MAKING HISTORY WITH MODERN MAINFRAME SOURCE CODE MANAGEMENT

If your business is focused on continuously improving quality, velocity and efficiency, you're going to win against those that aren't. This is especially true in mainframe-powered organizations, where 72 percent of customer-facing applications rely completely or partially on mainframe processing.

Driving improvements on the mainframe, and in turn throughout the business, requires the transformation of three things: culture, processes and tools. In other words, changing mindsets, implementing modern practices (Agile, DevOps, CI/CD) and replacing outdated technology.

You can't do one or two and be successful; you must do all three. And of these, replacing outdated technology is where every mainframe shop can derive great benefits, as modern DevOps tools enable improvements in efficiency, velocity and quality way beyond those possible with the historic tools. They also empower incoming mainframe professionals who lack the experience of retiring experts by allowing them to get up to speed and exploit the power of the mainframe.

Mainframe source code management is currently a critical area in need of modernization and should be one of the initial tooling changes organizations make when setting out to improve mainframe systems delivery.

Here's a quick history lesson to help you understand where mainframe source code management began, where it went and where it's going today so you understand just how critical this is.

The Early Years of Mainframe Source Code Management

In the '70s, you had version control in mainframe shops and that's really all you needed when it came to managing source code. These tools enabled check in and check out, provided a source repository, let you compare versions and reverse them; that was about it.

By the '80s, new source code management tools supported the entire software development life cycle and provided modern capabilities: automatic compiles and linking, configuration, auditing, etc.

These tools ate up the mainframe source code market through the '90s—despite few enhancements being made to them over the years. Unfortunately, this seemed to aid a strategy the vendors of these tools were keen to leverage.

The Never-ending '90s

Because these tools were so entrenched in mainframe shops around the world and faced little competition, there wasn't much need, in the eyes of vendors, to continuously improve the tools. The steady cashflow they were creating could was readily increased through strict maintenance contracts.

Customers were in a headlock with mainframe source code management. As other areas of the business advanced, mainframe shops were forced to make up for the lack of enhancements to source code management and improve the tools themselves.

