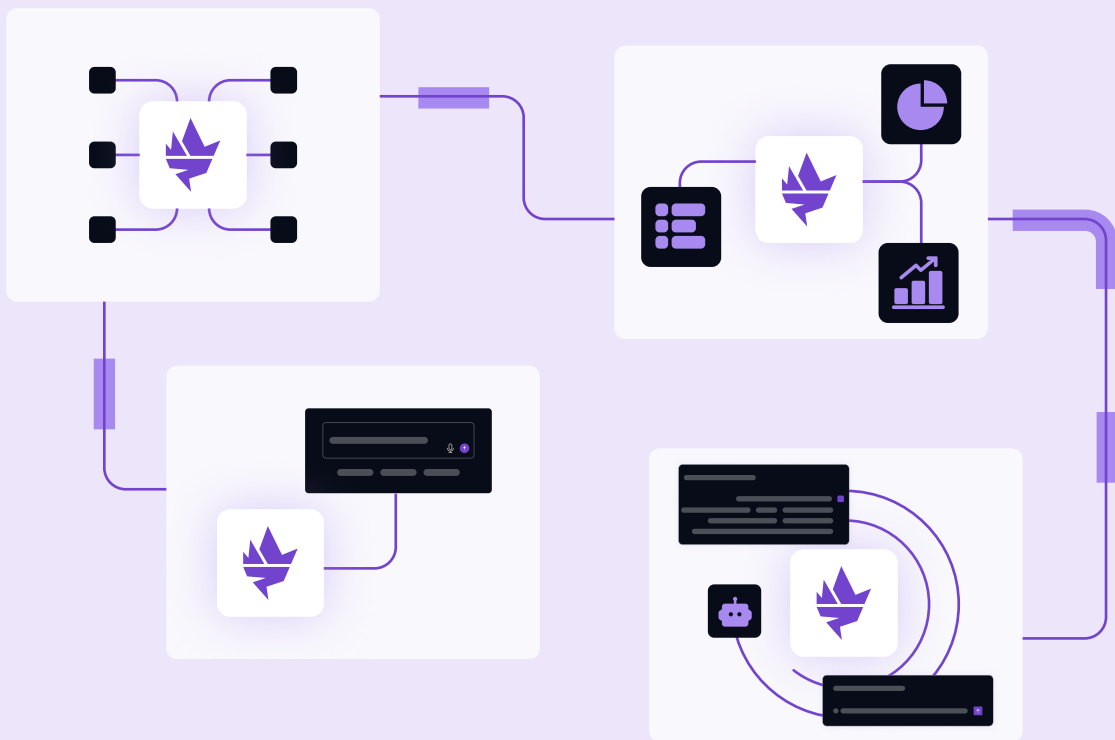


The GenAI Journey on the Mainframe

A Four-Stage Maturity Model for Modernizing
Mainframe Operations with Generative AI



From prompts to autonomous agents - how mainframe
customers are professionalizing GenAI to disrupt the
mainframe software market.

May 2026

Executive summary

Generative AI is no longer a curiosity on the mainframe, it is becoming a core part of how customers operate, troubleshoot, plan capacity, and engage their business users. But adoption is not a single event. Across the customer base, a consistent pattern is emerging: a four-stage maturity journey that begins with simple, conversational prompts and ends with autonomous agents executing complex operational workflows.

This whitepaper describes that journey, the practical use cases observed at each stage, and the implications for IT organizations and the mainframe software market as a whole. The thesis is straightforward: GenAI will reshape the mainframe ecosystem in the same way it has already reshaped the SaaS market and the customers furthest along this journey will set the standard for what running a modern mainframe environment looks like.

The four stages:



Prompts, Automation, Personalization & Supportability, and Agentic Capabilities – are not optional checkpoints. Each stage builds the trust, data foundation, and operational discipline required for the next.

Why this matters now

Mainframe environments are defined by depth: decades of accumulated configuration, custom code, undocumented tribal knowledge, and operational rituals built around legacy interfaces. For years, this depth has been a barrier to modernization. Generative AI inverts that equation. A model that can read documentation, parse logs, reason about transactions, and explain itself in natural language is uniquely well-suited to environments where the cost of context-switching has always been highest.

The customers leading this transition are not chasing novelty. They are pursuing concrete outcomes: faster incident resolution, shorter ready-for-business windows, broader access for business users, reduced dependence on a shrinking pool of specialists, and a path to automate the operational work that has historically required senior engineers.

