THE 5 MOST IMPORTANT STEPS FOR LEGACY MAINFRAME MODERNIZATION

Overview: Mainframe modernization was already a critical need for many businesses. In a post-COVID world, there will be even less time to waste—and smaller margins for error. Here are 5 crucial steps for successful modernization.

In countries like Australia, where we're still grappling with the pandemic yet beginning to chart our economic recovery, business realities and economic uncertainties are injecting new urgency into questions around legacy processes on the mainframe.

Among organizations that have tried to transition "off the mainframe," we've seen many teams invest millions of dollars and precious R&D time only to conclude that the mainframe's performance, reliability, security and scalability can't be matched. In another corner are organizations who've simply decided to tolerate delivery bottlenecks caused by legacy systems and tools.

Going forward, there will be little room for either approach.

Modernizing mainframe software development and delivery was already a critical step for making businesses more responsive to customer needs and competitive with smaller, nimbler players. Now, there's even less time to spare. So how can you get the ball rolling, especially in a new landscape that often demands quicker results from fewer resources?

Here are five vital steps for bringing the mainframe's legacy systems and processes up to speed—fast.

1. Understand Modern Mainframe Capabilities and Create a Culture That Supports Them

The biggest roadblock to faster mainframe development tends to be that organizations simply aren't taking advantage of modern mainframe capabilities. And the reason for that is often two-fold: they don't understand their current state nor the extent of what's available and/or don't have the cultural support for change.

Your first step should be making sure you've explored the full capability and availability of tools and processes. But, as many businesses have found out the hard way, that won't be enough by itself. You'll also need to accelerate a cultural shift toward continuous improvement and learning—part of what we call a "growth mindset," which involves actively looking for challenges and a willingness to "fail fast." That likely means re-evaluating standard processes and KPIs, but it also demands a quick show of value.

2. Map Out Your Development Processes to Identify High-impact Opportunities

A minor time investment at the beginning of your modernization journey can yield faster results down the track.
Some of the most successful projects we've seen have started with a Value Stream Mapping exercise. This involves outlining your entire development process, from an initial idea to final code delivery, which can help you diagnose your most serious bottlenecks or delays. Try to restrict your focus to an area that stands to deliver the biggest impact on time and cost. For example, the automation of testing processes is a common opportunity for a quick win.

3. Use Small Successes to Convert Your Biggest Skeptics Into Influencers

Those quick wins will be crucial for reinforcing the cultural change mentioned in the first step. Some long-time mainframers might be cynical about what's practicable, but they also tend to command understandable influence and are open to new evidence. That's where quick wins can help you prove value and turn skeptics into champions for change.

This approach has a dual benefit. First, it helps get your immediate project over the line, but it also builds the growth mindset that's necessary for more long-term success. There's arguably no better way to acclimate an organization to the value of rapid iteration and "failing fast" than to showcase immediate benefits, no matter how narrow or rudimentary they might seem.

4. Attract New Mainframe Developers by Updating Old Myths

It's crucial to find ways to get the mainframe's current custodians on-side. Still, we're already starting to see attrition in this group, meaning that the market is losing decades' worth of knowledge. You'll need a succession plan.

This is why modernizing mainframe tools and processes and cultivating a growth mindset are such crucial early steps—they're necessary for appealing to new talent and graduates who are hungry for

interesting challenges.

But that doesn't mean you can modernize and passively wait for developers to come to you, because there's an awareness problem here. Businesses will need to proactively reach out to new talent and work to dispel misconceptions about the mainframe.

Many of our own development team members are recent graduates, and those hires were often the product of actively drawing parallels to what they had learned in university. After all, once they're connected to what developers have learned, powerful servers that run huge portions of the world's activities aren't as boring as old myths would imply.

5. Create Developer Experiences That Are Consistent With Newer Talent's Expectations

Outreach is necessary but not sufficient. You'll also need to make sure you're offering a more modern developer experience, one that's more in line with the tools and approaches familiar to recent grads.

IDEs offer an instructive starting point. Green screens and shortcut keys feel foreign to those who've been learning through more modern graphical user interfaces. Tools that graduates recognize, ones that are fully user-interactive with options to cut and paste or drag and drop, can more quickly introduce them to mainframe code development and make it a more appealing career path.

Questions About Mainframe Modernization?

Disruptive models and agile start-ups have long threatened to upend more established businesses. Transforming the mainframe is a necessary step in making sure competitors aren't eating your lunch, especially now that so many recoveries hinge on more agile, customer-centric practices.

Our teams have helped many organizations with this process—we know what works and what doesn't. Get in touch for tailored advice about how to remove barriers, minimize costs and ensure your mainframe is fit for the realities of our new landscape.

This post also appeared on LinkedIn.