



Mainstreaming the Mainframe

Why Enterprise DevOps Should
Take Control of Legacy Applications—Now!



Like everyone else, you're being bombarded with content. So you probably read only a fraction of what you encounter. We ask that you approach this document differently, because we genuinely believe it offers insight that is unique, fact-based, actionable and essential to the success of any company with a mainframe. We also ask that you read it even if you aren't currently responsible for your company's mainframe—because, as we argue, that responsibility now needs to become part of mainstream IT.

Key Concepts

- 1) **The clock is ticking on siloed mainframe teams.** Skilled mainframe professionals are retiring—and there are few new ones to replace them. Yet mainframe applications remain essential. IT therefore needs a viable strategy for maintaining and nimbly adapting mainframe code in response to relentlessly changing business requirements.
- 2) **IT must respond by rapidly mainstreaming the mainframe.** Instead of continuing to depend on mainframe specialists, IT must empower DevOps staff with mainstream skills to treat the mainframe as just another platform in the multi-platform enterprise—using the same DevOps tools and processes they use for Java, x86 Linux servers, cloud, etc.
- 3) **Mainstreaming doesn't just solve a mainframe problem.** It delivers competitive advantage to the entire digital enterprise. To excel in digital markets, it's not enough to be big. You have to be fast, too. An enterprise DevOps environment that supports agility and collaboration best practices across all platforms—including the mainframe—is therefore a competitive requirement.

THE END OF THE MAINFRAME SILO

For decades, IT has treated the mainframe as a distinct functional silo. Mainframe staffs have distinct skills, use distinct tools, employ distinct processes and even possess a distinct subculture.

This silo came about for understandable reasons. The mainframe, lest we forget, significantly predates distributed computing—so for years it was the only IT game in town.

When distributed computing finally emerged, mainframe teams had concerns about protecting the stability and integrity of their environments. And many of those concerns—especially those regarding reliability, complexity, cost and security—turned out to be quite valid.

As with all tragedy, however, the isolation that kept the mainframe stable and secure through decades of technology tumult is now proving to be its undoing. The mainframe has become like a secret cult attended to by its own insular priesthood, aloof and disengaged from the wider world of commodity x86 infrastructure and the cloud. And most IT leaders have exacerbated the situation by aggressively cutting mainframe costs, rather than investing in mainframe application development.

The result: mainframe environments are insufficiently responsive to the business—not because of any inherent property of the mainframe itself, but rather because of how it has been managed.

Also, because the mainframe has been siloed and neglected for so long, few IT professionals outside of the mainframe subculture know COBOL or understand how MSUs work. They may write code that calls DB2 on the mainframe—but the platform itself is completely opaque to them.

The siloed, non-agile approach to mainframe DevOps is no longer tenable for several reasons, including:

- **Mainframe-specific skills attrition.** The highly skilled mainframe professionals who have been managing the mainframe and its applications are aging out of the workforce. There is little likelihood that younger IT professionals with similarly high technical aptitudes will want to consign themselves to isolated mainframe-specific careers.
- **Irreplaceable code.** Mainframe applications are so evolved and ingrained into the business that they are indispensable. And they cannot practically be re-platformed. Companies must therefore figure out how to preserve both mainframe applications and the mainframe platform over the course of the next decade or more, despite attrition of their mainframe specialists.
- **Inter-platform dependency.** Digital business superiority depends in part on IT's ability to leverage all available application logic and data across and beyond the enterprise, regardless of platform or programming language. So companies must do much more than merely maintain their mainframe applications as self-contained entities. They must also more aggressively evolve and leverage mainframe application code and data in tandem with their non-mainframe assets.

- **The need for speed.** The pace of mainframe development cannot continue to be egregiously slower than the rest of IT. Agile responsiveness is of the essence, especially when it comes to back-end support for customer-facing mobile apps. Slow DevOps processes on the mainframe undermine business agility in a toxic and potentially fatal way.

Simply put, companies cannot accept the mainframe status quo at a time when mainframe applications, data and processing power are more valuable to the business than ever. IT can only give the business all the digital capabilities it needs, when it needs them, if the mainframe is just as agile and accessible to developers, data analysts and operations as every other platform.

