The Bloor Group

The Agile Mainframe

Compuware Re-Energizes the Mainframe

Robin Bloor, Ph.D.
& Rebecca Jozwiak

WHITE PAPER

The Mainframe World

While the likes of such companies as Google and Facebook build upon the power of distributed computing, large corporations know that some tasks simply run better on mainframes. Organizations with high volumes of transactions – banks, telcos and retailers – have for decades trusted the mainframe for its impeccable performance and reliability with transaction processing. The applications running on mainframes support the business and create value, and just as technology has evolved to meet modern demands, the workhorse mainframe has evolved to address today’s business needs.

The z13 mainframe IBM introduced to the world in January 2015 is the most capable business computer on the planet. It is the product of a $1 billion development project that spanned five years. In its largest configuration, it is capable of processing over 2.5 billion transactions per day. It was built with mobile computing in mind, able to process vast numbers of mobile transactions, while encrypting and analyzing them on the fly. In effect, the mainframe’s characteristics of remarkably secure operation, extreme reliability, manageability, workload versatility and lightning performance were molded toward mobile computing.

Of course, that was not the only focus. IBM was nursing the mainframe as a virtual server environment long before the term “private cloud” was invented, and the z13 steps up this capability. It is able to run and manage 8,000 virtual servers – a kind of distributed computing that is not physically distributed at all and avoids many of the pains of distribution.

Technically, if you care about such details, the z13 sports “the world’s fastest microprocessor, 2x faster than most common server processors, 300% more memory, 100% more bandwidth and vector processing analytics.” It tops out at 10 terabytes of main memory and includes innovative architectural improvements to multithreading and parallelism. This should confirm, if you had any doubt, that IBM is still fully committed to the mainframe.

Mortality and the Mainframe

The demise of mainframe computing was erroneously “predicted” in the early 1990s with the advent of client/server computing and the dawn of the Internet. It is easy to understand in retrospect. There was a major paradigm shift to distributed computing, and many species of server computers, including mainframes from lesser companies than IBM, did indeed disappear in the wake of that. The Unix and Windows server markets mushroomed and grabbed the lion’s share of the Internet market. The IBM mainframe had a series of rocky years, losing some customers at the lower end of the market, while IBM scrambled to adjust to the rapidly changing technology landscape.

Nevertheless, the IBM mainframe was (and still is) the most powerful commercial transaction engine available, and many of the most demanding systems in the world were implemented on it. There was no comparable platform. The mainframe survived after all, and IBM continued to invest in and improve upon its capabilities, and while it lost some customers, it gained others.

Economies of Scale

Even in its worst years, the 1990s, the mainframe was still the most economic computing platform, as indicated repeatedly by independent surveys – and this is still the case. For example, in a study spanning 2012 to 2015, Rubin Worldwide surveyed 498 large companies and has found the gap between distributed and mainframe costs has widened given the rising density of technology requirements. In fact, while computing power has doubled over the last five years, server-heavy organizations’ costs have gone up 63% more than mainframe-heavy organizations. In addition, their analysis across 15 industries shows that the average IT cost of goods is 35% (on average) less for mainframe-heavy organizations, with the greatest differences in the financial sector, where the average cost is an astounding 69% less.

There is no great mystery as to why. The mainframe offers 99.999% availability, has a top-rated EAL5 security classification and can deliver close to 100% resource utilization. Additionally, it is particularly suited to mixed workloads, which are difficult to accommodate reliably and inexpensively in distributed environments.

The mainframe also proved remarkably capable of accommodating Linux virtual machines – a capability IBM built and made available a few years before VMware even had such a capability. This initiative has proven so successful that nowadays IBM provides a Linux-devoted mainframe capable of scaling up to 8,000 virtual machines (LinuxONE) that effectively supports many open source products such as KVM, Apache Spark and Docker. However, most mainframe users employ this capability to mix Linux workloads with mainframe workloads.

