PULSEMETER

Techstrong Group
a Futurum company

The Orchestration Challenge in Mainframe Software Delivery

SPONSORED BY

bmc

by Guy Currier, Analyst, Futurum Group

Mainframe software modernization is advancing on several fronts at once, with no single obstacle standing out. Goals related to speed, AI, and increased control are pulling in the same direction but aren't yet moving together.

Research Insights

Mainframes run much of the core transaction processing that banks, insurers, retailers, and government agencies depend on. These applications change constantly, and the processes used to deliver those changes face the same pressure for speed, automation, and AI that has reshaped delivery everywhere. A Techstrong PulseMeter poll of 145 practitioners and decision-makers in mainframe software delivery suggests that the pressure to modernize is very much being felt; that modernization itself is well underway; and that the work is harder to sequence than a simple maturity model would imply.

Several delivery models at once

According to the poll, the most frequent mainframe software delivery environment uses manual or legacy delivery processes (47%). Automated CI/CD in limited areas followed at 34%, a mostly standardized CI/CD platform at 30%, and hybrid environments spanning multiple platforms and toolchains at 26%. Only 17% of organizations report having an intelligent, governed software delivery platform.

Organizations commonly run more than one of these environments at once—about 1.5 on average. Mainframe delivery is not a single maturity curve, with each organization placed at one point along it. A single enterprise can run standardized pipelines for some applications while others still move through manual handoffs. More accurately, mainframe delivery is typically a portfolio of approaches, with legacy processes remaining the most common element.

Key Takeaways

- **Mainframe software delivery is modernizing, but unevenly.** Manual and legacy processes remain common even as organizations adopt CI/CD, governed platforms, and more modern delivery practices.
- **Security and compliance are both a leading priority and a leading barrier.** In 51% of organizations, security, governance, or supply chain is the hardest part of expanded delivery responsibility, leaving it open whether these needs are met inside the delivery process or around it.
- **AI's strongest foothold is developer assistance, and use has outpaced strategy.** While 86% of organizations use AI somewhere in mainframe delivery, only 30% name AI-assisted delivery as a top strategic outcome. Use is highest for code generation or developer assistance (57%) and drops to 25% for deployment automation.
- **Organizations most often expect gains to come from automation, AI, and better pipeline integration.** In 71% of organizations, one of those three areas is the capability most expected to accelerate progress over the next 12–18 months.
- **The most common challenges lie between systems and functions.** Security integration, coordination across tools and teams, and operationalizing AI rank closely together, pointing to an orchestration problem.

Control as both goal and constraint

On strategy, organizations lean toward speed and automation. Accelerating software delivery (47%) and automating build, test, and deployment workflows (45%) lead the list of outcomes that matter most, with integrating security and compliance into delivery close behind at 38%. Supporting AI-assisted delivery (30%) and providing governance and auditability (29%) form a second tier.



The barriers mirror those priorities. Legacy applications or development processes are the most common constraint (42%), followed by security, governance, or compliance requirements (37%). Security and compliance, in other words, rank as both a top priority and top barrier at nearly the same rate: organizations are trying to build them into delivery, and they're also significant in slowing delivery down. That may sound contradictory, but it's not: where mainframes run regulated, high-consequence workloads, control requirements are not optional. The open question is whether these needs are satisfied inside the delivery process or around it.

After these top two barriers, other constraints are crowded into a middle tier: fragmented tools and disconnected workflows (27%), skills or resource constraints (26%), budget constraints (26%), limited lifecycle automation (25%), and difficulty integrating AI (23%) all show similar frequency. Practitioners locate the friction mostly in systems, processes, and resources rather than in attitudes toward change. Only 9% of organizations face no significant limit on their progress.

AI at the keyboard, less so in the pipeline

AI is already widespread in mainframe delivery. Eighty-six percent of organizations use it in at least one area, and 57% use it for code generation or developer assistance, well ahead of any other application. Use drops to 38% for automated testing, 35% for monitoring and incident response, and 34% for build and pipeline optimization, then to roughly a quarter for security and compliance automation (26%) and deployment automation (25%). Organizations that use AI apply it in about two and a half areas on average.

The pattern follows the path of least integration. Developer assistance can be adopted by an individual or a team with limited dependence on the rest of the pipeline. Testing, deployment, and compliance automation require AI to work within shared systems, approval chains, and evidence requirements, which is harder in the mixed environments described above.