All IT leaders must therefore confront the mainframe issue decisively and presently, regardless of their personal prejudices and perceptions—and regardless of whether they even have direct responsibility for the mainframe as things stand today.

A Brief History of Mainframe Territoriality

Mainstream IT has always struggled to leverage mainframe resources in the context of siloed mainframe territoriality. That struggle has played out in five successive stages:

STAGE
1

Terminal Emulation

Mainframe staffs permit access to mainframe applications and data from users' PCs via "screen-scraping" and other techniques.

STAGE
2

Application Calls

Mainframe staffs allow distributed applications to access mainframe data, application logic and transaction processing on a case-by-case basis.

STAGE
3

End-to-end Monitoring

Mainframe staffs share selected platform telemetry and enable limited access by enterprise performance/fault monitoring tools.

STAGE
4

Batch Analytics

Mainframe staffs set parameters for delivering data in bulk to mainstream enterprise BI tools—and, later, Hadoop-based environments.

STAGE
5

Mainstreaming

Mainstream enterprise DevOps teams are fully empowered to leverage mainframe resources directly—with appropriate policy-based controls.

MAINSTREAMING THE MAINFRAME

The mainframe cannot continue in its current siloed state. But IT must ensure its ongoing viability, because its applications are indispensable and cannot be re-platformed. The one logical conclusion: IT must ultimately bring the mainframe into its mainstream cross-platform DevOps work processes.

This mainstreaming must address three core areas of mainframe functionality:

Mainframe Applications

Existing mainframe applications present IT with perhaps its most important and most challenging mainstreaming objective.

This objective is **most important** because:

- 1) Mainframe application logic is extremely valuable to the business.
- 2) The re-platforming of that application logic has proven to be highly impractical.

Existing mainframe applications represent IT's **most challenging** mainstreaming objective because:

- 1) Mainframe applications have been running so long and have been modified so often that they are usually not very well documented any more—and may not even be well understood by IT.
- 2) Fluency in legacy languages such as COBOL, PL/I and Assembler tends to be pretty scarce.

The good news is code is code. So while Millennial developers with mainstream skill sets may not be familiar with the particular syntax of COBOL, the basic principles of application logic still apply. And given the proven adaptability of today's developers when it comes to learning new programming language syntaxes, the current dearth of COBOL expertise does not actually present IT with an insurmountable problem.

In fact, mainstream developers can readily take charge of mainframe applications simply by being given the ability to:

- Write, modify, debug and manage mainframe code in their preferred IDEs
- Receive immediate guidance and feedback on software quality issues as they code
- Include mainframe code in the same automated test and delivery environments they use generally
- Better understand existing mainframe application logic through visualization of runtime behaviors, inter-program calls, etc.

Ultimately, the normalization of mainframe application code into the broader enterprise DevOps environment enables IT to treat mainframe code just like any other code in any given agile project—so it can be rapidly adapted to changing business requirements without problematic process friction or cost.

Mainframe Data

Unlike mainframe applications, mainframe data can at least theoretically be re-platformed. And there are some use cases where it makes sense to export specific mainframe data sets to other environments for application staging or analytic purposes.

Generally speaking, however, it makes sense to leave mainframe data on the mainframe. Reasons for this include:

- Better application performance
- Hosting data in multiple locations adds avoidable cost
- Security, compliance and/or governance requirements

That said, mainframe data should nonetheless be visible and comprehensible to any mainstream developer or data analyst with appropriate needs and authorizations. So, as with mainframe code, IT must give non-mainframe staff the ability to intuitively discover and understand mainframe data, metadata, data structures and data dependencies across mainframe programs and copybooks. And, again, this should ideally be done with tools that have a familiar look and feel (i.e. Java-like “projects”).

Mainframe Operations

Operations is a special case of mainstreaming—because, at some level, all IT operations are inherently siloed. Different technical teams with different skills and tools manage Windows and Linux systems, storage infrastructure, network devices, databases, middleware, etc. So IT will likely continue to depend on IBM z Systems specialists to perform certain essential mainframe management and tuning tasks.