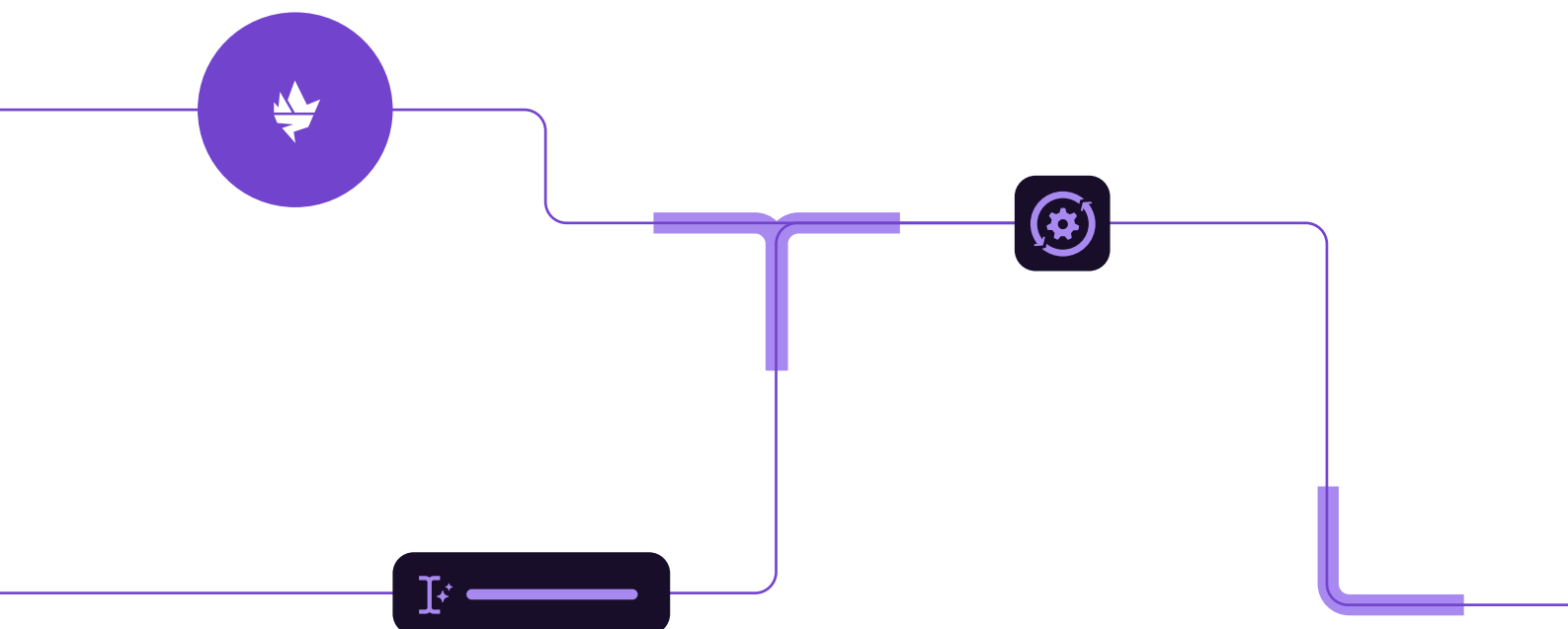


The Four-Stage GenAI Journey

The journey is cumulative. Each stage delivers value on its own, but the real return compounds as customers progress. Skipping stages is possible in theory, but in practice it tends to produce fragile deployments – agents that act on data no one has verified, or automations built on prompts no one has stress-tested. The table below summarizes the journey before we walk through each stage in detail.

Stage	Focus	Representative Use Cases	Primary Outcome
1. Prompts	Conversational Q&A against your different mainframe data sources	Different questions about current system status. How to improve configuration, database alignment, etc*	Faster answers, better efficiency, improved mainframe skills
2. Automation	Codifying repeatable prompt workflows	Ready-for-Business checks; Capacity on Demand analysis; recurring reports	Hours of manual work compressed to minutes
3. Personalization & Supportability	Teaching GenAI the customer's specific environment	Skills for home-grown apps; ingestion of internal docs; business-user prompting	GenAI that understands your specific environment and its language, not just the platform
4. Agentic Capabilities	Autonomous, multi-step task execution	Scheduled health checks; auto-triage and fix-download; ticket creation with diagnostics; upgrades and updates to multiple LPARs	Operational work that runs itself (with or without human in the loop)

* More use case can be found in previous whitepaper: [The 20 Most Popular Mainframe Prompts Our Customers asked the Operations Genie](#)



Prompts

Every GenAI journey on the mainframe begins the same way: someone types a question into a chat interface and gets a useful answer back. This stage is deceptively simple, but it is where trust is built. If the first answers are wrong, vague, or generic, the rest of the journey stalls. If they are accurate and specific, the organization quickly recognizes that something has changed.

What customers do in this stage

At this stage, prompts fall into three broad patterns. The first is understanding the system – getting a natural-language explanation of why something is slow or wrong in the system, what a return code means, or what a particular subsystem is doing. The second is documentation search – directing public manuals and reference material at a specific symptom rather than reading through hundreds of pages. The third, and often the most striking for customers, is targeted operational insight against live data.

- ▶ **System understanding:** "Explain what this MVS message means and what typically causes it."
- ▶ **Operational insight:** "Which DB2 tablespaces should I reorg this weekend, and why?"
- ▶ **Performance triage:** "Why is transaction TRN0023 running slower than usual today compared with last week?"

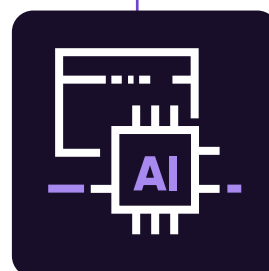
The shift from "how do I find this answer" to "what is the answer" is significant. Even at this earliest stage, the time saved by a senior systems programmer or DBA over a quarter is measurable.