Mainframe Economics

Mainframes account for 68% of production workloads but only 6% of IT spend.[1]

Server-heavy IT organizations' costs have risen 63% more than mainframe-heavy organizations' costs over the last five years.[2]

The average IT cost of goods is 35% less for mainframe-heavy organizations.[2]

[1] Solitaire International Analyst Report 2014 commissioned by IBM
[2] Dr. Howard Rubin, Rubin Worldwide, Gartner Senior advisor

The mainframe currently enjoys about 10% of the server market (by hardware revenue) and occupies a primary role in the computing infrastructure of many large corporations. Its unmatched economies of scale account for its popularity. It has been the focus of many server consolidation projects. Many mainframe users have reduced their investment in distributed computing as a result of moving applications to the mainframe. Another trend that has worked in its favor has been "application extension." It has become the home of mobile applications that extend or enhance the usefulness of transactional systems, and it is, as part of a parallel trend, likely to have a significant role to play in the Internet of Things (IoT).

What's Not to Love?

Given that it offers the best value for money in corporate computing, it is a little surprising that the mainframe is not more popular. However, there are reasons for this. The entry level costs of mainframe computing are high, and any new adopter needs to be convinced that the organization can grow its value quickly to a level where its natural economies of scale advantages take effect.

Another discouraging factor is the availability of skilled staff. The availability of both software development and operational mainframe skills is more constrained. In part, this is due to colleges and universities worldwide paying little attention to the mainframe world, as they believed that it was in terminal decline.

This had far-reaching effects. The steady stream of new ideas and techniques, which were nurtured in university environments and later gave birth to thriving ecosystems of startup companies, made little contribution to mainframe computing. Unix, Linux and Windows were hotbeds of innovation, giving birth to new development languages and innovative development products driven by new ideas: the graphical user interface, object orientation, document stores, multi-tier software architectures, web services, messaging systems and so on. In contrast, the mainframe software world became increasingly isolated with few software startups feeling the need to port their products to the mainframe and compete with traditional mainframe software vendors. This in turn acted as a barrier for IT staff to embrace, expand or migrate to the mainframe world.

It was an IT culture that held tight to its traditional ways of working. Experienced mainframe staff showed little interest in the IT world beyond their mainframe silos. It might have remained that way if the aging process of mainframe staffs hadn't gradually provoked a skills crisis, which, in turn, provoked IBM and other mainframe software vendors such as Compuware to action.

On one hand, technology developments outside the mainframe world failed to produce an alternative platform that could match the qualities and economies of mainframe computing – for many applications, migration made no sense because there was nowhere to migrate to. On the other hand, a mainframe skills crisis was looming, and it would only get worse with time. By 2005, the number of colleges providing mainframe courses had reduced to 24 worldwide.

IBM launched a z Systems Academic Initiative, which sought to gradually rectify the situation. There are now more than 1,000 colleges and universities in 66 countries providing courses and offering labs in IBM mainframe technology. IBM supports this by making mainframe access available via the cloud. It runs an annual Master The Mainframe contest for students and has also stimulated the z Systems jobs market by setting up a website, SystemzJobs.com, where prospective employees, including college graduates, can post their resumes and browse through a multitude of job openings.

These investments helped to turn the ship around, but staffing shortages remain, and the mainframe is still held back by its historical culture. While the mainframe has participated in some of the recent innovations in the software landscape, it has always done so as a laggard. It was never the leader – it was always the follower – and sometimes it never participated at all. For example, one does not think of agile software development as a characteristic of mainframe development. That's because it isn't.

The Mainframe Revisited

The mainframe occupies a stable and unchallenged position at the heart of corporate computing. It offers advanced technology, extremely high availability, the highest levels of security and repeatedly proven IT economy. The next generation of mainframe staff are entering the workforce, and they will gradually replace those who are leaving. In the past, it was often the case that a CIO would have a mainframe background, but it is no longer so common.

Unfortunately, most of the next generation of CIOs will have little exposure to the mainframe world, and yet it is they who will preside over the mainframes. Coming from Unix, Linux or Windows environments, they will probably be surprised and perturbed by the long development cycles and the lack of modern interfaces that curse most mainframe software. They will likely be impressed by the Linux capabilities but will quickly understand that while the mainframe is an excellent Linux platform, by far the most efficient software running on the mainframe is native z/OS software. The native z/OS applications will continue to be maintained and extended, but unless the software development culture, processes and tools change, few new ones will be built outside of Linux partitions.

