

5 Steps to More Effective QA Automation in a DevOps Environment

Questions and Answers

February 25, 2016

Q: How do you implement DevOps in a large, cumbersome enterprise-grade environment, like SAP?

A: In working with our customers, we advocate starting out small and perfecting and refining the process before rolling it out in a broad scale effort like SAP. Often people will begin with a project, an application, or a group/team to enhance the process and develop a solid methodology to map it out. If you have a team that is involved in making small changes within SAP, this may be an area where you can certainly begin the process. Ultimately, start small, refine the process, and scale from there.

Q: What 1 or 2 things do you think are key for implementing DevOps culture in an organization?

A: One of the key aspects in an effective DevOps strategy is collaboration. This begins with knocking down any silos that exist traditionally, and finding ways to facilitate communication between the BAs, Developers, Testers, and Operations teams. By sharing information continuously, you are able to eliminate many unnecessary surprises and give everyone visibility into what's happening throughout the process.

Secondly, it is important to make sure everyone is on board from the start. This means full commitment to the end result. It's no longer acceptable that the developer's job is done when they hand the software off to QA; everyone is invested in the quality of the final product. This requires a level of executive support from the very beginning. The goal of better/faster/cheaper is hard to achieve within a first iteration; there is some process refinement that needs to take place. Having executive support to make these changes and gain everyone's buy-in is a critical success factor in making DevOps work.

Finally, once you have this, it is important to hold people accountable for the quality they deliver.

For those writing the requirements, are the requirements testable?

For developers, does the software work from a unit perspective?

Testing teams, can they ensure the whole application is working properly from end-to-end?

And lastly, the operations team, are they equipped to fully support for the new application?

Q: What are some good ways to tear down silos between Development, QA and Operations that you have seen successfully implemented in practice?

A: It is important to have executive management support to ensure silos are broken down, communicating from the top down to all levels that collaboration is key, and everyone will be held accountable for the quality and timely delivery of the application. Additionally, and probably most importantly, companies need to involve their Development, QA and Operations teams from the beginning of the project so that they all have a say in the design, requirements, and metrics to ensure how quality is ultimately delivered. Putting in place a project "team" made up of representatives from all

of these groups that work together on the entire project delivery should tear down these silos very quickly.

Q: With outsourcing, developers can be literally worldwide. Any strategies on implementing DevOps when the team members may be in different countries/time zones?

A: This situation is very common; often, even non-outsourced teams seem to be dispersed. To successfully manage this, it comes down to a few critical factors: commitment from the team, a common platform that facilitates vs. inhibits communication, and having the right ALM structure in place that provides the same data, results, information to all users.

In addition, giving all those involved in the process, even outsourcing teams, a feeling of comradery and inclusion, helps build collaboration and commitment. In one situation for example, a movie studio employed a consulting firm that used outsourcing resources for test automation. The movie maker helped the outsourcing team understand the company objectives from the very start, and as they moved to project completion, the team was given an opportunity to see the final movie before it was released to the public. This helped the outsourcing firm feel like they were part of the actual company they are working for, and gave them a deep understanding of their role in delivering a quality, successful project.

Q: Am I correct that there is big upfront time commitment to build the components and that the time savings will be down the line due to reuse?

A: Yes, like any technology investment, there is some time and commitment upfront, but you reap the benefits of the automation down the road. As you go through multiple iterations, where you are reusing the test components and test cases that you have built, you gain efficiencies in the process, shorten the time to test and minimize risk to the application.

This is what's nice about the TurnKey cFactory tool; it is easy to maintain. With a simple click, you're able to update your components and these changes are pushed out to all the test cases that use it. Putting the upfront time and effort in building an automation framework will reap huge benefits down the road.

In addition, this is why it is important to have a clear purpose and clear objectives. Statistics indicate that the cost of quality when not integrating QA into a DevOps initiative increases rather decreases from 18% to roughly 30%. There are huge gains to be had. It all comes down to keeping an eye on quality. With a quality focus, this is where you can drive a ton of your savings. The upfront investment is less in tools than in more people and processes and figuring how to make it work in your environment.

Q: You specifically mention functional testing. Do you include performance testing as part of DevOps?

A: There are a wide variety of various testing disciplines, and this is one of the traps for people who are entering into DevOps, to trivialize testing. If you think about all the various forms of testing—functional testing, performance testing, unit testing—there are

so many different things that have to be or should be done in order to ensure quality. All the various types of testing are required to have a high performing, quality application and should be included in the process.

Q: TDD, BDD vs data driven testing?

A: TDD/BDD are used for unit testing or development testing type approach in a more iterative, continuous development model. Where we are using data-driven testing is tailored more towards full regression and end-to-end testing across your environment. Again, these are different forms of testing that all need to be including throughout the process.

Q: Data driving isn't always the best automation approach. Does TurnKey only help when data driving tests?

A: Data-driven testing offers the greatest value in functional end-to-end business process validation and regression testing. This is the key area of testing that TurnKey supports. Its goal is to ensure quality delivery in an integration or user acceptance type environment.

Q: Who decides about "what is meaningful to be tested"?

A: In an ideal world, it should be your QA infused DevOps team. Everyone from the BAs, who decide what is business critical. Development, who has insight into what software was the more difficult code to build and could potentially have most failures or impact. Testers, with their testing expertise, may also have the best feel for where applications tend to fall down. And, Operations, who support the applications from day-to-day have a sense for what issues they are currently seeing. Ideally, you want input from everyone. This gives you unique vantage points from each of the key areas, but the final decision should, ultimately, be with the Business Users/Business Analysts that are designing and implementing the new system.