WHAT'S NEXT IN THE AGE OF SOFTWARE

Overview: As organizations adopt Agile and DevOps, they must also foster innovation and peak performance in their development teams by encouraging modern, collaborative work methods and processes. Included in this effort are the adoption of automated testing and shift-left methodology, as well as granular evaluation of the delivery process, backed by measurement and analysis of core KPIs.

In the Age of Software, competitive advantage—or disadvantage—is determined by the velocity, quality and efficiency with which organizations can continuously turn digital ideas that matter into digital experiences that customers care about. Large enterprises that used to dominate their markets are today scrambling to compete against nimble digital disruptors who are flexed to respond to customers' always escalating expectations for more, better, faster.

Companies Will Move Toward Creating High-Performance Development Teams

In this new world order, where every company is a technology company, the role of the developer is appropriately changing for the better. These digital artisans are no longer order takers expected to bend to the will of the business. More and more—thanks in part to leaders who understand the immense value they bring to the business—developers are empowered to innovate on existing core systems, as well as deliver and support new means of digital engagement with customers.

But to do so, they require a milieu only afforded through Agile and DevOps, namely an open and

collaborative culture, inspiring and challenging projects, modern methods of working, and tools and processes that continuously improve their abilities. As agents of innovation, they must be coached like high-performance athletes based on KPIs of velocity, quality and efficiency to ensure their ongoing success.

Enterprises Will Place a Greater Focus on Automated Testing (And It's a Long Time Coming)

Enterprises are continuing to lose vital mainframe development and operations skills. Automating processes like testing helps to mitigate the effects of that lost knowledge.

However, unit and functional testing in the mainframe environment have traditionally been manual and time consuming for experienced developers and prohibitively difficult for inexperienced developers to the degree that they skip it all together.

According to an independent study commissioned by BMC AMI DevX, the vast majority of IT leaders believe that test automation is the single most important factor in accelerating innovation, but less than 10 percent of organizations automate tests on mainframe code. Arcane manual testing practices are creating a bottleneck that hinders the delivery of innovation and prevents organizations from meeting their business goals.

The good news is modern mainframe testing tools enable developers to automatically trigger tests, identify mainframe code quality trends, share test assets, create repeatable tests and enforce testing policies. Empowered with these capabilities, developers can confidently make changes to existing code knowing they can test the changes incrementally and immediately fix any problems that arise so they can deliver updates faster.

Development Organizations Will Experiment with Coupling Test Automation with a "Shift-Left" Approach

Businesses expect to achieve significant benefits by not only automating more testing on the mainframe, but also doing it at every stage of the development process.

To that end, as companies ramp up automation, they are also experimenting with coupling test automation with a "shift-left" approach—where developers write unit tests at the same time as they write source code. This enables teams to focus on quality as soon as a project is kicked off instead of waiting for defects to be surfaced later in the app dev lifecycle—defects that could disrupt operations, introduce security risks, hinder customer experiences or impact business revenues.

While a shift-left approach can help reduce the number of bugs that make their way into production, it can put more pressure on developers. That's why it's imperative developers have access to tools that enable them to automate the creation and execution of unit, functional, integration and regression testing on the mainframe, while empowering even novice developers to validate COBOL and PL/1 code changes with the same speed and confidence as they can with other code.

Automation coupled with a shift-left approach improves the quality, velocity and efficiency of mainframe software development and delivery.

Value Stream Methodology Will Be Increasingly Applied to Software Development

Value stream management (VSM) aims to connect an organization's business to its software delivery capability. VSM, together with Agile and DevOps, helps development teams focus on what matters most—providing greater value to customers, while reducing costs and boosting throughput.

More specifically, value stream mapping, the practice of granularly evaluating the end-to-end software delivery process, from ideation through product/service delivery, enables teams to identify and remediate friction points within their streams, so they can accelerate the pace of their innovation delivery.

Enterprises today should leverage VSM to continuously improve in four key areas: delivering new capabilities; resolving defects; reducing risk in security, privacy and compliance exposures; and removing constraints (e.g. technical debt) to improve the throughput of future deliverables. Neglecting any of these areas hinders the ability of development teams to deliver innovations that provide their business competitive advantage in the marketplace.

Enterprises Will Continuously Measure to Improve Software Delivery Quality, Velocity and Efficiency

Today, 57% of firms run more than half their mission-critical applications on the mainframe, and 72% of organizations say their customer-facing applications are completely or very reliant on mainframe processing, according to a 2018 Forrester Consulting study commissioned by BMC AMI DevX.

As organizations work to create high performance development teams to support accelerated mainframe application development and delivery, they need a way to continuously measure and improve mainframe DevOps processes and development outcomes. A program of KPIs is necessary for accomplishing this.

Measures such as mean time to resolution (MTTR), code coverage and number of defects trapped in test vs. production can provide a picture of developer efficiency and quality metrics. Machine learning can be leveraged to continually monitor and analyze behavior patterns enabling teams to continuously improve on essential measures. Enterprises that strategically leverage their data to tackle development and delivery constraints will see significant improvements at the individual, team and organizational levels.

Conclusion

In 2020 it's not enough for enterprises to adopt Agile and DevOps. They must recognize that their development teams hold the key to customer satisfaction. As such, developers must be treated like high-performance athletes and given challenging projects that are meaningful, modern methods of working, and tools and processes that continuously improve their abilities. Their behaviors must also be continuously measured so their performance will consistently improve, benefiting them as well as the business. To improve software delivery quality, velocity and efficiency, mainframe organizations must adopt more automation, especially automated testing, and test earlier and often in the DevOps lifecycle. And, they must take a granular look at their software delivery process to uncover bottlenecks and resolve friction points so ideas that matter can be turned into customer deliverables that make a difference—continuously.