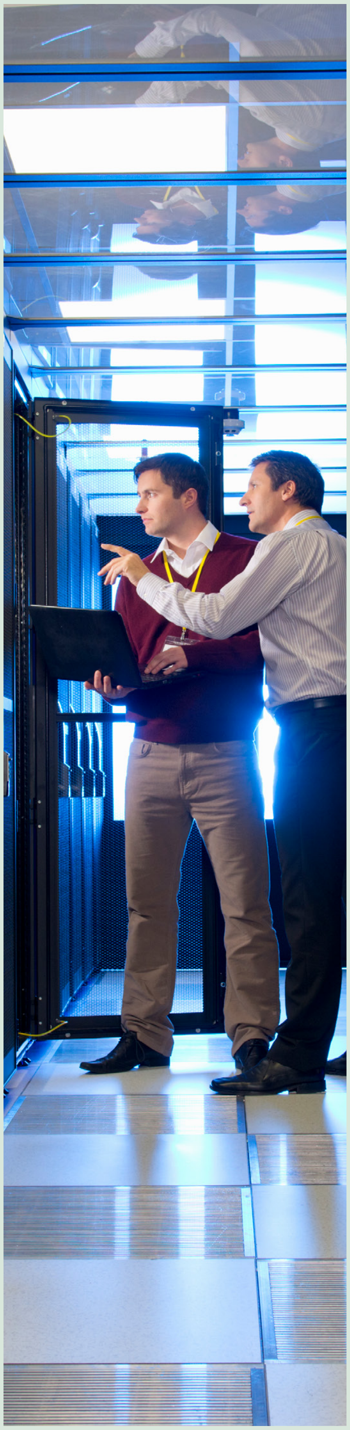
WHITE PAPER



INFOSYS MAINFRAME DEVOPS EXPERIENCES WITH IBM

Infosys®
Navigate your next

Introduction

Around 80% of Fortune 500 companies have their core businesses running on a mainframe. As much as 70% of enterprise data resides on a mainframe. According to IBM, the volume of transactions on mainframes dwarfs the number of Google searches, Twitter tweets, YouTube views, and Facebook likes combined.

Given the imperative for enterprises to innovate faster to stay relevant, many of them have already started on the DevOps journey for their distributed as well as front-end systems of engagement. However, somewhere along the way, they realize that their mainframe applications, which follow the traditional delivery process with long release cycles, are hitting the brakes on the overall speed to market.

Mainframe DevOps can be the solution for speeding up mainframe releases.

The response to this approach often is: "Why DevOps? Why not migrate and modernize?" The reality is that migrating applications off the mainframe is challenging for some enterprises. It involves significant time and effort since all dependent system integrations must be modified after the migration. Due to the lack of SMEs in this space within the organization, confidence about migrating applications that include critical business logic written decades ago is fairly low. So, for organizations that find mainframe modernization undesirable, DevOps on mainframe can bring the required agility.

Mainframe agility is achievable through mainframe DevOps by integrating products that provide automation in every stage of the lifecycle. These also apply relevant gating and thresholds to facilitate zero-touch automation.

Mainframe DevOps: A disruptive or incremental approach?

Mainframe DevOps is different from distributed DevOps and takes a longer time to implement. The approach must be adaptive to get early return on investment (RoI).

Based on the experience Infosys has across several projects, a typical mainframe DevOps journey begins with assessing the IT value stream and doing a time-motion study of where the key bottlenecks are. We recommend a practice-based DevOps approach that identifies processes that must be automated earlier than others to realize maximum and early benefit from the first DevOps intervention. While the target could still be to build a complete CI-CD solution, a practice-based approach works better than a disruptive 'Big Bang'.

