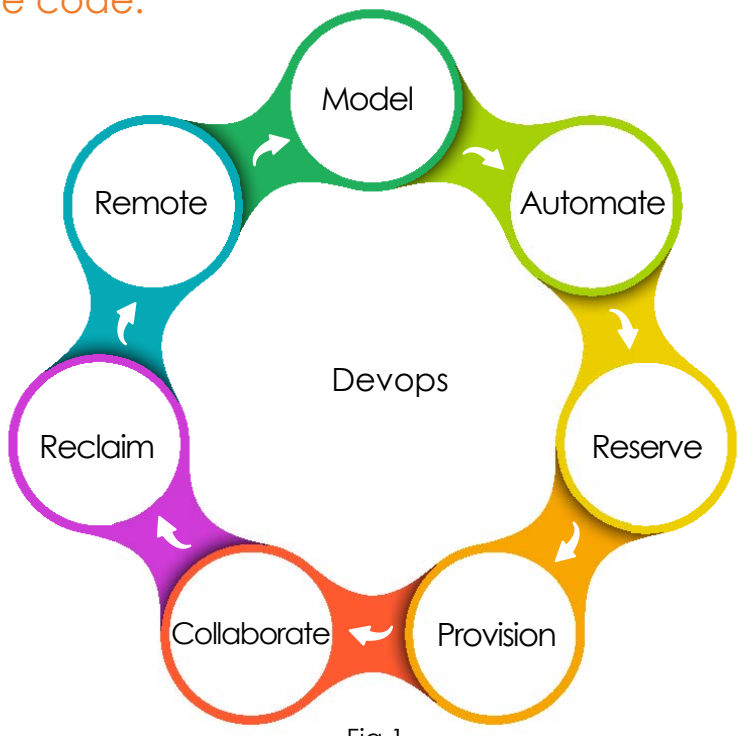


Fig.1

2. How DevOps adoption helped Netflix achieve high productivity?

Before DevOps adoption, Netflix was going through a hard time in terms of code building, its management, and deployment to the systems in their in-house data center. The pace which was supposed to get faster in terms of delivery, got slowed down due to increased error rates in code compatibility with underlying infrastructure, network outages etc.

As perfection couldn't be achieved using previous strategies, Netflix decided to adopt DevOps culture in their organization. The teams were reconstructed at this level as few IT Ops were moved to developers team to understand the need of developers and to increase their efficiency. Previously, developers were engaged in heavy discussions with their Ops team to get their requirements, bug fixes at system level, modifications in current configuration, forecasting the capacity but now they needed few seconds to do the changes by themselves.

Netflix calls it partial DevOps adoption plus NoOps culture. Few points were missing like continuous deployment to all the stages throughout the pipeline without any human intervention, use of centralised configuration management tools like Chef, Puppet and Ansible. The adoption of such kind of automation relies upon the organizational structure, mindset of the top level management, required expertise and the application's workflow. Freedom and responsibility to developers, automation of Ops activity leading to the restructure of the organization are the main intents of DevOps adoption.

Case Study Source: http://www. slideshare. net/adrianco/ netflix-what-changed-gartner-catalyst

Few take-aways from the above case study :-

- **Automating tasks using tools that require manual efforts like load testing, automated deployments, version controlling etc.**
- **Knowledge sharing between Ops and Dev team**
- **Learning attitude of the team members to try new technologies and get the maximum benefit**
- **Adopting a behaviour of continuous experimentation**
- **Identifying the areas of problem and their resolution**

Let us look at some numbers now

Statistics about DevOps adoption and its future.

Source: Gartner report: http://www.gartner.com/newsroom/id/2999017

"Gartner Says By 2016, DevOps Will Evolve From a Niche to a Mainstream Strategy Employed by 25% of Global 2000 Organizations"

The future for DevOps seems to be showing a positive growth in the coming years. Gartner says that DevOps culture has picked up its pace with a total for DevOps tools reaching $2.3 billion in 2015, an increase from $1.9 billion in 2014.

The study also says that DevOps culture strategy will be adopted by 25% of the Global 2000 organizations by 2016.