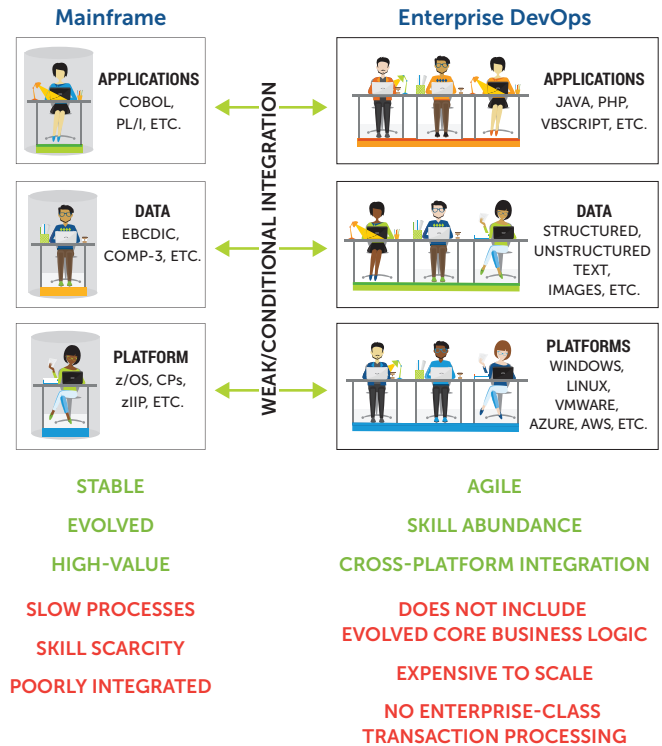
But, as noted above, end-to-end application service levels often depend on all these separate components combined. To safeguard these service levels, IT must more fully integrate mainframe operations into enterprise operations. This integration has two basic aspects:

- 1) Management of data and alerts from the mainframe must be integrated into mainstream enterprise management workflows.
- 2) Enterprise operations staff must be able to “drill down” into the mainframe as they investigate application service level issues—without always having to depend on IBM z Systems SMEs.

Some IT organizations have already given their mainstream operations staff some rudimentary visibility into the health of certain mainframe resources and processes. But the integration of mainframe operations must be taken much, much further if companies are to gain the considerable benefits of true cross-platform DevOps.

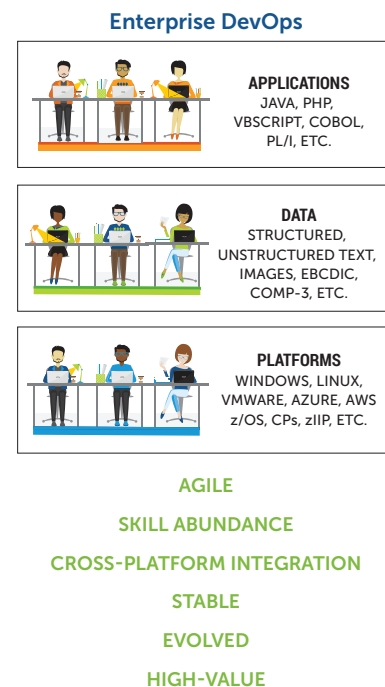
The key principle across all three of these imperatives is that normally skilled IT professionals must be able to work on the mainframe using their chosen tools and processes. Yes, IT must safeguard the integrity of the mainframe environment. But the mainframe cannot continue as an isolated silo in the enterprise. Responsibility for the mainframe must ultimately be transferred to mainstream developers, data analysts and operations staffs.

THE SILOED MAINFRAME



IT organizations that let the mainframe remain a functional silo will be less responsive to the business and face the existential threat of failed systems of record.

THE MAINSTREAMED MAINFRAME



By empowering mainstream developers, data analysts and Ops teams to own the mainframe, IT can optimize end-to-end digital agility and ensure the long-term integrity of core systems.

THE MAINSTREAMING BUSINESS CASE

IT leaders have a lot on their plates. So there have to be some pretty compelling reasons to escalate mainstreaming of the mainframe to the top of the 2016 to-do list.