A gap also exists between use and strategy. While 86% use AI somewhere, only 30% of practitioners count the support of AI-assisted delivery as one of their top strategic outcomes. One reading of this is that AI adoption in mainframe delivery has so far been driven from the bottom up, through developer tools, rather than directed as a delivery objective. That would help explain why 23% of organizations are held back by difficulty integrating AI into delivery workflows and 21% say that operationalizing AI is their single most challenging area (see the next section). Using AI and operationalizing it are proving to be quite different problems.

No single hardest problem

As delivery responsibilities expand beyond build, test, and deploy, organizations are split almost evenly on which area is most challenging: integrating security throughout the lifecycle (24%),

coordinating delivery across tools, teams, and environments (23%), and operationalizing AI (21%).

The near-tie among these top three is telling. Each describes work that crosses a boundary: between development and security, between teams and toolchains, between a model's output and a governed release. Taken together, the three control-oriented options (security, governance, and supply chain) are the top concern for 51% of organizations.

Where organizations expect progress and where they don't

Looking 12 to 18 months ahead, organizations expect acceleration to come mainly from greater automation across the lifecycle (26%), broader use of AI-assisted workflows (25%), and better integration across tools and platforms (20%). Together, those three are the leading expectations for 71% of organizations.

Meanwhile, only 11% of organizations expect stronger governance, policy, and compliance automation, and just 4% anticipate improved software supply chain visibility. Governance and compliance carry real weight elsewhere: security, governance, or compliance requirements rank second among barriers (37%), and 15% of organizations find meeting these requirements as their most challenging area. Many appear to treat compliance as a condition to satisfy rather than as a lever for speed. Whether that view holds as automation and AI extend further into regulated delivery workflows will shape how quickly the expected gains arrive.

Software supply chain concerns rank low across the board: 20% of organizations treat them as a strategic priority, 12% as the top challenge, and 4% as the most likely accelerator. These may be concerns that have not yet reached mainframe delivery teams with the same force as elsewhere.

Analyst View

Mainframe software delivery is in the midst of a modernization effort. It is focused on extending established software delivery practices to the platform, including automation, continuous integration, AI assistance, and embedded controls. Organizations' priorities point in that direction: speed and automation lead, and 71% expect the next gains to come from more automation, more AI, or better integration.

The sequencing of these improvements is the open question. Nearly half of organizations primarily have manual or legacy delivery processes in their environment, and legacy applications are the most common constraint. Extending AI into testing and deployment, automated policy enforcement, and connected toolchains assumes a delivery pipeline that is already automated and instrumented. Where that pipeline does not yet exist, AI has little to attach to beyond the developer's editor.



This may be why AI adoption looks broad but shallow, concentrated at the point of code creation but thinning out as software moves toward release.

Security, governance, and compliance surface repeatedly: as a strategic priority, as the second most common barrier, and as half of the challenges today in mainframe software delivery. Yet few organizations are using AI or automation to address automate governance and compliance. For workloads as regulated and consequential as those typically found on the mainframe, modernization that treats controls as an external checkpoint seems

likely to reproduce the friction organizations already face. Modernization that makes policy enforcement, auditing, and attestations products of the pipeline itself could reduce that friction.

Interest in modernization is not in doubt. The issue in mainframe software delivery is when modernization efforts will become holistic and integrated. Can automation span the legacy and modern toolchains that coexist in most environments, rather than serving only the newer ones? Can security and compliance checks run inside delivery workflows and produce the evidence auditors need? Can AI-generated code, tests, and pipeline changes be reviewed and traced with the same rigor as human work? Can coordination across teams and tools be managed deliberately rather than left to handoffs? The variable to watch is whether organizations can orchestrate automation, AI, and control as one system, or whether they will continue to advance each on its own track.

Recommendations

- **Establish a modernization baseline before choosing the next initiative.** Document where manual or legacy processes remain, where CI/CD is standardized, and where delivery spans multiple tools and environments. Because most organizations run a mix of delivery models at once, a baseline distinguishes foundational gaps from areas ready for more advanced automation or AI and gives later improvements something to measure against.

- **Plan for orchestration across tools, teams, and environments.** The biggest challenges lie at organizational boundaries, not within any single stage. Define ownership, handoffs, policy enforcement points, and shared visibility across the lifecycle so gains in one area don't create new friction elsewhere.

- **Prioritize automation where recurring manual effort is highest.** Start with parts of the lifecycle where legacy processes, disconnected workflows, or manual approvals consume the most time repeatedly, and where automation's impact can be measured directly.