3. Top Ten DevOps Practices

1. Tests Automation

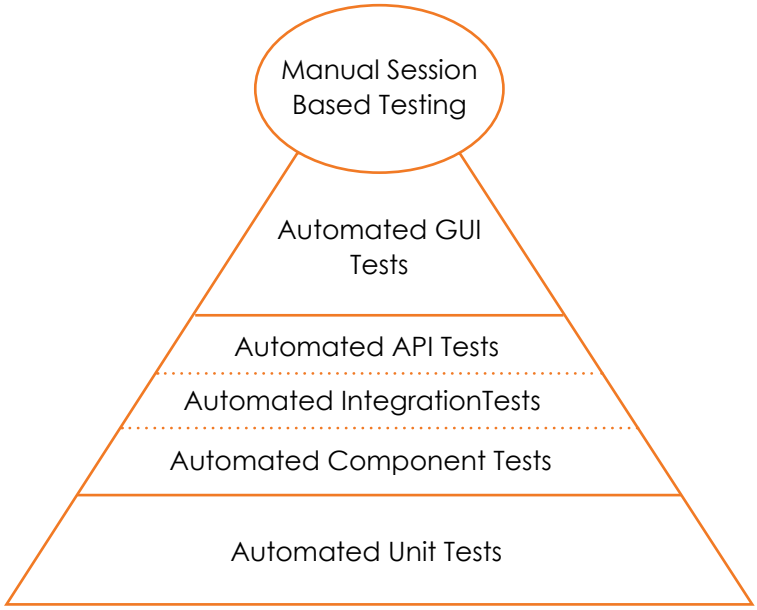
There is a need to test code and underlying infrastructure but in an automated way rather than doing everything manually. To do it in an automated way, the tools like webrat can be used for automated integration testing. Automation here means that the code will be the input and the final report specifying the details of all the bugs and test cases will be the output. This is a way you can involve ops in testing and hence call it a good DevOps practice.

How is it done:

- **Identifying Test Cases and Test Scenarios:** Select from the manual testing cases the ones that can be automated
- **Selection of Automation Tool:** Depending on the nature of the application and the set of resources on the team, selection of tool is done
- **Setting up the Environment**: Sometimes a separate environment/machine is set up to run automated test cases
- **Run Test Cases:** Automated scripts are executed on UAT several times to find inconsistencies and defects
- **Analyzing Test Results:** Different options are provided by the tool to generate test results reports

Ideal Software Testing Pyramid

watirmelon.com



2. Centralized Configuration Management

"Configuration management is the process of standardizing resource configurations and enforcing their state across IT infrastructure in an automated yet agile manner." [PuppetLabs]

Configuration Management believes that configuration change of the resources and their management will be done in an automated manner with the help of a standardised tool from a centralised place. To do so there are various tools available such as Chef, Puppet, Ansible, etc.

The amazing capabilities and opportunities of building proximity based contextual experiences for customers makes beacons a sure shot deal for brands. With beacons based contextual solutions, a brand can achieve the following:

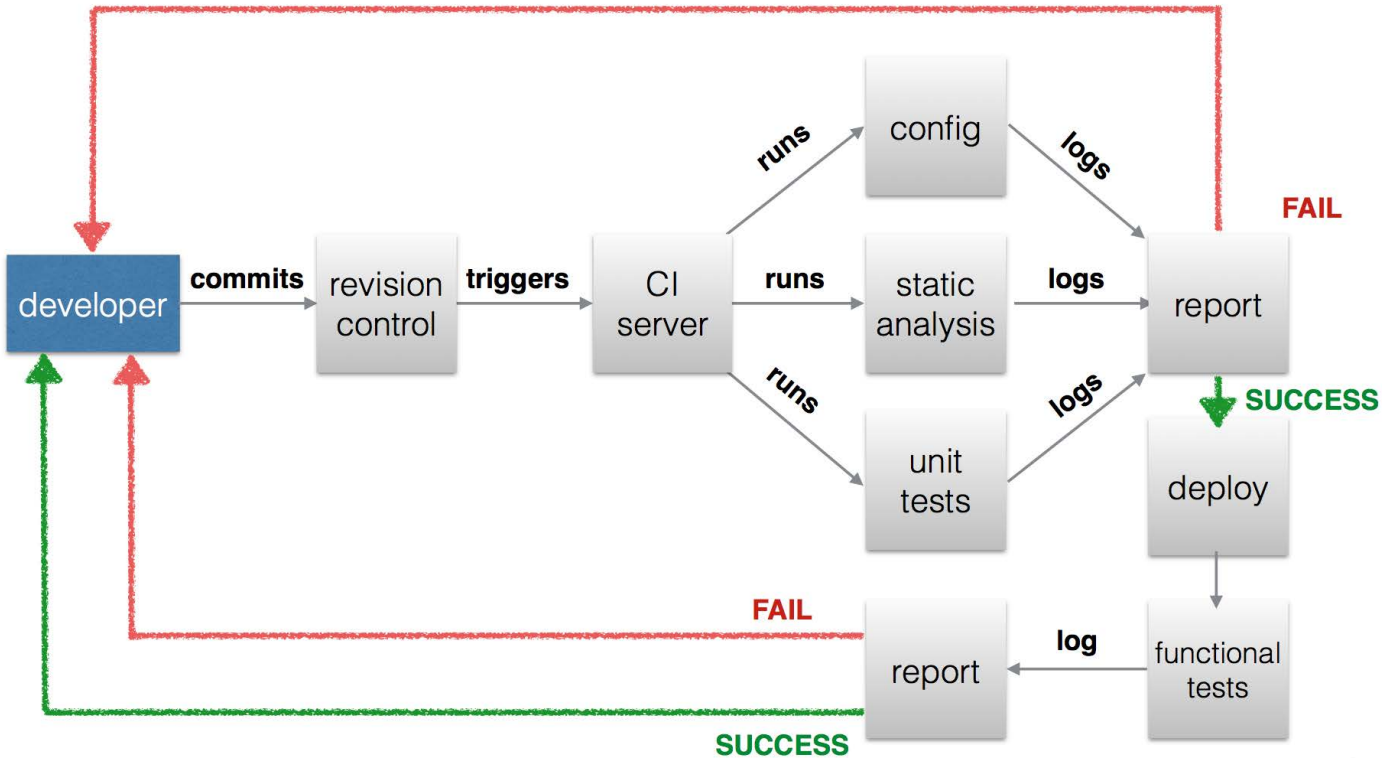
For eg., if the infrastructure consists of 100+ running servers and you have to make the configuration changes in a file then it is better to automate this process rather than doing it manually. Set it up once and fire it when needed again. The best part is that you can configure all environments from a centralised place.

Keeping track of all the changes and configurations should also be considered. If required, you can revert to earlier configurations.

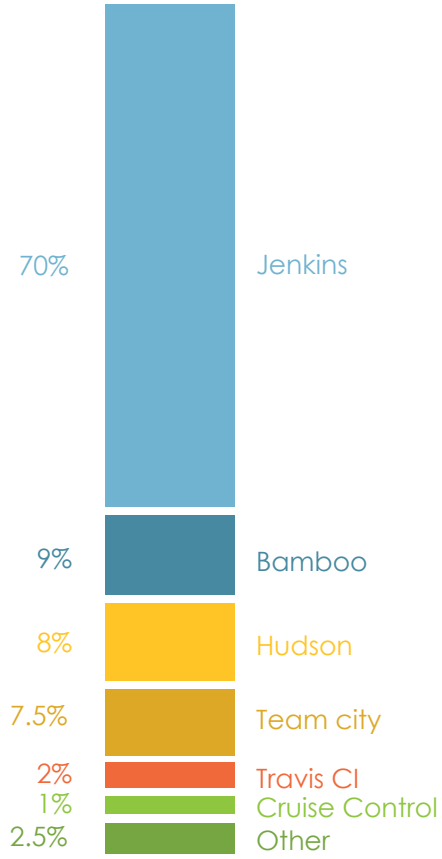
Chef case study for facebook: https://www.chef.io/customers/facebook/

3. Continuous Integration

The deployment process is a combination of many processes. It involves code development, versioning, testing, deployment followed by post-deployment tasks like shooting a mail to the concerned person or the team.To automate all these tasks, DevOps practice aims at using tools like Jenkins and Bamboo, which are capable of executing the desired tasks in proper order defined by the team.The use of such tools reduce human intervention in the deployment process.When the team has all the things proper in place, then a tool like Jenkins is enough for executing all the jobs in right order and doing all the pre as well as post-deployment tasks as per the requirement. So, it saves time, and efforts, keeping the build pace faster.



Continous Integration (CI) server used*



*The results were normalized to exclude non-users

Analysis By : REBEL LABS

4. Continuous Delivery

Continuous delivery is the way in which the delivery process resulting from new repository changes starts flowing from staging to any environment, or even production. Continuous delivery ensures that there won't be any issues, breakpoints in moving the code from staging to any other environment. Continuous delivery is MUST if you already have adopted the DevOps practices. The need for continuous delivery comes from the fact that if the production code starts behaving abnormally, you can quickly fix that bug in real time after passing the changes through the whole delivery pipeline to production environment without causing big loss to production site.It aims at making the whole infrastructure flexible enough so that the required changes can be easily propagated to a specific environment without any barriers.

