

Empowering Mainframe Development with Git: Enterprise Benefits and Opportunities



Arnal Dayaratna
Research Vice President,
Software Development, IDC

Table of Contents



CLICK BELOW TO NAVIGATE TO EACH SECTION IN THIS DOCUMENT.

Executive Summary	3	Risks of Adopting Git for Mainframe Development.....	10
Benefits of Using Git for Mainframe Development.....	4	Advice for Overcoming Risks and Challenges	11
Developer Velocity and Agility: Git Complements Traditional Mainframe Software Configuration Management to Make Development Faster.....	5	Choosing a Branching Strategy.....	12
Collaboration: Git Enhances Opportunities for Developers to Work Together	6	Summary of Essential Guidance	13
Modernization: Git Facilitates Use of Modern Development Tools for the Mainframe.....	7	Case Study #1: North American Health Insurer Integrates Git with Traditional Mainframe Source Code Management Tools	14
Talent: Git Expands the Universe of Mainframe Developers	8	Case Study #2: A European Bank Uses Git to Streamline Mainframe DevOps Implementation.....	15
Challenges to Adopting Git for the Mainframe.....	9	About the IDC Analyst.....	16
		Message from the Sponsor.....	17

Executive Summary



Git is the de facto standard for version control for contemporary software development.



The mainframe continues to be strategically important to organizations because of its large-scale transactional processing capabilities, reliability, security, scalability, and durability.



The use of Git for mainframe development streamlines the integration of modern development and DevOps tools and accelerates developer velocity and agility.



Organizations that use mainframes need to adopt a hybrid approach to software configuration management (SCM) by supplementing their use of traditional mainframe SCM tools with Git.



Git is not the only SCM technology for mainframe development. Mainframe developers may use mainframe-specific SCMs alternatively or in conjunction with Git.

Mainframes are central to contemporary application development.



More than **10%** of all enterprise applications are primarily deployed on the mainframe.

Source: IDC's Modern Software Development Survey, August 2024

Benefits of Using Git for Mainframe Development

DEVELOPER VELOCITY AND AGILITY



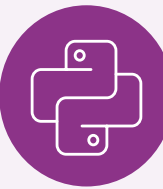
Development is accelerated because developers can access source code and work in parallel with their colleagues.

COLLABORATION



Collaboration is enhanced because developers can access source code in conjunction files with their peers, thereby enabling developers to work together on code development.

MODERNIZATION



Opportunities to integrate with modern development tools such as IDEs, CI/CD tooling, automated testing, and security analysis of code tools are enhanced.

TALENT

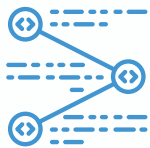


The universe of developers that can work on the mainframe is expanded to include Gen Z and Gen Y developers.

DEVELOPER VELOCITY AND AGILITY



Git Complements Traditional Mainframe Software Configuration Management to Make Development Faster



Mainframe developers have access to the complete code base via Git, which enables them to manage the changes offline and create branches and merge them whenever required.



This ability to work in parallel with other developers means that development is no longer monolithic, and that developers can work concurrently on a multitude of application features and functionalities.



Git absolves mainframe developers of the need to track what other developers and team members are working on, and thereby reduces manual work associated with development-related planning and resource allocation.



Git notifies developers of code conflicts that arise as a result of merges and provides tools to adjudicate merge-related conflicts.

COLLABORATION



Git Enhances Opportunities for Developers to Work Together



Git facilitates parallel development and democratizes access to mainframe development, thereby amplifying opportunities for development-related collaboration.



Experienced mainframe developers that use traditional software configuration and library managers can work alongside Gen Z and Gen Y developers that use modern development languages, IDEs, and tools.



Git standardizes development processes for mainframe and non-mainframe development.



As a result, **contemporary developers can use a familiar and consistent set of development tools** for developing on the mainframe.

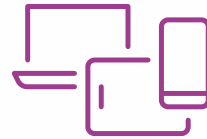
MODERNIZATION



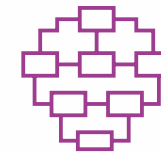
Git Facilitates Use of Modern Development Tools for the Mainframe



Git streamlines the use of modern development languages and frameworks for the mainframe.



Git enables developers to perform cross-platform development that leverages the mainframe with modern tools such as mobile and cloud-based tools and infrastructures.



Git provides an on-ramp to the implementation of an enterprise-wide DevOps strategy by providing an infrastructure that integrates with popular CI/CD tools.



Organizations can choose whether to pursue a Git-native modernization approach, or a strategy that combines the use of Git with traditional mainframe SCM tools.



Git Expands the Universe of Mainframe Developers



Retiring mainframe developers are in the process of handing over responsibility for mainframe development to a newer generation of developers who are trained on modern development tools such as Git, VS Code, GitHub Codespaces, and CI/CD tools.