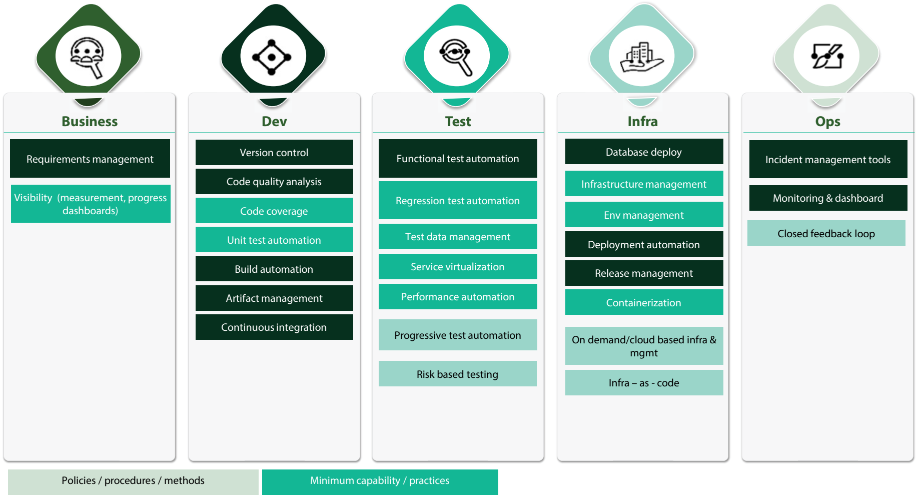


Fig 1: A practice-based approach to Mainframe DevOps

The advantages of a practice-based approach are manifold. It drives early and maximum ROI by bringing in prioritized DevOps interventions for the identified key bottleneck areas across the value stream. Since the approach is progressive, it allows gradual and incremental adoption by the developer community and enterprise groups. Consequently, it allows greater chances of overall success than a disruptive approach. Rather than forming new teams and introducing all the needed capabilities into that team, our approach is to automate DevOps practices incrementally within the respective team boundaries. This prevents major re-alignment of people and disruption in existing ways of working.

The experiences

The mainframe DevOps journey for Infosys customers starts with assessing and identifying the practices to be prioritized. Then, we evaluate and recommend the right-fit product/tool based on a thorough understanding of the application portfolio. To the extent possible, we try to leverage their existing products/tools.

This article examines some interesting lessons learnt during our mainframe DevOps engagements where practice-based DevOps approach was leveraged, leading to end-to-end CI/CD pipelines.

1. Solving environment challenges

Automating environment provisioning can reduce environment cost, environment contention and time to market

A global banking company wanted to achieve operational agility in order to deliver value to customers faster and more frequently. While channels and web layers could move to agile, the business layer on the mainframe, which constituted nearly 70% of the overall technology landscape, followed the traditional waterfall model with limited automation. Infosys value stream analysis uncovered that the primary roadblock was environment contention. Releasing enhancements after development and testing took significant time due to limited and shared environments. Infosys focused on this bottleneck as the first intervention with the aim of making more environments available on-demand.

After deliberating multiple approaches to provision environments on-demand (including additional infrastructure on the mainframe), the IBM z Development and Test Environment (zD&T) was finalized as the most suitable solution for spinning up the virtual development and test environments off-host. This would not only cut down the CPU utilization of the mainframe but drastically reduce the cost of additional mainframe infrastructure. It would also minimize the time taken to provision a new environment and resolve the environment contention.

An environment provisioning strategy was built around the IBM zD&T product. An RHEL server was used as the target zD&T environment and the customer mainframe landscape was mirrored. As the customer zOS infrastructure was too large to be accommodated on the zD&T, it required careful analysis and identification of a subset to be taken to the virtualized environment. The solution was finally able to mimic the customer mainframe environment on IBM zD&T virtualized environments.

Infosys further proposed and implemented an 'environment on demand' wherein a golden source would be created and stored centrally. It would be cloned as per demand to provision an environment container. This ensured that environments stayed in sync. Infosys and IBM joint innovation teams worked with the bank's infrastructure team to build a strategy for creating and maintaining off-host testing environments.

2. Improving developer productivity with IDE

Engaging developers to use a modern IDE increases efficiency and reduces errors

For a large financial services customer headquartered in the US, the practice-based approach unearthed developer productivity issues when working with mainframe green screens (ISPF panels).

