TIPS FOR DEVOPS ORGANIZATION STRUCTURE

[DevOps](#) is not a silver bullet that will make all your problems go away. Nor is it a system you can just turn on like a light switch. DevOps helps to optimize your IT organization's process for software development along each step of the development chain. Introducing DevOps systems into your organization isn't something that will happen overnight but leveraging the advantages that DevOps can provide is well worth the investment required to implement its solutions.

Before we go into more detail on the implementation of DevOps into your organization, let's briefly go over what DevOps is so we can be on the same page.

The Basics of DevOps

Much like Agile, DevOps is as much a mentality as it is a system. It's a mentality of communication and cooperation that should be integrated into your entire organization. DevOps also requires support and reinforcement by management to perform at its best.

DevOps is the confluence of development and operations but is more than the sum of its parts. Specifically, DevOps is a system for software development that focuses on creating an ongoing feedback loop of analyzing, building and testing while leveraging automation to speed up the entire process. To achieve this kind of seamless and constant loop of software building and testing, you need to create teams of cross-functional disciplines that work in concert.

Outputs of a strong DevOps system are ideally modular and containerized to allow for rapid deployment. Creating modular deliverables ensures that each piece can function on its own and

issues won't impact the entire project negatively. Containerization allows for easier implementation into current features as well as simpler rollbacks. Keeping each deliverable to a smaller, more manageable size helps to maintain the quality of work while accelerating the speed at which changes can be made.

The key to DevOps functioning at optimal levels is engendering a culture of communication where teams can coordinate among themselves and with other teams effortlessly.

The Importance of Communication in DevOps

DevOps requires individuals from various backgrounds to band together as a team working on a singular goal. This kind of collaboration has been avoided in the past which created communication silos where each discipline works in their own bubble and then hands off their work to the next discipline in the development chain. Siloing creates bottlenecks and makes it easy for communication to get lost in translation.

Infusing your organization with a culture of communication enables your teams to have a deeper insight into the pain points each discipline can experience. This understanding makes it easier for each team member to see the whole picture which leads to shorter development times with fewer errors. Rapid deployment allows for more experimentation that will see your product constantly improving upon itself through iteration.

DevOps and Agile roles are important aspects within each team to help ensure members own the process as well as their contributions to the projects. Using rotating roles will also help team members to better understand the entire process so they can make informed decisions regarding process changes in the future. This understanding also serves the purpose of improving lines of communication through shared knowledge and experience.

Analyze Your Organization's Bottlenecks

One of the main pursuits of DevOps is the automation of processes, but it's important to focus on where your processes can most be improved through the use of automation. Find tasks that are done often enough to warrant automation but avoid trying to automate everything for the sake of it. An analysis of your organization's bottlenecks will provide information on some good places to start applying automation that will help speed up production.

To analyze your bottlenecks, pay attention to the root causes of these bottlenecks. They can generally be attributed to three primary causes: staff, systems, and management. Staffing issues can be mismanaged resources or a lack of adequate training. Systems failures are caused by outdated systems or incompatibilities with the rest of your stack that slows down the whole process. Management issues refer to the poor direction or a lack of clearly outlined goals.

This means your bottlenecks might not necessarily be caused by something that can be repaired through automation. A detailed analysis of your processes will help to identify which issues can be fixed by applying DevOps practices and which ones require different approaches.

Hold Regular Retrospective Meetings

Retrospectives give time for team members to talk about what happened in the past couple of weeks and what they felt went right and what didn't work for them. This allows teams to agree on

processes they will employ over the coming weeks without creating too much friction because they know the processes can be modified if they end up not working in everyone's best interest. A system like this allows teams to be more productive through the use of experimentation instead of wasting too much time on theorizing.

Meetings like these keep the team on the same page and give everyone a chance to communicate their thoughts on how things are going. The feedback loop encourages those who thrive upon it to find ways to improve. Retrospectives also provide valuable data concerning the success of the process and its approval rating from the team members utilizing it. All this information can be used to inform future decisions and increase the effectiveness of the system as a whole.

Remain Fluid in Your Processes

Don't adhere to any prescriptive methodologies if you find they aren't working for your organization. No two organizations are alike and that means taking someone else's system wholesale and applying it to your team isn't a guarantee of success. The beauty of DevOps and Agile is that they encourage experimentation and enable rapid changes to be made. Take advantage of this expectation of DevOps and make sure to embrace new ideas at least for a short testing period to see what works best for you.

Remember that DevOps is something you are leveraging to improve your organization's processes and products so if the solutions you're using aren't working for your company - change them. There's no sense in being married to a system that isn't aiding you. Just be sure that you understand what is and what isn't working as well as why. When in doubt, try something new but make sure you track the changes and analyze the results.

Putting It All Together

Starting your DevOps transformation will require diligence, but the payoffs of a well-managed system will be more than worth the efforts. Forming cross-functional teams that integrate each discipline of the production chain (dev, testing, and ops) will require special attention for creating solid lines of communication. By engendering a culture of communication throughout your organization, you will empower collaboration within teams and between them that will improve development speed and product quality.

The understanding each team member brings from their discipline will reduce the need for handoffs and will make sure problems are found sooner or prevented altogether. Rebuilding your service delivery cycle with DevOps automation in mind will increase the pace of development while enhanced communication improves the quality of each output thanks to team members bringing their own perspectives to the table.

Finding the pain points and bottlenecks in your organization and identifying their causes will give your DevOps teams a focus towards which they can direct their efforts. Finding opportunities where automation can speed up production and reduce confusion will vastly increase productivity across your entire organization. While identifying opportunities, make sure you don't go overboard and try to automate processes that you will spend more time automating than the time you would save from that automation.

Hold regular retrospective meetings within each team that gives members an opportunity to express their opinion on how things have gone and how they would like to do things moving into the

future. Take advantage of the fluid nature of DevOps and Agile by encouraging experimentation and embracing a fast rate of change. Avoid becoming married to set systems and protocol because not every solution will work for your teams or your organization.

DevOps is a powerful system for IT organizations to expedite their processes while improving the quality of their outputs. Although you'll experience growing pains during your transformation, keep open lines of communication, keep everyone on the same page, and ensure they understand the reasons for the changes.