Given the operational characteristics of the mainframe, this makes no sense. Agile development methodologies and techniques need to be imported to invigorate the mainframe software culture, which is still, to some degree, based on the waterfall methodology and waterfall thinking. There is no technical reason why mainframe software development could not be infused by the proven effectiveness and the innovative spirit of the agile distributed computing world.

It may seem odd to think of agile tools for developing and maintaining COBOL and PL/1 applications, but agile tools are no strangers to C++ and Java, which in many mainframe sites have superseded Assembler and COBOL or PL/1 as primary development languages. Because of that, the Eclipse integrated development environment (IDE), which was invented for Java, has gained a significant foothold in mainframe development environments and nowadays supports COBOL and other mainframe programming languages. Eclipse is the de facto standard for teaching programming globally. Gartner claims that 98% of developers are familiar with it.

If a full suite of agile software development tools were available on the mainframe, there is no reason why the agile development techniques and practices wouldn't blossom there, as they did in the distributed software world. This would, in our view, provoke a much-needed evolution to the staid culture of mainframe software development and further revive innovation opportunities in leveraging mainframe advantages. Imagine the mainframe embracing agile development: collaborative cross-functional teams, early delivery, frequent releases, rapid and flexible response to change and continuous improvement. Such a renaissance is possible; indeed, it may have already begun.

Compuware and the Mainframe

The mainframe renaissance we described is the vision Compuware, a company that has the acumen to lead the evolution towards mainframe agility. This is not dreamware born of a fertile marketing imagination; it is a cultural change that Compuware has first applied to itself. As such, Compuware is "converting to its own evangelization," not just in terms of the software products it builds but in the cultural change its CEO, Chris O'Malley, has imposed on the company.

So not only does Compuware promote the idea of frequent software releases, it has also organized its development teams to deliver them. It doesn't just evangelize agile development and continuous delivery, it is demonstrating it to its customers. For example, in the past year it has delivered four releases of its flagship product, Topaz.

The immediate question is: "What does Compuware mean by continuous delivery?" It is not simply a slogan; it is based on the overall DevOps methodology, illustrated in Figure 1.

As the diagram suggests, the software development process is iterative and intertwined with operational deployment. On the development side, as user requests are processed, applications and data are analyzed. Source code and data are edited, and changes are continuously built, tested and debugged into the product. This process supports continuous sprint reviews where the product is validated with customers. Changes are fed back into this process until the right user experience is achieved. It is then deployed. At that point, an automated deployment cycle kicks in, where application usage is monitored and audited, problems are diagnosed and application performance is tuned. Collaboration between Dev and Ops is further supported as feedback from customers and operational problems are provided to developers. Any proposed changes are automatically tracked, authorized and deployed. While we have described this in terms of a single application, the scope of this dual cycle could be a whole system of applications.

This agile DevOps methodology is typical of the full stack development culture that pervades

