Will mobile development bring mainframe development to the mainstream?

By Jason English

October 28, 2019

A large bank saw a 4X increase over the course of a year in customer requests and data interactions due to the adoption of mobile applications. Many customers started using the banking app several times a day and demanding integration with other popular payment applications.

To deliver on mobile, they needed to innovate on the mainframe.

"It took the mainframe and mobile teams a while to reach a shared test-first discipline, and streamline their automated pipeline with documented changes in the mobile app and back end, but once they nailed it down they were able to move very rapidly," said a project leader.

Banking. Retail. Travel. Insurance. In just about every major industry, mobile devices are becoming a common platform for work, and mobile apps are the de facto standard for how customers expect to interact with businesses.

It stands to reason that mobile is a primary focus of agile development efforts and budget at just about every company that sells a differentiated product or service. Front-end design, integration and support solutions cater to the rapid delivery of a superior mobile experience for customers.

But does this mean the mobile-first camp leaves the mainframe development camp behind, to toil away in maintenance mode rather than participating in innovation? Far from it!

The mainframe arena offers untapped innovation potential, and the mainframe development team has every reason to join the next agile standup meeting, help drive superior customer experiences, and deliver real value to mobile development.

Building the integration muscle of mainframes over time

Large, well-established companies have built their businesses on the mainframe for the last 40 years or more, and they're not about to walk away from such a center of accreted business logic now.

Much of the code of prior generations of mainframe development – think COBOL, C or Assembler on CICS – wasn't even object-oriented. The mainframe world must seem like a foreign environment to today's mobile development teams.

Source: https://intellyx.com/2019/10/28/will-mobile-development-bring-mainframe-development-to-the-mainstream/

In the early mainframe days, making modifications to business logic could be a herculean effort, so instead, IT teams got better at adapting their application code to talk to mainframes.

A bank trying to have a teller system call the mainframe for a balance lookup in the 1990s would likely need experts with paper printouts in hand, specifying exactly what CICS fields were needed for each query, and finding the most efficient way to make that request with the smallest possible payload for the core system.

Fortunately, over the years, enterprise IT teams have constantly adapted and built up new abilities to integrate to the mainframe, from message queues to WSDL-based web services, to native IDE support for developers to code and deploy atop mainframes in a familiar interface.

With the popularization of new open integration standards and component libraries, APIs, REST and JSON style approaches, innovation has really started heating up on the mainframe again. With open tools like Swagger, Connect and now the Zowe open mainframe project coming into fashion, it's easier to integrate mobile apps to the mainframe now, than at any point in history. So what is holding it back?

What's impeding the mainframe team in mobile?

Mobile development teams generally have a very agile release cycle – often delivering code several times per week, or per day in many cases. Since mobile interfaces, application logic, and back-end integrations are decoupled, significant changes can be pulled off in no time.

By contrast, major changes to core mainframe systems can take much longer. Mobile teams therefore fear that mainframes are fragile, or slow to integrate to, or perhaps that they'll have to use a green-screen client, so they avoid making too many change requests to the mainframe team.

"Since mobile teams turn around new releases much faster than mainframe can match, this causes an impedance mismatch in the organization, so companies harbor a two-speed IT function," says Venkat Balabhadrapatruni, Chief DevOps Architect of Broadcom's CA Mainframe group. "Mobile teams start relying on mock interfaces or service virtualization to simulate the mainframe and delay the need to interact."

The whole DevOps movement isn't just for born-on-the-web teams and technology. Mainframe teams can accelerate updates to more than monthly intervals, automate build processes and create well documented test suites.

Both sides need to improve here. If companies can simply provide better documentation for mainframe APIs, mobile app teams find it's easy to run agile development projects using logic from mainframes.

"An insurance company told us the practice of documenting APIs with REST and Swagger has been much easier to adopt than the old SOAP/WS* days, and instilled a real sense of collaborative, test-driven discipline between teams," says Venkat.

Source: https://intellyx.com/2019/10/28/will-mobile-development-bring-mainframe-development-to-the-mainstream/

Performance and reliability at scale: Critical then, critical now

One thing is certain – mainframes shouldn't have any issues reliably supporting the demands of mobile functionality.

The modern mainframe usually has an abundance of horsepower to spare, and the ability to handle tens of thousands of requests or events a second. Performance-minded teams need to focus on load balancing and tuning all layers of the mobile application that interact with the mainframe in deployment to avoid bottlenecks.

"The scale and QoS (quality of service) characteristics of mainframes are legendary – in fact the mainframe team could educate just about any app development team about high availability," Venkat said.

In some environments, you may find the mobile app service co-located in a container workload, running on the mainframe, to seamlessly serve up information from multiple back end services with close to zero latency.

The Intellyx Take

Embracing openness will pave the path from mainframe to mobile. Open source projects like Zowe messaging and open API standards are opening up the doors for any developer to call on these workhorses through simple command line scripts or web-based integration tools.

"Overall, M2M has changed the IT organizational dynamics of most large companies," Venkat mentioned. "Mainframe teams aren't off in a silo anymore, they are part of the mainstream app development group in the organization."

Good news. Mobile development, and mainframe development, are already getting along swimmingly in the mainstream.

Jason English
Principal Analyst & CMO, Intellyx. Twitter: @bluefug

Source: https://intellyx.com/2019/10/28/will-mobile-development-bring-mainframe-development-to-the-mainstream/