5. Continuous Deployment and Versioning

Continuous deployment is the process to automate the deployment flow from staging to production environment based on certain conditions. However, some organizations treat it as an optional process. The process generally varies withorganization. The deployment process can be automated by means of tools like Jenkins (continuous integration), Capistrano and Chef etc. The deployment process automation says that the code will automatically be deployed to production environment as it passes all the test cases in QA environment followed by UAT environment and so on.

Code versioning ensures that the must-have multiple versions of the codes are available at proper places. Without, version control system, it would be really difficult to make a large number of developers work on the same project simultaneously. In case of any errors or issues, managers can easily locate the bugs and revert the code back to the previous version without any hiccups.

6. Enriched Monitoring

Whether the infrastructure is on cloud or on local data centers, monitoring is a must. Monitoring of even the smallest entity of the infrastructure is a must-do task. Monitoring of web pages load time, slow query logs and other important details can be very helpful in optimizing the performance of the application.

Buggy code from one of the recent deployments or a third party component that the team just integrated into the application can cause the overall application to fail and thereby requiring the need for application checks.

DevOps needs not just monitoring, but pro-active monitoring. Proper monitoring can be important for sales, marketing and finance department as well. There are a lot of tools that give enriched, deep monitoring of everything that affects the project/application. One such online tool is geckoboard, which combines a lot of metrics related to infrastructure, application, analytics, sales, graphs for different tools etc.

What does one want to do next, once the monitoring is in place? You should have proper alerts set! The alerts should be proper in place like alerts for the event when the CPU utilization of servers goes above 80/90% or when the application stops responding. This is where monitoring comes handy and is crucial for any project.

7. Team / Organizational Culture

You should take out sometime and identify the right stakeholders at the start of any project. Wouldn't it be great if the decision makers have all insights of the project, code delivery process, the application performance, and are a part of regular meetings? They would actually know all the bottlenecks and even give direction to some ideas.

Stakeholders need to review the design documents, participate in product demonstrations, give inputs during retrospectives, gather additional requirements and discuss them. They should try to include the customers in some of the meetings to keep a check on the project goals. Also, stakeholders may arrange for various trainings for the members so that they can give better inputs and make the project a success. Using some tools that collaborate such activities may be another boon and may save time as well. For instance, analytics reports could be shared with

sales team, which they can use to get insights about what the users are frequently seeing on their website.

8. Data-driven Approach

As one of the good DevOps practices, you should focus on performance. Factual data is what you want to rely on. DevOps is all about continuous improvements in performance and if the application is not performing well, then the development is not yet over. Good performing apps is the need of the hour, right?

You can bring together various aspects like graphs, usage patterns etc. and share it with the team members, which will help everyone think in the same way. Also, inclusion of scalability, testability and deployability to agile task will help ease the process.

The concept of DevOps requires IT development and operational teams working together to build and improve application performance and usefulness, using a cohesive, repeatable, continuously refined approach. The combined focus on throughput and operability is the key to high IT performance via DevOps.

Building A Data Driven Culture

Bottlenecks in operations or development process actually slows down the pace of the organization; as a result the productivity goes down. It is very important to understand what all things are affecting the deployment process as the code is in the pipeline from QA to production environment. The best way to achieve this is logging. The data from timesheets & tracker can also help in finding the most time taking areas in the process that are responsible for the slowdown.

With DevOps mindset, you can also identify the bottlenecks in other departments. The blockers and time taking tasks can be reviewed and automated to improve overall productivity.

9. Centralised Log Management

Logs are a vital part of an application. Not just the error logs, but normal logs of an application are equally important. You need to have a proper strategy for maintaining logs. Logs need to be well structured and easy to view and interpret. Logging done end-to-end is also beneficial, helps to see the performance of the application from the point of view of an end-user.

Tools like Log-entries can be a boon if properly used. Centralised logs for the application having more than 3 servers will increase the pace of debugging in case of any outage. Alerts in the centralised logs can prevent a huge outage. Logs can be viewed as:

Graphs | Tables | Pie Charts

It should be made sure that the practices like log rotation are enabled, as the team doesn't want to end up using all the disk space. Logs are helpful but too many can be a pain to handle. Log rotation is a much recommended practice to follow.

Innotas

Innotas is a cloud-based project portfolio and application portfolio management solution that delivers a seamless way to manage projects, resources, applications, and budgets across all of IT.