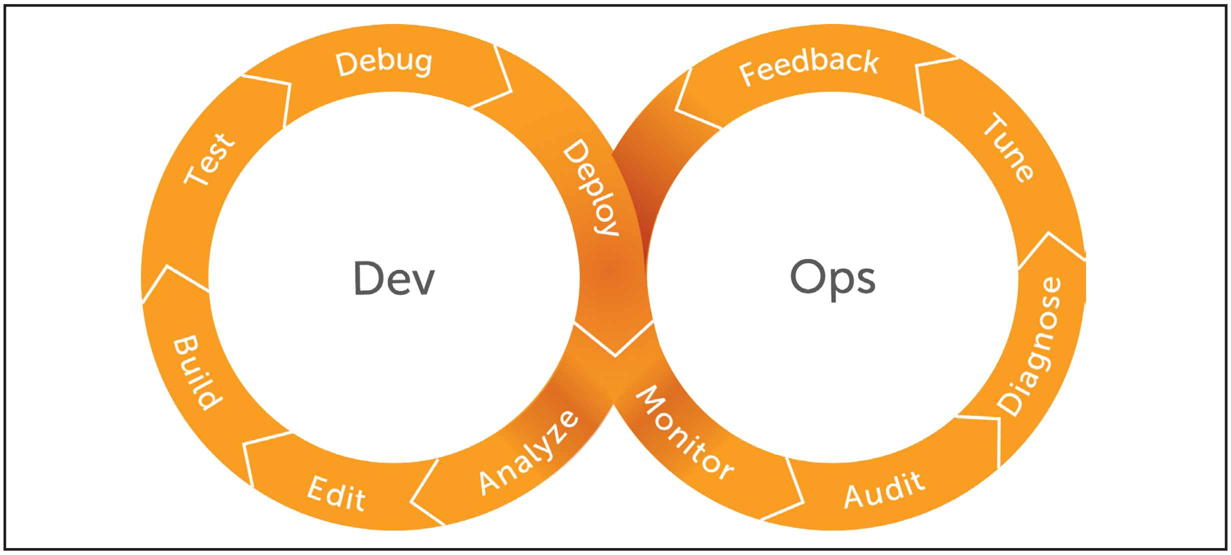

Figure 1. Compuware's DevOps Methodology: Outline

the Intel commodity server world of Linux and Windows virtual machines and the newly emerged Hadoop world of large clusters. It provides a distinct and refreshing contrast to the ponderous, silo-ed mainframe world of waterfall development methodologies and slow, year-long-or-longer development cycles.

With the introduction of Topaz, Compuware is well positioned to enable their customers to adopt the agile DevOps methodology it has adopted and now evangelizes; but they are also aware that no single vendor can singularly provide all the software tools necessary. To advance their mission, Compuware announced two significant events at the start of 2016. First, the company acquired the assets of ISPW, a leading agile source code management and release automation solution for cross-platform development. Second, they announced far-reaching partnerships and integrations with non-mainframe DevOps leaders, including open source providers.

The Topaz Suite

Topaz is the proof of Compuware's commitment to revolutionize mainframe software development so that a single development team can support any type of code, whether it be COBOL or Java. It is a bold and unapologetic departure from the old mainframe developer style, delivering a modern development environment that is powered by many of Compuware's leading solutions.

Topaz also provides new capabilities such as an enterprise data editor and intuitive visualization capabilities that further empower non-mainframe trained developers. Newer generations of developers are insulated from the arcane nature of mainframe development, but even traditional mainframe developers would be remiss not to take advantage of the more productive environment and capabilities.

The Topaz suite of modern tooling consists of:

Topaz Workbench

Powered by stable and reliable Compuware components – Abend-AID, File-AID, Hiperstation, Strobe and Xpediter – the Topaz Workbench conceals the complexities of traditional mainframe interfaces with an integrated interface that would not look out of place on a Linux, Windows or OS workstation.

In short, the fundamental capabilities of the products are there, but the way you work with them is entirely different.

If you look under the hood, you'll discover the following Compuware mainframe products:

- Abend-AID – source code and data analysis tool for swift application failure analysis and resolution
- File-AID – data management tool that enables data reformatting and provides quick and convenient access to and securing of test data and files
- Hiperstation – automated testing and auditing tool for quality assurance and data breach deterrence
- Strobe – application performance monitoring and analysis tool for identifying and resolving performance and throughput problems and inefficiencies
- Xpediter – debugging and analysis tool that enables interactive testing and surfaces knowledge of application functionality

The point with Topaz Workbench, however, is that you don't have to look under the hood.

Compuware achieves the ease of use modern programmers are accustomed to by leveraging Eclipse, the popular open-source IDE. Eclipse provides Topaz Workbench with all the capabilities that a veteran mainframe developer has acquired through years of experience, delivered via Java – already a popular mainframe programming language – while supporting traditional mainframe languages. The soon-to-retire mainframe developer may choose to remain with what is familiar, but Java is now a well-supported option, and modern languages can be employed where appropriate in cross-platform developments. This will serve to invigorate what would otherwise remain a staid and stale environment.

Topaz for Enterprise Data

Getting access to and managing mainframe data has always been idiosyncratic, and it has never been easy to materialize views that combine mainframe and non-mainframe data. The cultural disconnect between veteran mainframe professionals and the new generation of developers is constantly plagued by this challenge.

With Topaz for Enterprise Data, both camps can access and manipulate data on very different platforms and DBMSs in the same format. This provides a huge productivity boost across the organization. The Topaz Enterprise Data Editor eliminates the need for source-specific tools to manage data at the mainframe level by providing a single user interface that spans both mainframe and distributed data.