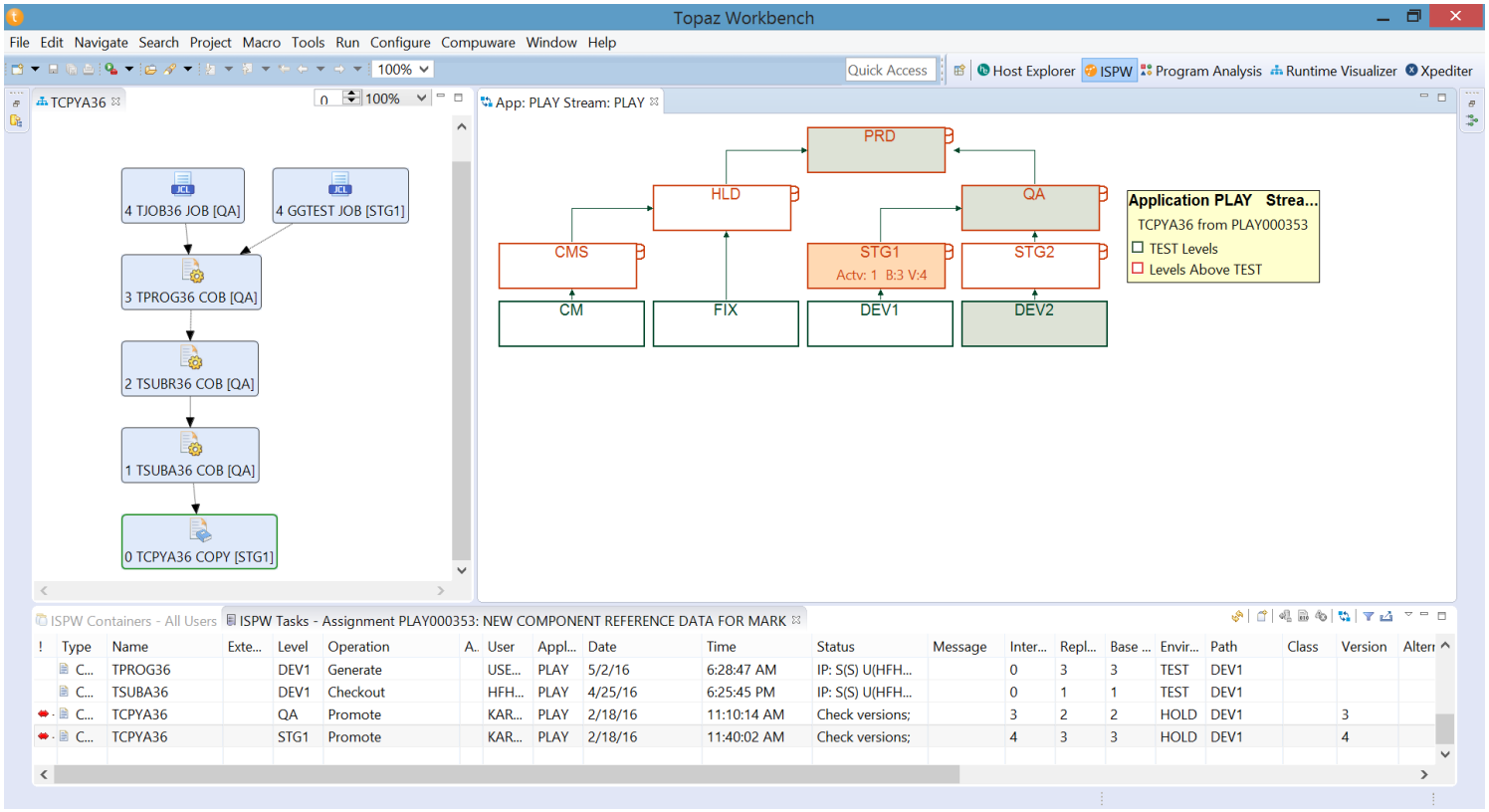
They began adding layers of in-house customizations that were extremely difficult to re-engineer on top of the old SCM tools. Many of these customizations also arose as workarounds to the few administrators who controlled the tools.

Somehow, this cycle never ended. Most organizations are still stuck with '90s-era mainframe source code management tools that present several challenges in a digital world where standards for quality, velocity and efficiency are rising:

- Outdated, customized tools don't integrate well into a DevOps toolchain
- Traditional mainframe source code management admins are retiring, leaving a skills gap
- Incoming mainframe pros are unfamiliar with old UIs and are more productive with GUIs

Modern Mainframe Source Code Management

All this time, in the background was a company with a mainframe source code management tool that took a different perspective on how organizations should manage changes. That tool was ISPW.



Designed to enable agility, ISPW was silently forging a separate path away from the status quo of mainframe source code management tools. Years later, this path has led ISPW to be the only true DevOps SCM tool available to mainframe-powered organizations.

At Compuware, we discovered firsthand that ISPW is the only real option for Agile mainframe source code management when we determined it was necessary to become Agile and begin a DevOps journey in 2014. We ended up dropping our mainframe source code management tool of over 15 years for ISPW. We loved it so much, we bought the company!

As a result, we were able to drive towards our desired state as an Agile/DevOps mainframe company (which we're now continuing as part of BMC), and we're enabling the same for other mainframe-powered organizations that are moving to DevOps. Through inbuilt capabilities and integrations with best-of-breed DevOps tools like CollabNet VersionOne Continuum, Digital.ai Release, Jenkins and SonarSource SonarQube, ISPW enables modern mainframe source code management best practices like:

- Continuous Integration/Continuous Delivery
- Automated, continuous deployment
- Automatic code scanning
- KPI measurements
- Quality gates
- Automated testing
- Integrated code improvement
- Ops IT controls automation
- ITSM integration

These are the modern mainframe source code management capabilities mainframe-powered organizations require to succeed in a digital economy. However, keep in mind that *just* changing your mainframe source code management tool isn't enough for true success; it's only one part of a

larger DevOps toolchain you need to review alongside your people and processes.