

regular daily operations and the changes necessary to resolve those issues. DevOps organizations in particular thrive within a culture that best enables collaboration among the traditionally disparate departments of Devs, Ops and QA, among others. DevOps process workflows and tooling that focus on automation portray one part of the story. The other key element is the synchronization between all entities responsible for following those workflows and using the automation tools to perform product development tasks. Digital communication channels and project management technology that support [SDLC](#) workflows can be crucial in achieving synchronized and automated collaboration among all stakeholders of a DevOps project.

2. Keep up the Motivation

Since the modern-day business centers around software-enabled product offerings, organizations must ensure that IT is aligned with the business goals of the company. Motivation toward the intended business goal and ownership of the business impact of pushing well-developed products into the market ensure that all stakeholders are on the same page. In order to achieve this mindset, organizations must engage employees who are best fit for the intended projects and incentivize them well to justify the extra efforts and interest. Secondly, team building and teamwork should be planned strategically to ensure that every member of the team can make their best contributions on an individual as well as a collective basis. Thirdly, managers and executives should support the team to identify the bottlenecks and problems that reduce team motivation and hamper productivity.

3. Establish Cross-Functional Roles

One of the reasons for silos in IT is the presence of a limited and concentrated skillset within disparate teams and departments. With traditional SDLC methodologies, the development, testing and operations work was treated as separate and independent assignments. Naturally, the teams were developed in a way that cross-departmental training was neither required nor encouraged. In the DevOps world, this practice is intended to transform dramatically as, for instance, development and testing tasks are performed simultaneously earlier within the SDLC pipeline. A developer must assume roles of the operations and testing personnel as needed. However, that is not to say that all skills and knowledge necessary will be directly available to within DevOps teams, which is why organizations must facilitate on-demand collaboration and streamline communication between disparate IT departments.

4. Leverage the Tools and Processes

Modern software development practices according to Agile and DevOps principles focus on culture and collaboration as a key component of an effective SDLC strategy. The tooling and process frameworks are considered as obvious DevOps imperatives. Practices such as continuous integration, testing, delivery and deployment should be employed to address IT silos and process bottlenecks. When the DevOps framework compels developers and operations to work collectively early and continuously during the SDLC lifecycle, team members will be more likely to collaborate in achieving their collective DevOps objectives. For instance, there are no walls or barriers between developers and testers when the DevOps organization is following the philosophy of “you build it, you run it” as opposed to completing a software build ready for testing and responding with the “it’s not my job” philosophy when the tests are failed.

Automation, and not just tools to automate tasks, is necessary to fulfil these DevOps principles. For

instance, Devs, Ops and QA personnel are equally engaged toward continuous delivery practices. Automation tools for collaboration, resource provisioning, infrastructure as code and automated testing are critical to meet the collective goals of these individuals in realizing a continuous delivery pipeline.

Since IT heavily relies on tooling and technology capabilities such as SaaS solutions and cloud infrastructure, organizations must also eliminate the possibility of vendor lock-in. For instance, a complex technology that requires specific skillset only available to a few members, or a technology stack that creates data silos may prevent convenient flow of information between teams and IT departments. In this situation, the workforce is less likely to collaborate or follow true Agile and DevOps principles.

5. Measure and Feedback

IT silos often occur when an existing strategy fails to account for unforeseen or unexpected circumstances in the way IT organizations operate. The effectiveness of the appropriate organizational strategies, IT process workflows, collaboration tools and team building should therefore be measured and evaluated frequently. The feedback loops should also extend beyond the organization and evaluate the end-user or customer experiences associated with team building and product development strategies. For instance, a critical security issue or performance bug that remains unpatched and affects end-user experience is a sign that appropriate IT teams are not collaborating to resolve the issue, or reduce its chances of occurrence in the first place.

The process of reducing silos within the organization is an ongoing process and may require continuous improvements in terms of the technology, processes as well as the way that IT organizations operate. For successful DevOps adoption, it is critical to identify and address the silos before the impact escalates and encompasses the project lifecycle and the wider IT organization.