- **Treat AI as a delivery-system capability, not only a developer tool.** Developer assistance is established. Extending AI into testing, pipeline optimization, monitoring, and deployment depends on integration, governance, and review processes that the pipeline itself must supply. Define how the organization will review, trace, and approve AI-produced code, tests, and changes before expanding their use.

- **Address integration as a first-class modernization requirement.** Better integration ranks among the top three expected accelerators, and fragmented tools rank among the most common barriers. Assess how mainframe delivery connects with the repositories, pipelines, and platforms used for the rest of the application portfolio.

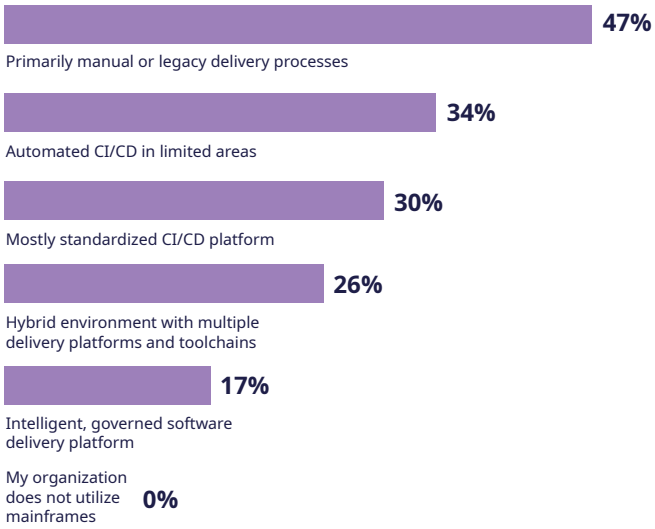
- **Build security and governance into delivery workflows, not around them.** Control requirements are both a leading priority and a leading constraint. Make policy checks, approvals, and audit evidence automated outputs of the pipeline, and revisit the assumption that governance automation is rarely the first lever for delivery speed.

- **Measure progress in operational outcomes, not modernization activity.** Track lead time for changes, automation coverage, manual effort, exception and rework rates, policy compliance, and cross-tool handoffs. Outcome measures distinguish meaningful progress from activity and give teams a defensible basis for deciding what to scale.

Research Results

Current Mainframe Software Delivery Environment

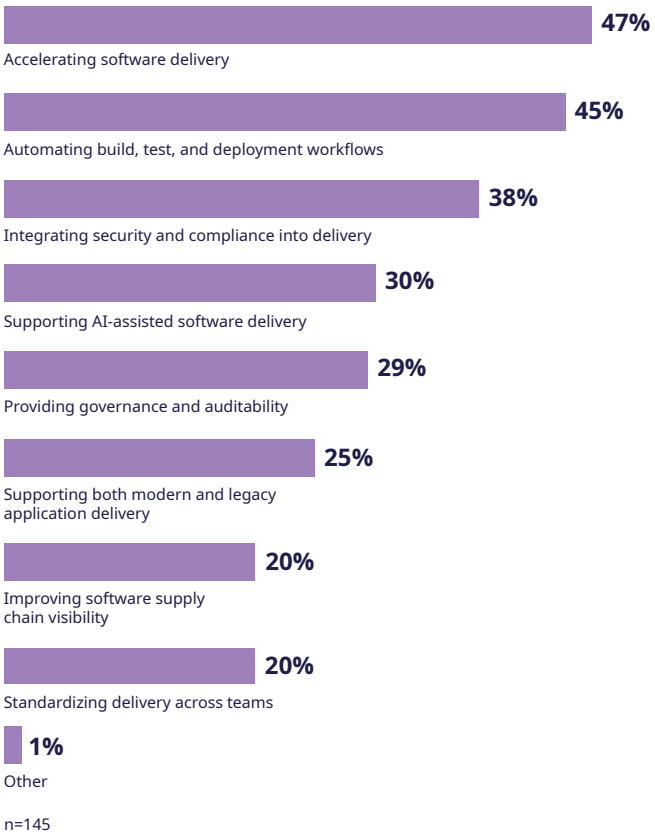
How would you best describe your organization's current software delivery environment(s) on mainframe computers? (Select all that apply)



Mainframe software delivery combines legacy and modern approaches. Primarily manual or legacy processes are the most frequently selected environment (47%), followed by automated CI/CD in limited areas (34%), a mostly standardized CI/CD platform (30%), and hybrid environments spanning multiple platforms and toolchains (26%). Only 17% report an intelligent, governed delivery platform. Because organizations could select more than one environment and averaged about 1.5 selections each, the results describe the range of approaches in use rather than mutually exclusive maturity stages.