Git empowers modern developers who have little to no experience with mainframe development to develop for the mainframe.



Git expands the ability of organizations to recruit and retain developers for mainframe-based development.

Challenges to Adopting Git for the Mainframe



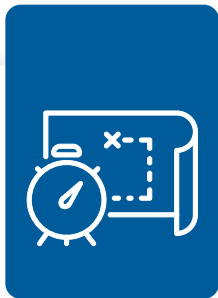
1

Organizations need to master communicating the value of using of Git to veteran mainframe developers.



2

Because of the importance of SCM to application development, IDC recommends that **the adoption of Git proceed incrementally** to ensure that developers who are used to traditional mainframe SCM tools are properly trained on the use of Git for the mainframe.



3

Organizations need to develop plans for accommodating the coexistence of traditional SCM tools with Git, in conjunction with long-term plans for standardization on Git.

Risks of Adopting Git for Mainframe Development



1

Resistance from experienced mainframe developers has the potential to derail adoption of Git for mainframe development.



2

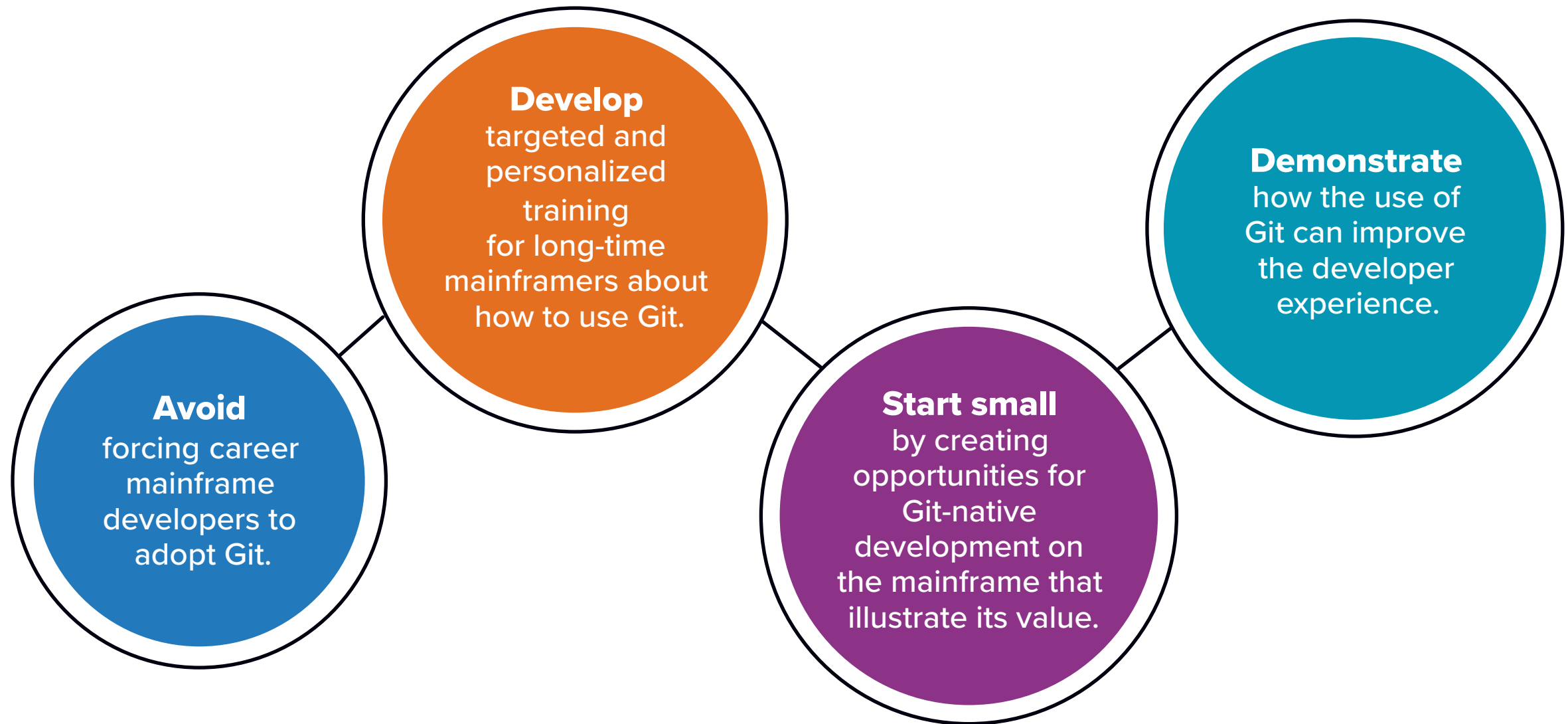
Excessively rapid implementation of Git for mainframe may lead to code quality issues. The newer generation of developers may lack experience with mainframe development.



