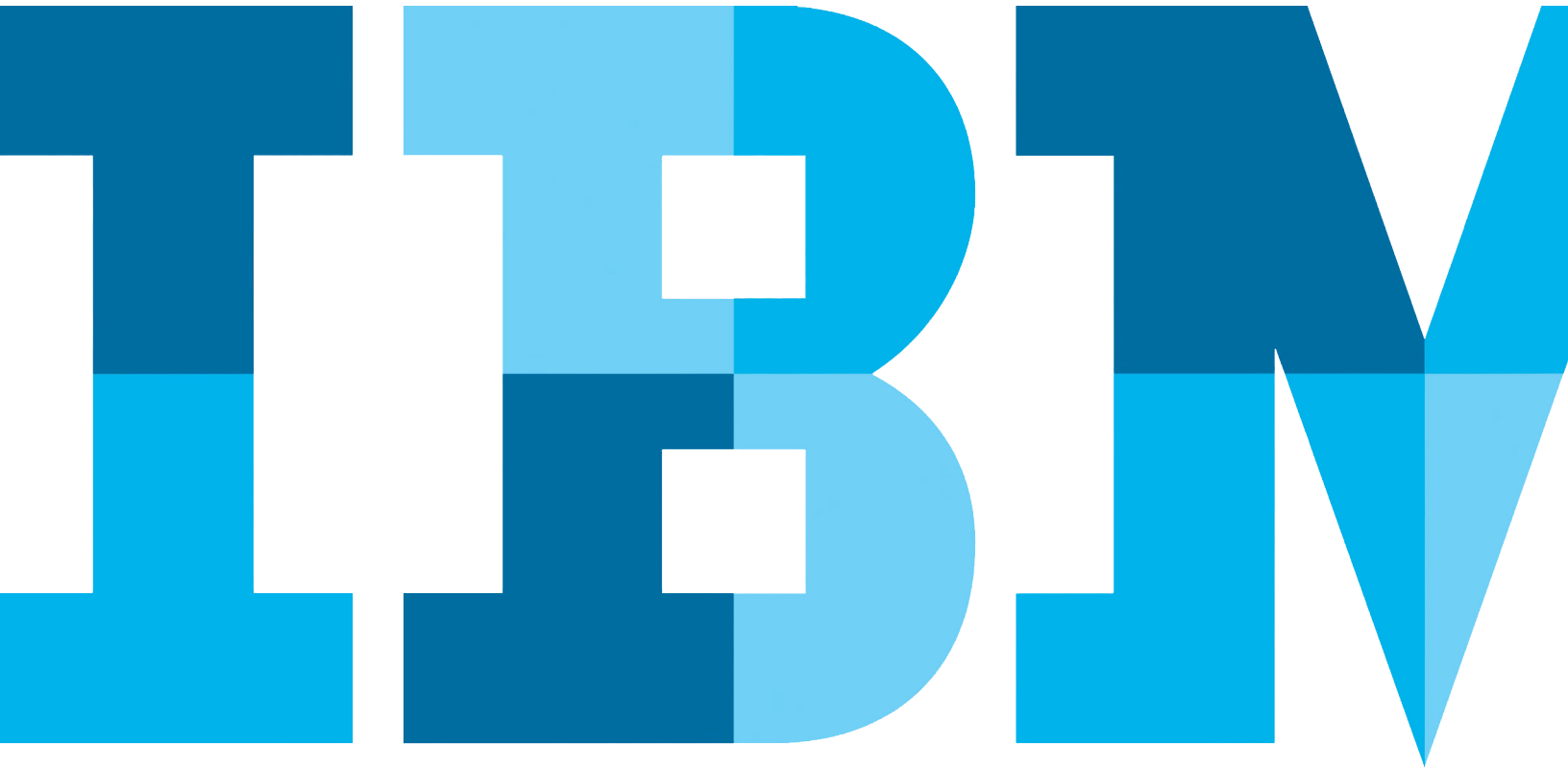
IBM Systems and Technology
Thought Leadership Article

June 2014

Bridging the Testing Gap in Continuous Delivery

Improving the software delivery process by spotting and remediating defects before the software is released.



IBM

Leading organizations are discovering that reducing the amount of time required to release new software can help them get to market or build market share faster, as builds from one's continuous integration process are continuously deployed across the many stages into production. While mobile and software are driving new business, the real problem is less time to deliver and meet end-user/consumer demands—which results in less time to do thorough testing.