Infosys recommended moving out of the green screens and adopting an integrated development environment (IDE) like IBM Developer for z (IDz). This is a modern development environment with features like integrated code analysis and easy review. It is proven to increase productivity in the developer workflow. The main challenge here was resistance by the workforce in moving away from the traditional ISPF screens.

To combat resistance from the developer community and enable active adoption of IDE, a custom integration solution between IBM IDz and the source code management tool (ChangeMan) was implemented. The Menu Manager feature of IBM IDz was used to integrate all the ChangeMan functionalities. In this way, code check-in, build, packaging, code promotion and other functions could be performed right from the developer desktop. This custom solution resulted in cost savings of procuring an external plugin and significantly boosted the developer's IDE usage.

Features like a thorough impact analysis, automated unit testing and static code quality checks further enhanced the developer experience, cementing a positive foundation for DevOps adoption.

3. Building code quality

Shift-left of code quality in mainframe applications using enterprise-wide code quality tools

A US-based financial company had been using SonarQube as its code quality tool. They were keen to use the same tool for their mainframe applications as well. As SonarQube's default code quality rules were insufficient for full coverage, Infosys created custom code quality rules for COBOL.

IBM IDz is an eclipse-based IDE and was able to integrate seamlessly with SonarQube. This increased the developers' productivity, allowing them to run their code through preset coding standards, thereby detecting defects much earlier in the development lifecycle and improve code quality.

4. Automate unit testing

Automate unit testing to detect issues right at the developer's desktop

Even as the above finance company was reaping the benefits of increased productivity through a modern IDE, Infosys brought in the next DevOps practice of automated unit testing at the developer's desktop by introducing IBM zUnit. Developers could create test cases for online and batch testing right from their IDz and store it on the mainframe. Existing test cases could be reused to create additional test cases, thus significantly reducing cost of quality effort. Executing these unit test cases resulted in a shift-left of defect detection. Additionally, it helped identify how extensive the test suite was in terms of the code coverage. This way, developers could track if all their lines of code had been tested successfully and, if not, they could create more test cases to increase the code coverage.

5. Orchestration of build

Orchestrate the developer workflow in the CI pipeline through automation

For the same customer referred to above, Infosys orchestrated a developer's workflow with IBM IDz to include a check-in to the source code management tool. Additionally, code quality was checked using SonarQube, and unit testing and code coverage with zUnit. Infosys then automated these processes in a continuous pipeline using Jenkins as an orchestrator. Besides integrating all these processes into an automated pipeline, quality gating was also brought in as part of continuous integration to abort the build if it did not meet the coding standards or if there were any defects that were revealed by the testing.

6. Test automation

Enable continuous validation of functional test cases

For a UK-based bank, Infosys introduced service virtualization using IBM Rational Integration Tester (RIT). Another Infosys customer, a US bank, adopted IBM Rational Functional Tester (RFT) for functional test automation. Infosys brought in its expertise and services to implement RIT and RFT and integrate them into Jenkins, thus enabling continuous testing in the application software delivery process. By creating the test cases in a central test repository GIT, the UK bank was also able to achieve a unified repository source.

7. Deployment automation

Implement deployment automation as part of the end-to-end DevOps solution

One of our US-based banking clients was plagued with issues of manual and error-prone deployments. After deployment was partially automated through custom scripts, Infosys offered its expertise to help the bank adopt IBM Rational Team Concert (RTC) as the build tool and IBM Urban Code Deploy (UCD) as the deployment tool. This created automated processes that deploy, upgrade, roll back, and uninstall applications.