Problems faced by Innotas

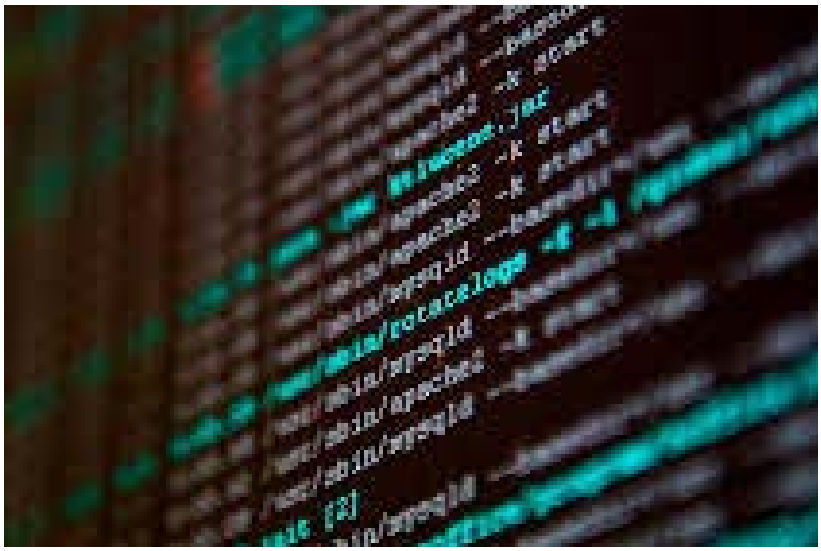
- Quickly debug issues while restricting access to production environments
- Monitor customer activity and address concerns in real-time
- Correlate log events with related databases

Solutions

After considering ELK and Papertrail, Innotas chose Logentries for AWS/Apache infrastructure monitoring and application debugging. With a tool called Logentries, the following are possible.

- Monitoring and debugging customer issues in real-time
- Tracking customer email volumes
- Analyzing and improving the performance of individual customer integration

A similar approach could be followed.



10. Centralized Dashboards

Another good DevOps practice is having automated dashboards. Wouldn't it be great if you could view all the configuration changes to the servers, databases and deployments that have taken place? These are nothing but visual dashboards of various metrics, logs etc. Just like monitoring, dashboards bring these metrics & graphs together. A tool like Hygieia Dashboard integrates with VersionOne, Jira, Subversion, Github, Hudson/Jenkins, Sonar, HP Fortify, Cucumber/Selenium and IBM Urbancode Deploy. This might get complicated, keeping it simple is in your hands by keeping the important things in the dashboards.

A brief summary of all DevOps Practices

- An organization needs to take care of the performance reviews of all the team members that are deployed on a particular project.

- Point out the most time taking areas that are slowing down the pace of the organization

- The real time project visibility helps the organization to forecast the delivery dates ensuring that the quality and customer satisfaction remains the first priority

- The use of automation should be maximized in order to have an efficient DevOps workflow. The automation tools should be compatible with each other so that issues can be minimized or removed.

- Always start with the smaller assignments when adopting new culture into the organization. This helps the organization to build appropriate confidence for managing bigger assignments/ projects and provides shelter from losing the customer due to lack of expertise.

- **"Never forget the users"**. The intent of every step should be to ensure that the end customer should not be affected by the changes.

- Changes should be done in a collaborative manner to avoid any situation of ambiguity amongst the users and intended audience.

Continuous improvement in the product always helps the organization to attract more and more customers leading to an improvement in the reputation of the organization in the global market.

Always take feedbacks and opinions from all the stakeholders associated with the product. It could be customers, technical teams, top level persons, sales team, finance team etc.

4. Seven Principles of DevOps

1. Culture of Continual Experimentation

Continuous experimentation leads to innovation and improvement of the processes. Risk taking processes allow people engaged in various activities of the software lifecycle to recover more quickly from the failures that can't be predicted. Accepting failures, their correction and optimization help the organization develop more robust software solutions. This kind of attitude helps the organization to perform better than their competitors in the market. Risk taking basically extracts the hidden phases of the software system which in the end, can lead to huge dissatisfaction from the customer, if left untouched. Experimentation always leads to new things.

2. Learning Attitude: Open to Failures

Learning from failures not only improves the checklist to ensure the robustness of the software system but also helps the team in gaining ample amount of confidence in themselves which is a positive sign for every organization, be it a small MSME or a large MNC. A learning attitude helps to choose the right tool/ practice to carry out the processes. If there are 5 ways of doing a task, after trying all the five tasks, you can choose the best one or make your own that can be a combination of one or more of the ways.