In order to create a better software delivery process, organizations need to consider managing risk by having an awareness of the defects before the software is released. Continuous integration can accelerate development speed, while service virtualization can facilitate the early testing required for delivering high-quality software at this faster pace. This makes it possible to test new business services and incorporate feedback into newer releases earlier and continuously.

Enter Continuous Delivery

Continuous delivery is a set of practices for automating and improving the process of software delivery. It leverages techniques such as automated testing, continuous integration and continuous deployment to release high-quality code with minimal manual overhead. This makes it possible to push out enhancements quickly.

Quality gates are used to ensure that code that does not meet specific criteria is not automatically deployed to the next test stage. This kind of agile process also requires the ability to automate the rollback of changes and return the environment to a known good state if a catastrophic defect is discovered that cannot be quickly fixed.

The goal is to get to a point where continuous testing is the checking/testing of the software, and continuous deployment across development, test and production environments is the validation of the delivery process itself. So while you are continuously testing the code, you are continuously validating the process by which that code is delivered. But continuous delivery is not just about releasing software faster, as defective software might end up costing more and providing less benefit.

Removing Testing Bottlenecks with Service Virtualization

Unfortunately, many of the services required for testing are not readily available during development. To address this gap, service virtualization uses a variety of techniques to model existing or planned services. These virtual services enable teams to stand up realistic and complete test labs simulating missing yet dependent software or services faster.

As opposed to writing ad hoc mocks and stubs, service virtualization allows an enterprise to create realistic simulations that can be used across the organization. Developers can use it to do integration testing as they are writing new code. This allows defects to be discovered earlier and fixed faster, at less cost. The ability to create realistic software models with service virtualization brings the concept of "write once/use by many" to the testing environment.

For example, a large European bank with 30 million customers and about 3,500 branches found tremendous productivity and quality gains when it incorporated service virtualization into its delivery lifecycle.

https://www.ibm.com/developerworks/community/blogs/rqtm/entry/forrester_research_total_economic_impact_study_on_service_virtualization_and_test_automation_solutions

Using service virtualization as part of its continuous delivery pipeline allowed the bank to double the number of new software projects released, from 40 to 80. The bank was able to achieve this without additional staffing, since the development team spent less time fixing the code that had gone into production.

Testing earlier in development also reduced the number of service-impacting production incidents from 2.5 to 0.3 per project after implementing service virtualization. These incidents resulted in application outages that cost an average of $100,000. The cumulative return on investment from incorporating service virtualization into a continuous delivery process was 1,333% over three years.

The reduction in incidents was attributed to greater scope, sophistication and frequency of integration testing. Service virtualization also allowed the bank to reduce the time required to perform system integration testing from three weeks to half a day.

Testing as a Forethought

The notion of "shift left" is about conducting better testing earlier in the development cycle. In the past, the biggest challenge was that developers and QA personnel had limited access to the infrastructure required to do testing. They would write ad hoc mocks and stubs to test basic functions, but they might defer testing or test only the software that was available. As the code moved along the process, they were delaying the inevitable and finding defects later in the process. In some cases, this means not just rewriting a few lines of code but also revisiting the application architecture or framework. Service virtualization combined with more complete testing early on allows teams to avoid these kinds of issues.

By making it easier to test throughout the development process, developers can find bugs when they are cheaper to fix. Early and continuous automated testing at the API level and automating the development of realistic test labs frees up testers to do other added-value activities that are harder to automate, like exploratory testing.

Test automation allows quality assurance teams to incorporate their knowledge about problems that can occur in the enterprise's deployment environment. Once they have been set up, more basic testing can be done by the development team, which can free QA and development to test further.

In addition, if the operations group has visibility into the level of quality earlier in the process and understands that these architectures have been vetted earlier, it will be able to push out new releases with a higher level of confidence.

For more information

Are you ready to shift left and take that next step into a new DevOps era of Continuous Testing and Service Virtualization?

Learn more:

1. Visit the IBM Service Virtualization page on IBM.com to learn more about how IBM can help: **http://ibm.co/servicevirtualization**

2. Watch the latest webinar about IBM's DevOps approach to Continuous Testing and Continuous Delivery: **http://bit.ly/shiftleft**

3. Join the IBM DevOps conversation on Twitter at **#ibmdevops**

IBM Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
June 2014

Please Recycle

RAL14098-USEN-00