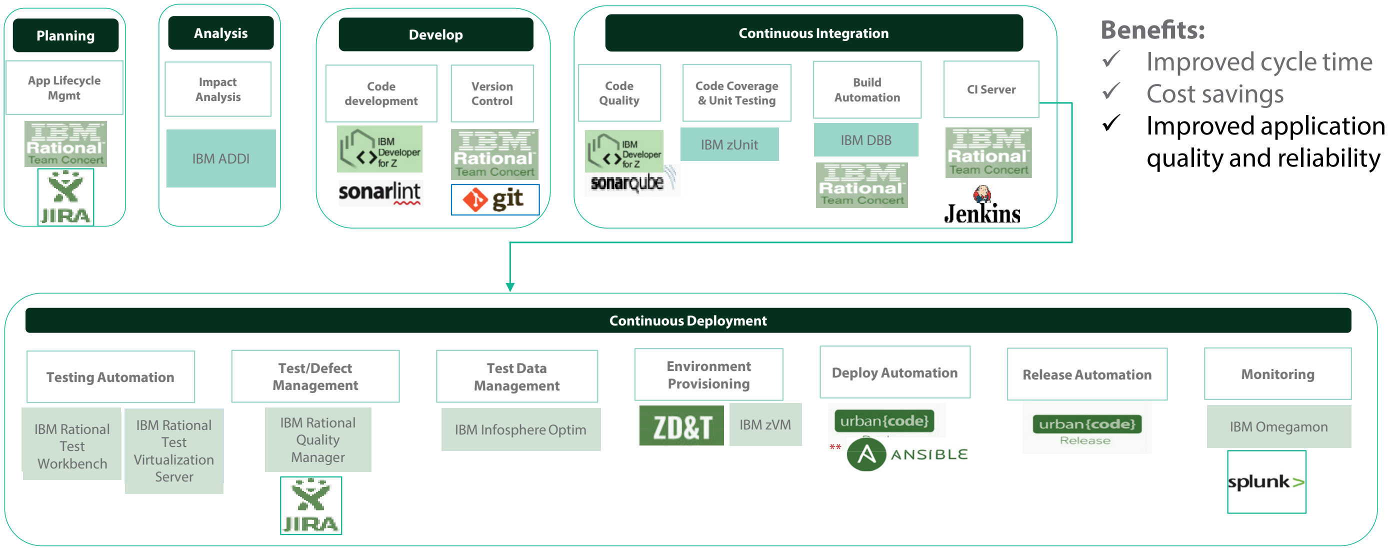
8. Modernize the source code management

Use a modern source code management tool to advance in the mainframe DevOps journey

Many enterprise customers seek to migrate away from legacy SCM systems to a more modern SCM as these support parallel development and enable standard processes/capabilities across mainframe and distributed portfolios. Moving to GIT is one such viable approach.

After migrating from traditional SCM like Endevor, ChangeMan, etc., enterprises must integrate build-and-deploy features. These features are inbuilt in traditional SCMs but are absent in modern SCMs like GIT. Infosys is liaising with IBM, our strategic partner, to adopt IBM's build solution (GIT-DBB). We are also working with a few customers to use IBM DBB capabilities to build traditional z/OS applications such as COBOL and PL/I and to provide automation capabilities based on a modern scripting language that can be used on z/OS and that can be triggered from continuous integration servers like Jenkins.

IBM products based Mainframe DevOps solution



Benefits:

✓ Improved cycle time

✓ Cost savings

✓ Improved application quality and reliability

Fig 2: Infosys mainframe DevOps solution framework based on IBM suite

Conclusion

Over the last few years, Infosys has been working to create seamless mainframe DevOps journeys through approaches that help customers realize value early on despite long implementation times. Our approach begins with thorough due diligence, assessment, strategizing, tool evaluation, piloting, and creating a roadmap for mainframe DevOps. Zero touch orchestration and automated gating is enabled through various tools. Finally, the applications are onboarded and teams align with mainframe DevOps through continuous improvement. Mainframe DevOps helps organizations improve speed-to-value, developer productivity, quality, and system reliability while reducing wastage and cost.

About the Authors

Monika Gupta

Mainframe DevOps Architect, DevOps Practice, Infosys

Prasanna Ghanekar

Engineering Head, DevOps Practice, Infosys

For more information, contact askus@infosys.com

Navigate your next