And there are. They include:

- **Mitigation of Existential Mainframe Risk**

The looming loss of mainframe skill sets is as serious a risk to the business as Y2K was. It's just that the date of the disaster isn't fixed. Companies that continue to procrastinate on this issue will eventually see the value of decades of investment in business-critical application logic evaporate.
- **Essential Business Agility**

If you can't modify your mainframe application logic quickly and with confidence, your business can't be sufficiently nimble to compete in today's environment of constant digital disruption. Enterprises simply have to integrate mainframe code management into their mainstream Agile/Continuous Delivery SDLC processes.
- **More Value and a Better Experience for the Customer**

Companies in all markets need to leverage information and insight to do more for their customers than their competitors. Much of that information and insight resides on the mainframe. Companies that can't aggressively and adaptively leverage that information and insight because their mainframes are slow and closed will invariably lose to nimbler competitors.
- **Compliance with Greater Confidence and Less Friction**

Silos are the enemy of compliance. They prevent policies from being implemented in a common manner across the enterprise, and they fragment auditing in ways that add cost and arouse the skepticism of auditors. By more fully integrating the mainframe into the broader enterprise environment, IT can unify compliance processes to optimize credibility while reducing costs.
- **Attracting Millennial Talent**

Enterprise IT organizations have to get the next generation of reasonably skilled and motivated IT professionals working on mainframe applications, data and operations. That will be very hard to do with tools and processes from the 80's.
- **More Scalable, Reliable, Secure and Cost-efficient Enterprise Computing**

The IBM z Systems platform, it turns out, is actually a terrific place to run all kinds of Linux and open source workloads—especially when compared to sprawling x86 infrastructure that is hideously complex, unreliable and expensive to operate. By de-siloing the mainframe, IT can take advantage of its superior performance and reliability, fixed physical footprint and remarkably low incremental costs.

Ultimately, though, IT has to mainstream the mainframe because it has little choice. Its current condition won't be tenable much longer, and it cannot simply be eliminated. Mainstreaming offers the only way out. It also offers a great way up.

Mainframe applications are irreplaceable and incalculably valuable business assets. Bimodal IT consigns them to irrelevance. Transformational inclusion of the mainframe into enterprise DevOps, on the other hand, ensures their long-term value and optimizes the ability of large enterprises to successfully compete in the world's increasingly digital markets.

No Platform Left Behind

Confronted with the importance of mainframe applications and the historical difficulty of making the mainframe fully Agile, a number of leading industry pundits have suggested something termed “Bimodal IT.” The essence of Bimodal IT is that companies should abandon any attempt to make the mainframe Agile—and should instead simply segment itself into “stable mode” and “agile mode.”

While this approach may be tempting to those who flinch at the prospect of actually transforming the mainframe, it is not a viable alternative for several reasons:

- **Business agility requires mainframe agility.**
Enhancements of the digital customer experience frequently depend on core databases, transaction processing systems and highly-refined business logic that still reside on the mainframe—and probably always will.
- **Stability and agility are not mutually exclusive.**
To suggest that they are is actually to deny the whole movement towards DevOps, Continuous Delivery, etc.
- **Defaulting to the status quo is not a strategy for competitive differentiation.**
If it were easy to bring Agile best practices to the mainframe, every company would have already done it—and it wouldn't offer such significant competitive advantage.
- **Mainframe reliability, performance at scale, and security are needed more, not less.**
It has proven to be excessively expensive (and often impossible) to replicate the mainframe's qualities in distributed/cloud environments. So IT should focus on better exploiting them, rather than letting them go to waste.

ABOUT COMPUWARE

Compuware empowers the world's largest companies to excel in the digital economy by fully leveraging their high-value mainframe investments. We do this by delivering highly innovative solutions that uniquely enable IT professionals with mainstream skills to manage mainframe applications, data and platform operations.

Learn more at compuware.com.

Compuware Corporation World Headquarters • One Campus Martius • Detroit, MI 48226-5099

© 2016 Compuware Corporation. Compuware products and services listed within are trademarks or registered trademarks of Compuware Corporation. Java is a registered trademark of Oracle and/or its affiliates.