Top Software Delivery Strategy Priorities

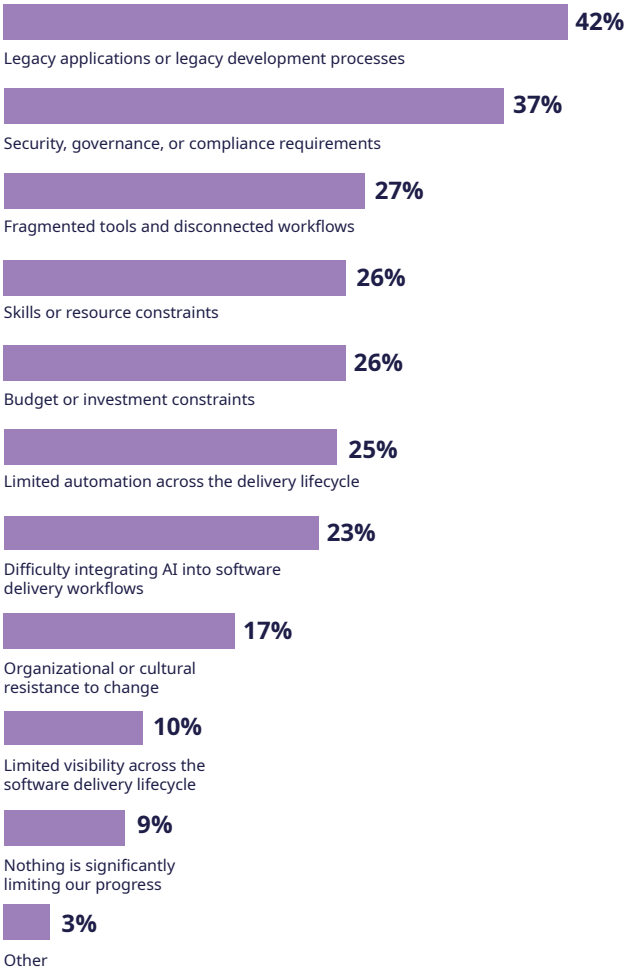
Which outcomes are most important to your organization's software delivery strategy today? (Select up to three)



Speed and automation lead, with security close behind. Accelerating delivery (47%) and automating build, test, and deployment workflows (45%) are the most frequently cited outcomes, followed by integrating security and compliance into delivery (38%). Supporting AI-assisted delivery (30%) and governance and auditability (29%) form a second tier, and a quarter (25%) prioritize supporting both modern and legacy application delivery. Organizations define delivery performance to include consistency and control, not speed alone.

Top Barriers to Mainframe Software Delivery Progress

What, if anything, is limiting your organization's mainframe software delivery progress? (Select up to three)

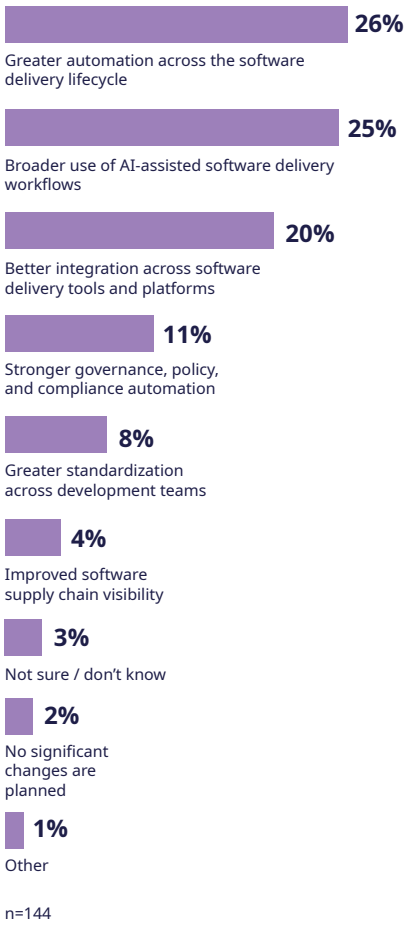


n=145

Legacy environments and control requirements create the strongest drag. Legacy applications or development processes (42%) and security, governance, or compliance requirements (37%) are the most-cited barriers. Five more fall between 23% and 27%: fragmented tools (27%), skills or resources (26%), budget (26%), limited lifecycle automation (25%), and difficulty integrating AI (23%). Cultural resistance (17%) and limited visibility (10%) rank lower, and only 9% report no significant limitation. The distribution points to cumulative friction rather than one blocker that can be removed in isolation.