Topaz's Relationship Visualizer allows users – through highly graphical visualizations – to view and manage data object relationships spanning the enterprise. As is often the case when different teams are not working on the same views of enterprise data, few truly understand the complex data relationships. The Relationship Visualizer specifically addresses that problem.

Topaz for Program Analysis

It's not uncommon that a mainframe program has been running for 20 years, and only one person knows what it does – or worse, no one is quite sure. As the pool of mainframe experts shrinks, it becomes increasingly important to ensure someone can step in and take over the preservation and advancement of invaluable and irreplaceable mainframe programs. Mainframe programs that are well understood by veterans can be completely foreign to newcomers, especially when years of code changes have gone undocumented. Replacing retiring developers can then be an intractable problem.

Topaz for Program Analysis is designed to turn what has often been a daunting learning curve into a seamless transition. This component performs analysis on running mainframe programs and returns the results in a graphical manner via the Runtime Visualizer. The visualizations include detailed views of how mainframe programs interact with each other when an application is launched, what external calls a program is executing at any given time and how often those calls are being made. It can also visualize a program while you are editing to understand the current behavior and see the effects of changes.

Topaz Runtime Visualizer provides ground-breaking capability with easy-to-understand graphical visibility into the complex interactions between programs – as they execute. It enables IT to confidently enhance and adapt even old and poorly documented systems.

Topaz for Program Analysis also has an impact analysis capability that lets users drill down into the code and copybooks associated with each program. In some circumstances, developers can perform maintenance tasks such as debugging and troubleshooting without touching the source code.

Topaz for Java Performance

With enterprises continuing to embrace Java technology, the deployment of Java Virtual Machines (JVMs) has become far more common. Many mainframe environments now leverage Java for batch processing and WebSphere for application and integration middleware. Topaz for Java Performance serves this market in ways that a large WebSphere environment does not. Java Batch is a relatively young technology on the mainframe but one that will require tools to help customers maintain the same level of performance they have become accustomed to from IBM's highly optimized COBOL.

It is a dynamic visual analysis tool for Java Batch and WebSphere performance. Users can, for example, monitor CPU usage and determine if more resources need to be allocated to a JVM. They can identify the JVM's heap memory and adjust the memory allocation as needed. Topaz for Java Performance also delivers visibility into issues such as memory leaks and blocked threads.

The Java class library is enormous, and when performance issues arise, developers can spend an inordinate amount of time sifting through the classes to determine which one is causing the problem. Topaz for Java Performance identifies called Java class methods and allows sorting of the class methods by CPU usage, thus reducing the time required for a developer to discover which class is degrading performance and negatively impacting service levels.

ISPW to Speed Change and Improve Quality

Compuware recognized the necessity to reduce development and delivery delays caused by outdated mainframe source code management products that is inherent to waterfall methodologies. With the acquisition of ISPW, they added modern and elegant source code management and release automation capabilities into their portfolio. ISPW supports agile development and continuous deployment by enabling parallel and concurrent development. Source remains in an ISPW central library allowing universal access. Developers benefit from synchronization, visibility and notification through each stage of the application process. The product also provides end-to-end tracking for a view of every action that affects code running in production – or at any level. When checking out code it will alert you to potential impacts with other programs. The browser interface for approvers and management enables any-time approvals, even from a mobile device. When used in conjunction with Topaz Workbench, developers have access to Compuware's full suite of developer tools.

Partnerships and the Topaz Ecosystem

Consider Figure 2. It illustrates Topaz's central position in the intertwined DevOps cycles while, at the top of the diagram, it highlights Compuware's efforts to enrich the Topaz ecosystem with a series of strategic partnerships with best of breed vendors of complementary DevOps technologies. The goal here is to deliver a modern, fully functional environment that spans both the mainframe and the distributed processing worlds.

