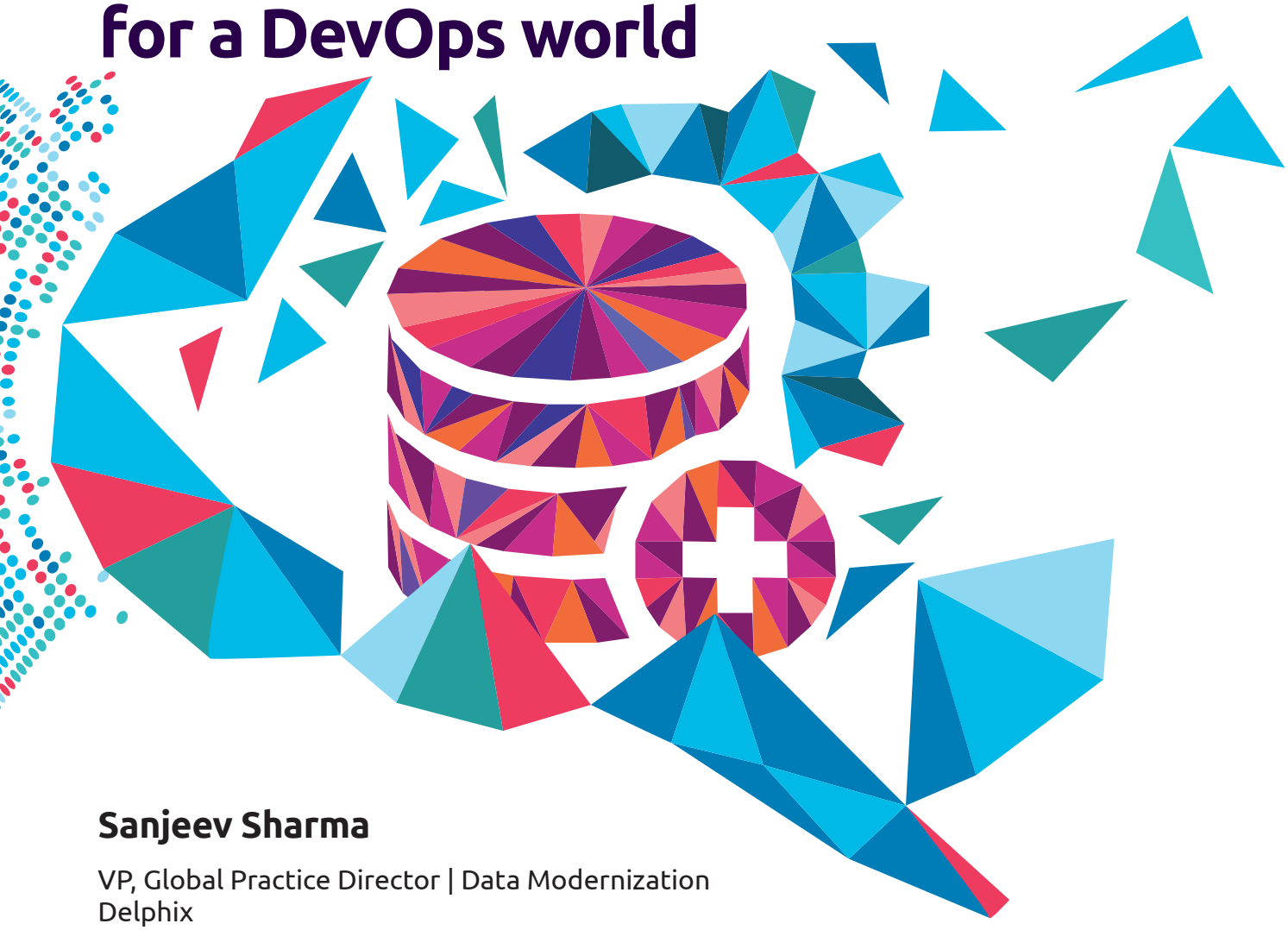
Key trends in Quality Assurance

Modern testing approaches for a DevOps world



Sanjeev Sharma

VP, Global Practice Director | Data Modernization
Delphix

Agile and DevOps are two software development methodologies that have become the standard for building continuously deployed systems at speed with quality. As large enterprises with complex interconnected systems look to scale their DevOps adoption, there is growing pressure on test and QA teams to streamline their processes, tools, and organisational structures to support the rapidly accelerating need for Continuous Testing. The complexity is further exacerbated by the 'hybrid' nature of the technology stacks in most enterprises from mainframes that run both modern container-hosted applications and legacy COBOL-based monolithic applications, to distributed systems with myriad technology stacks, workloads distributed across multiple public and private Clouds, and SaaS properties scattered across multiple vendors–each with their own proprietary ecosystems.