For e.g., if someone wants to choose between Chef and Puppet, then he should be implementing them on a single server with all the possible use cases and then deciding the best choice for the project or the organization. By adopting this learning attitude, you will end up configuring a better tool for

the whole infrastructure which is very important for the product and the organization. Fail fast and learn fast should be the key point while adopting DevOps culture.

3. Iterative

As DevOps practice is based on agile methodology, you can never be sure that the pre-planning will always work. You have to mould your plan according to the feedbacks and the solutions needed to remove the bottlenecks of any project. This principle lays down the focus on making the team flexible to adopt new changes and welcome them with a positive attitude with an intention of improving the quality of the software system. Iterative nature of feedbacks, surveys and their resolution steps make the whole system efficient.

4. Continuous

The word "continuous" works in many ways in DevOps. Continuous delivery is the way in which the delivery process resulting from new repository changes starts flowing from staging to any environment, or even production.

Continuous deployment is the process to automate the deployment process from staging to production environment. It is an optional process that changes with different organizations.

5. Collaborative

Every member should have common objectives and goals regarding their deliverables. The trust factor is equally important in the DevOps practice. The main focus is to remove any dependency on a particular team member. There is no need to realign the whole organizational structure for this collaboration but a common mission is a must. Transparent communication can be one of the major points to increase collaboration among the team members.

Silos are basically the boundaries between different teams that often lead to problems and outages. The various secluded divisions like QA, development and operations may create confusion between its team members.

The culture of saying "Well, I am done with my part, now next is yours" doesn't really work in DevOps culture.

DevOps is a common name for all the team members. The old model is now outdated. In DevOps, there is a mixed level of team, expertise and specialists but with no silos. Using right tools, right approach and management in a proper adoption of DevOps can be a huge success. There is a need to change the mindset of team members to adopt the culture of DevOps. There is a need to own the whole process instead of owning just the area of expertise.

6. Systematic

DevOps approach mainly focuses on system as a whole. DevOps looks at the impact of changing organizational structure, processes, technology to evict as many constraints as possible. The main focus is on the value to improve the IT business so that optimum level of the services can be delivered to customers. The improvements should be done at the system level in an appropriate way and not at the local level, so that at the delivery time, there won't be any issue that ends up as a difficulty to the customer in using the services. DevOps practice needs to analyse the problem at the system level and should come up with the best solution that may lead to system level changes.

7. Automated

The purpose of bringing automation in any scenario is to eliminate repetitive and exhaustive tasks being done manually. Automation is a technical aspect of DevOps culture and is one of the key principles. DevOps focusses on automation in terms of development, testing, operations and security to deliver cost-effective products at a faster pace. The intent of automation is to prevent the team from doing repetitive works again and again. To achieve this speed, there should be least human intervention and tasks should be rather done automatically with the help of tools.

5. Common misconceptions about DevOps

It is not NoOps. It is not just about Developers handling the tasks of Ops team

It is not all about tools. DevOps is more of a thought, a thinking which drives the procedures in a project. Tools just act as drivers.

It is not just the culture - It is a practice that needs to be followed for better delivery

It is not about just Devs and Ops. What about Network admins? Designers? - It is an idea of collaboration where everyone works towards a common goal without any silos.

It is not just a job title - Consider all the principles of DevOps, which require regular changes at system level. You can't bear the fruits without that.

It is not everything - DevOps deals at system level but if you goes deeper, DevOps could be found improving Operations work with the help of other things

DevOps is still emerging and is in the growth phase. It may not be at its best yet and might have some flaws. For implementing DevOps, an organization should consider all its pros and cons. What suits your organization may prove to be disastrous for another. DevOps should be made a practice.

TO THE NEW has adopted DevOps and has seen more benefits than any other organization-wide practices. DevOps is definitely a game changer in today's market.

6. About Us

TO THE NEW is a digital technology company that builds disruptive products and transforms businesses. We leverage the power of experience design, cutting-edge engineering, cloud and analytics led marketing to enable digital transformation.

Our passionate team of 750+ people includes passionate technologists, digital analytics experts, video specialists and creative mavericks who have transformed businesses of more than 300 companies spread across 30 countries worldwide. We take pride in our culture which is driven by passion for making an impact through technology.

Top Devops Practices and Principles to Improve IT Efficiency

LET'S CONNECT

www.tothenew.com

info@tothenew.com