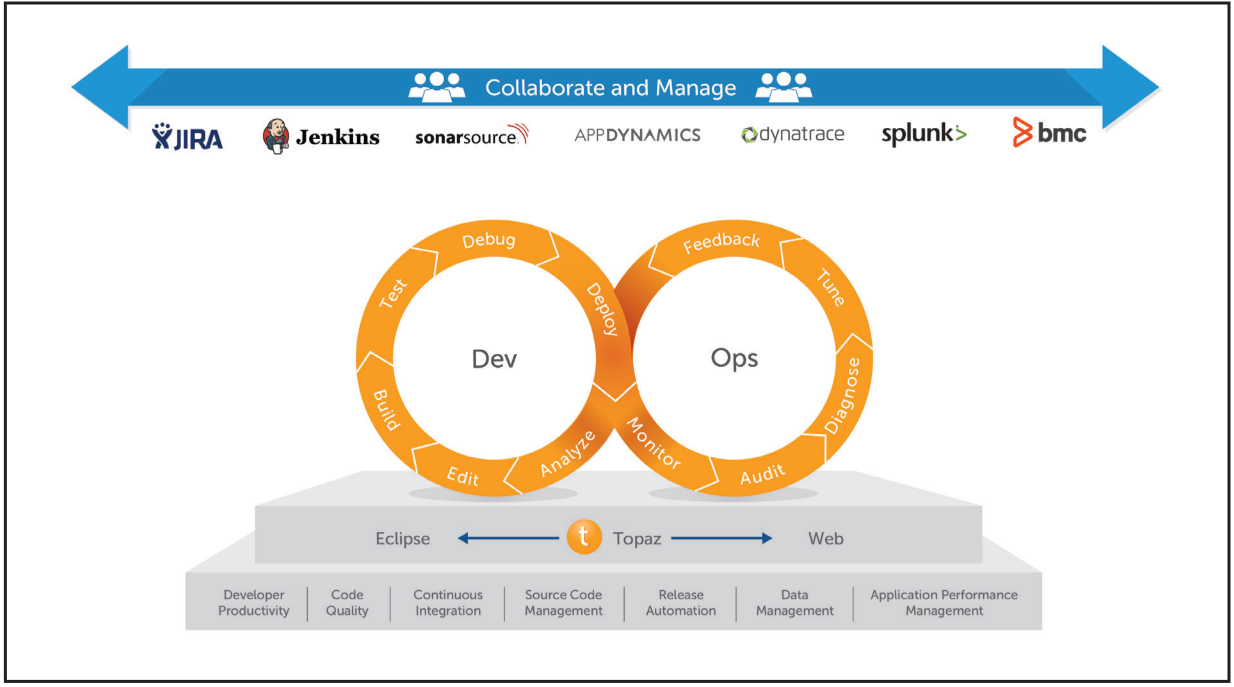

Figure 2. The Topaz Ecosystem

The partners that Compuware has first chosen for the significant steps in enabling DevOps are:

- **Atlassian Jira Software**: Jira is an agile, comprehensive team planning and project management product. It covers everything from full application build-test-release to bug tracking and issue resolution.

- **Jenkins**: Jenkins is a cross-platform, Java-built, continuous build and integration tool. It is open source and hence, free software, released under the MIT license. It comes with a rich set of prebuilt plugins that allow it to integrate with a very wide range of source configuration management (SCM) and test/dev technologies. Builds can be triggered by a commit in a version control system, by scheduling via a cron-like scheduler, by the completion of related builds or even by requesting a specific build URL.

- **SonarSource**: SonarSource tools (e.g., SonarQube and SonarLint) provide continuous code inspection and code quality management. It will identify duplicate code, coding errors and trends, and it will automatically verify that new or rewritten code has been integrated. SonarSource tools provide dashboards and visualizations to enable granular visibility into code quality and management. The significance of this particular capability cannot be understated. Change to code is inevitable, yet without a continuous and automated management process, the system will incur "technical debt," which slows development and release cycles. SonarSource tools do not allow developers to take the bandaid approach by writing inelegant or incomplete code.

- **AppDynamics**: AppDynamics provides application performance management and IT operations analytics for a wide range of distributed environments, embracing all popular programming languages, complex enterprise platforms such as JMS and queuing technologies such as TIBCO, WebMethods, etc.

- **Dynatrace**: Dynatrace provides an alternative application performance management capability. It includes specific user experience capabilities, including deep transaction tracing, synthetic monitoring, real user monitoring and network monitoring (the company was once part of Compuware, and hence, the product is often deployed in Compuware-centric environments).