Although it was promulgated by startups that were early adopters of DevOps, "the notion of Move Fast and Break Things" is passé today. It was a Silicon Valley thing, and that era no longer exists. Enterprises require both speed and high quality, and the need to deliver products and services faster, while maintaining the high expectations of quality and performance are challenges modern day testing and QA teams must address.

Shift left test

One of the common approaches leveraged by test and QA teams to scale their capabilities is shift-left testing. Shift-Left, a core tenet of DevOps, is the ability to execute earlier (left) in the application delivery pipeline. In the area of testing, this means enabling the developer to self-serve tests – including unit tests, functional tests, integration tests, and even performance tests – and allowing developers to run them early and often. Testers still maintain, and in fact, enhance their core competencies, in addition to focusing on designing and running tests that leverage techniques such as test-driven development (TDD).

Democratising test data

A prerequisite to shift-left of test and QA capabilities is the democratisation of test data. In order for developers to run tests on demand, they need access to the right test data on demand. To enable this requires completely transforming how test data management (TDM) is done.

So how does test data management fit in with DevOps? DevOps requires developers and testers to deploy applications on a continuous basis in order to validate their integrations, functionality, and performance. This implies fresh sets of test data are needed each time developers deploy a new version of the application, but providing good sets of test data is inherently challenging. To automate and scale test data management, it is necessary to eliminate the need to manually create, clone, provision, and maintain test data. In addition, the ability to execute these actions in an automated fashion must be available via APIs to developers, who can then integrate them into their DevOps toolchains.

The following are ways to help modernise test data management for DevOps:

- Automate the creation of production-like test data. Organisations are typically creating test data by manually cloning their entire production system or creating synthetic data, rather than leveraging data virtualisation technologies. In fact, nearly half (47%) of global enterprises say it takes four to five days to provision a new data environment, according to a global research and advisory firm. By leveraging data virtualisation, automated test data provisioning can allow for rapid creation of test datasets for various types of testing on demand.
- Mask sensitive information for compliance and protection. Protecting data privacy is no longer optional—it is the law. Organisations must have procedures in place to de-identify data across non-production environments to comply with data privacy regulations and avoid data breaches. Data masking provides development teams with meaningful test data without exposing sensitive private information, such as personally identifiable information (PII) and protected health information (PHI), by replacing the original value with a fictitious but realistic equivalent
- Refresh test data for continuous delivery. To enable continuous delivery, testers and developers need access to test data continuously, so teams can run tests each time a new version of the application is delivered and repeatedly run them for the next iteration. For that reason, organisations need to streamline test data delivery by enabling testers and developers with automated tools and processes to refresh data without involving DBAs.

Leveraging AI in automated testing

The next evolution in the area of automated testing for DevOps will be around leveraging Artificial Intelligence (AI) and Machine Learning (ML). As organisations make the move to containerised microservices, the frequency of deployments will increase exponentially, driving up the need for more testing. Microservices-based architectures also require testing to be performed more intelligently by driving synthetic transactions through sets of services that are orchestrated together to deliver the business functions needed to execute the transaction.

In parallel, new approaches, such as chaos engineering (which is based on the premise that servers in production will go down), will go mainstream. It suggests that things will go wrong, but the services being delivered should not go down. The goal of such techniques is to ensure that the services being deployed are resilient in a chaotic, real-world environment. All of these efforts will drive the need for AI and ML-driven testing, meaning testing and QA are guided by learning from the data generated by the tests being run, by the performance of systems in production, and by introducing randomness — chaos — into systems under test.