Top Accelerators of Mainframe Software Delivery Progress

Which of the following do you expect is most likely to accelerate your organization's mainframe software delivery progress over the next 12–18 months? (Select one)

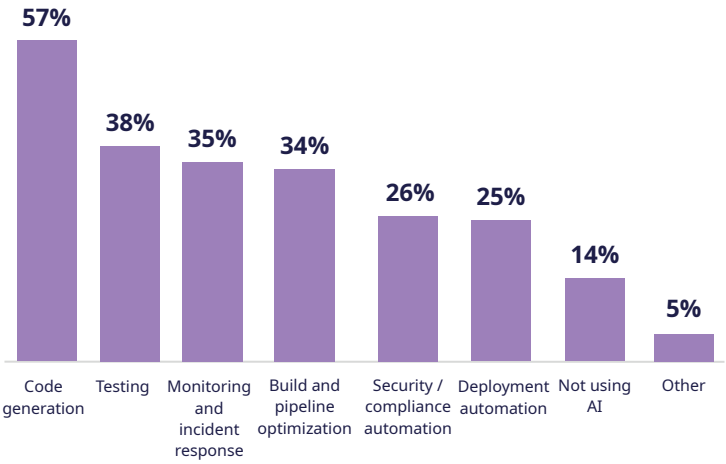


n=144

Organizations expect gains from automation, AI, and integration. Greater lifecycle automation (26%) and broader use of AI-assisted workflows (25%) are virtually tied, with better tool and platform integration third (20%). Together they account for 71% of responses. Governance and compliance automation (11%), team standardization (8%), and supply chain visibility (4%) trail well behind. Organizations expect progress from making the delivery system more connected and automated, not from optimizing any single function.

Current AI Use Across Mainframe Software Delivery

In which areas of your mainframe software delivery environment are you currently using AI? (Select all that apply)

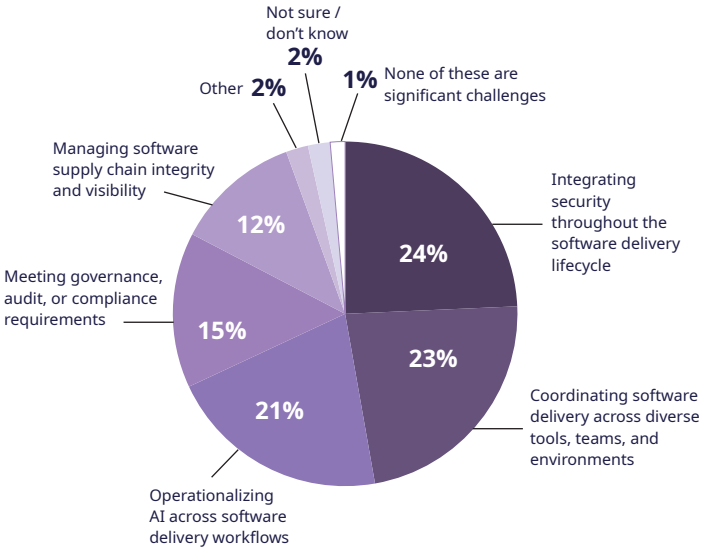


n=145

AI use is broad but concentrated among developers. Fifty-seven percent use AI for code generation or developer assistance, ahead of automated testing (38%), monitoring and incident response (35%), and build and pipeline optimization (34%). Use falls to about a quarter for security and compliance automation (26%) and deployment automation (25%), and 14% report no current AI use. Extending AI from individual productivity into testing, operations, and deployment requires integration across more systems and governance boundaries, which may explain the drop-off.

Most Challenging Areas in Mainframe Software Delivery

As software delivery responsibilities expand beyond traditional build, test, and deploy, which of the following areas is the most challenging for your organization today with respect to mainframe software delivery? (Select one)



n=144

The hardest challenges sit at the boundaries. Integrating security across the lifecycle (24%), coordinating delivery across tools, teams, and environments (23%), and operationalizing AI (21%) are nearly tied, followed by governance, audit, or compliance (15%) and supply chain integrity (12%). No single challenge dominates. Each of the top three requires delivery to span organizational and technical boundaries that traditional build-test-deploy pipelines were not designed to cross, suggesting the problem is increasingly one of orchestration.

This Techstrong PulseMeter is sponsored by BMC.

To learn more about BMC, visit [AI for Mainframe Development - BMC Software](#).