3

Developers need to be cognizant of the possibility of security risks specific to applications developed and deployed on Git.

Advice for Overcoming Risks and Challenges



Choosing a Branching Strategy

Consider these core principles when choosing your branching strategy:

Principle	Explanation
 Minimize branching	Branching creates isolation but can lead to excessive merging, which, ultimately, slows development; resist the temptation to model traditional lifecycles.
 Practice “ build once/test many”	When merging to another branch, unless you rebuild, the code may not match the binaries; however, rebuilds invalidate previous test. To mitigate this, deploy builds to multiple test environments.
 Isolate changes early in the lifecycle	Use feature branches to isolate changes and enforce quality gates (e.g., code reviews, pre-integration scans).
 Avoid long-living feature branches	Long-living feature branches tend to diverge from the main codebase, leading to complex merges, higher risk of conflicts, and delayed integration and testing; using short-lived feature branches improves quality with quality gates (e.g., pull/merge requests).

Note: Exercise caution using Git Flow because it results in excessive branching and complexity, resulting in reduce developer productivity and increased likelihood of escaped defect.

Summary of Essential Guidance



Develop a pragmatic modernization strategy to drive DevOps adoption.



Educate mainframe developers on the value and use of Git but do not force them to use it.



Consider using an intermediary platform that integrates Git code with code stored in mainframe SCM tools.



Adopt an incremental approach to the use of Git for the mainframe that progressively increases the share of development that leverages Git.

Case Study #1:

North American Health Insurer Integrates Git with Traditional Mainframe Source Code Management Tools

A large health insurer examined options for using Git in conjunction with traditional mainframe SCM and release management tools. The company demonstrated the viability of a product that automates the integration of code stored in a traditional mainframe library manager with Git.

The company illustrated the efficacy of hybrid development that leverages Git and traditional mainframe SCM tools.

The bridging of code between traditional mainframe SCM tools and Git democratizes access to the mainframe by empowering developers who have little to no experience with mainframe development to start working on mainframe development immediately.

Case Study #2:

A European Bank Uses Git to Streamline Mainframe DevOps Implementation

A European bank used Git to implement DevOps for mainframe development, inclusive of CI/CD pipelines built using Jenkins.

The bank noted the following benefits:

Accelerated development
because developers were able to work in parallel

Shortened feedback loops between business owners and developers due to the ability of developers to rapidly prototype solutions for feedback from the business

Increased automation
empowered developers to spend more time coding

Enhanced collaboration
between mainframe and non-mainframe developers

Faster time to market

Improved customer experience due to more frequent updates to software

About the IDC Analyst



Arnal Dayaratna
Research Vice President,
Software Development, IDC

Dr. Arnal Dayaratna is research vice president of Software Development at IDC. Dayaratna focuses on software developer demographics, trends in programming languages and other application development tools, and the intersection of these development environments and the many emerging technologies that are enabling and driving digital transformation. Dayaratna’s research examines how the changing nature of software development relates to broader trends in the technology landscape.

[More about Arnal Dayaratna](#)

Message from the Sponsor



Broadcom offers the most comprehensive development toolset for z/OS-based applications, including seamless access to Git. Whether adopting your enterprise orchestration layer or continuing to orchestrate code changes with Endeavor, there is real value in Git adoption, especially in talent recruitment and onboarding.

To realize this value, we offer a "Git Adoption Kit" that drives continuous integration with Git-native development, as well as the award-winning Endeavor Bridge for Git, which provides a Git experience without the risks of moving code.

This Kit offers a risk-managed approach to modernization, enabling adopters to realize the many benefits of Git tooling while mitigating the costs and risks associated with potentially disruptive change. This combination, along with new Endeavor-native features for parallel development, facilitates modernization across your z/OS application portfolio.

IDC Custom Solutions

This publication was produced by IDC Custom Solutions. The opinion, analysis, and research results presented herein are drawn from more detailed research and analysis independently conducted and published by IDC, unless specific vendor sponsorship is noted. IDC Custom Solutions makes IDC content available in a wide range of formats for distribution by various companies. This IDC material is licensed for external use and in no way does the use or publication of IDC research indicate IDC's endorsement of the sponsor's or licensee's products or strategies.



IDC Research, Inc.
140 Kendrick Street, Building B, Needham, MA 02494, USA
T +1 508 872 8200

[idc.com](https://www.idc.com)

[in @idc](https://www.linkedin.com/company/idc)

[X @idc](https://twitter.com/idc)

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications, and consumer technology markets. With more than 1,300 analysts worldwide, IDC offers global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries. IDC's analysis and insight helps IT professionals, business executives, and the investment community to make fact-based technology decisions and to achieve their key business objectives.

©2025 IDC. Reproduction is forbidden unless authorized. All rights reserved. CCPA