- **Splunk**: Well known and heavily deployed in enterprise operational environments, Splunk can search, monitor and analyze "machine-generated" data, which generally means log files from applications or devices of any kind in the data center or in the cloud. It is a real-time or near real-time capability which can be configured to generate useful graphs, reports, alerts, dashboards and visualizations.

- **BMC**: In this partnership, Compuware and BMC have integrated BMC's Cost Analyzer for zEnterprise (CAZE) and its MainView with Compuware's Strobe, providing application-aware workload management and cost optimization. This enables users to better understand workload resource consumption and to pinpoint inefficiencies, which can help them to minimize IBM's monthly license charges. This integration is the first of several that the two companies plan as part of a broader partnership.

The common thread running through these best-of-breed partnerships is the creation of a modern, agile DevOps environment both for the mainframe and for cross-platform development. This naturally aligns with underlying design ideas behind Topaz: a culturally disruptive, modern, agile and productive platform.

Re-Imagining the Mainframe

The fundamental economics of the mainframe, coupled with the weight of mission critical applications and data that it harbors, argue strongly for the maximum utilization of the resource and for a natural expansion of its usage. For many mainframe users, this direction is encouraged by the need for application extension – the current extension of mainframe applications to cater for mobile users and the likely future extension of mainframe applications to cater for the rapidly emerging Internet of Things.

The major downsides to the mainframe are its old-fashioned, arcane software environments and its aging skilled workforce, which is gradually disappearing into retirement. Remove these two significant issues, making the platform different only in syntax, and there is no reason why the mainframe ecosystem would not return to the vibrancy it once possessed. Remarkably, this now seems possible.

If that occurs, it will be due in no small way to Compuware's re-imagining of the mainframe as a thoroughly modern computing environment based on agile development tools surrounded by a complementary ecosystem of software that supports both mainframe and cross-platform development and deployment.

In such an environment, the skills issue begins to heal itself. There will be little difference between development for mainframe deployment and development for other platforms. The data coding may be different, as may some of the programming languages, but the continuous development style will be the same, the tools will feel the same and they will have the same modern interfaces. The mainframe culture will gradually change, and its long development cycles and dogmatic resistance to change will gradually disappear.

Compuware has already delivered a good part of its forward vision and demonstrated, by its own release schedules, that it practices what it preaches: continuous, agile development for the mainframe.

Compuware began work on Topaz in mid-2014, consulting first with five large mainframe customers to get a detailed appreciation of what the mainframe opportunity was and to get their input on what a practical solution to their problems might look like. After that, the Compuware development team, organized for agility, went into action, meeting its first delivery target in January 2015. At the time of writing, it is still less than a year since then, and the interest in Topaz continues to rise.

A good example of the customers it is attracting is provided by a large financial services company. It was attracted to Topaz for two specific reasons:

- It had a mainframe project that called for continuous development and deployment, and it was investigating what was possible.
- Skills were an issue. It needed a platform that would be productive for staff who had no Interactive System Productivity Facility (ISPF) experience.

For their immediate requirement, the components that the customer was specifically attracted to were the Topaz Program Analysis and the Runtime Visualizer feature. The project proved successful in the cultural sense – that a move to agile development methods proved possible – and in the practical sense that the desired increase in productivity was achieved.

In our view, these are early days for Topaz. The product will no doubt be adopted for simple productivity reasons. Modern, agile development is faster and a necessary tenet of innovation, and Compuware is providing the tools (and ecosystem), strengthened by the addition of ISPW, to make that possible. Additionally, Compuware has a well-established customer base that will undoubtedly migrate to the new tools with time.

The more intriguing question is how quickly the culture of the mainframe world will change. There are reasons to believe it may occur more swiftly than one might expect. The simple fact is that the old mainframe guard is retiring, and the mainframe culture that it established and helped to preserve for so many years will almost certainly retire with it. The agile culture promoted, enabled and adopted by Compuware could, and in our view should, replace it.

About The Bloor Group

The Bloor Group is a consulting, research and technology analysis firm that focuses on open research and the use of modern media to gather knowledge and disseminate it to IT users. Visit both www.TheBloorGroup.com and www.InsideAnalysis.com for more information.

The Bloor Group is the sole copyright holder of this publication.

Austin, TX 78720 | 512-524–3689