What good looks like

A successful Stage 1 deployment shares a few characteristics: answers are grounded in the customer's real telemetry and not just generic platform knowledge; the model knows when it does not know and says so; and the team has a feedback loop for capturing prompts that worked well so they can be reused.



The point of Stage 1 is not to solve every problem. It is to prove that GenAI can give useful, accurate, environment-aware answers and to identify the prompts the team keeps coming back to. Those prompts become the seeds for Stage 2.



Automation

Once customers see that their prompts produce reliable answers, the next instinct is unavoidable: if a prompt works once, why type it again? Stage 2 is about turning trusted prompts into repeatable, scheduled, or event-triggered workflows. The team stops asking the question and starts receiving the answer.

From single answers to standing workflows

The work that gets automated first is rarely glamorous, and that is the point. It is the routine analysis that consumes hours of senior engineers' time every week – the morning checks, the pre-market validation, the periodic capacity reviews, the executive briefings that follow every unplanned event. Each of these tasks is structured enough to script, but historically too nuanced for traditional automation to handle on its own. GenAI changes the economics.

Use case

Ready-for-Business validation

Before markets open, financial institutions run a battery of checks to confirm the platform is healthy: batch completion, subsystem availability, queue depths, replication status, and dozens of similar indicators. The legacy approach is a checklist worked through by an operator; the modernized approach is a single automated prompt that gathers the same signals, summarizes them in plain language, flags anomalies, and routes exceptions to the right team – all before the duty manager has finished their first coffee.

Use case

Capacity on Demand (CoD) post-event analysis

When a Capacity on Demand event fires and additional capacity is brought online, executives want to know what happened and why. Was it driven by a planned workload, an outlier batch run, a runaway transaction, a seasonal spike? Today, that analysis often falls to a capacity planner who pulls SMF data, correlates it with workload schedules, and writes a narrative. A well-designed automation can produce a draft of that narrative in minutes, citing the actual workloads, jobs, and time windows responsible – and including the licensing implications.

Other workflows commonly automated at this stage

- ▶ Weekly health reports for application owners, written in business-relevant language
- ▶ Daily DB2 reorg candidates ranked by impact, with a recommended maintenance window
- ▶ Change-window readiness summaries covering active locks, in-flight units of work, and replication lag
- ▶ Post-incident retrospectives drafted automatically from logs, alerts, and on-call notes



Automation is where GenAI starts paying for itself. The same prompt that saved an engineer fifteen minutes in Stage 1 saves the team fifteen hours a week when it runs on a schedule against every system.

Personalization & Supportability

Every mainframe customer eventually runs into the same wall: their environment is not generic. It contains home-grown applications with names no one outside the building recognizes, CICS transactions wired to business processes that exist nowhere in any vendor manual, internal terminology that has accumulated over decades, and operational conventions that are documented only in the memory of long-tenured staff. A GenAI that knows the platform but not the environment can only take a customer so far.

Stage 3 is the point at which GenAI starts to understand the customer's specific world – and it is also the point at which the journey becomes a strategic differentiator rather than a productivity tool.

Skills: self-configuration in natural language

Skills are the mechanism by which a customer teaches GenAI about their environment without writing code. A skill might describe how a home-grown application is structured, which logs matter, which return codes indicate which business conditions, and how the team would normally investigate a problem. The customer writes this once, in plain language, and from that point forward the model behaves as though it has worked on the team for years.

Uploading internal documentation extends the same idea. Architecture diagrams, runbooks, internal naming conventions, ownership lists, and post-mortems all become part of the model's working context. A question that previously required a tribal expert can now be answered by the system itself, with citations back to the source.

Connecting CICS transactions to business processes

One of the most valuable outcomes of Stage 3 is the ability to bridge technical artifacts and business meaning. A transaction ID alone tells an operator very little. The same transaction ID, mapped through a skill to "the trade settlement instruction flow for the institutional desk," tells a business stakeholder exactly what they need to know. The mapping is built once and reused everywhere; incident notifications, capacity reports, change reviews, executive dashboards.

Opening the mainframe to business users

Once GenAI speaks the customer's language, an interesting possibility opens up: business users themselves can interact with the mainframe directly, using their own terminology, without ever knowing they are doing so. A wealth-management analyst asking "are all overnight valuations complete and reconciled for the European book?" does not need to know which jobs, which datasets, or which subsystems are involved. The skill layer translates the question into the appropriate queries, runs them, and returns an answer in the language the analyst expects.



Stage 3 redraws the boundary between "mainframe people" and "the rest of the business." The mainframe stops being an opaque back end and starts being a system anyone in the organization can ask questions of safely, and in their own words.

Agentic Capabilities

The final stage of the journey is the most transformative, and the one that depends most heavily on the foundations built in the previous three. Agents are not just prompts on a schedule. They are systems that perceive a situation, decide on a course of action, execute that action, observe the result, and adapt – operating across the same boundaries a human engineer would, including external vendor portals and ticketing systems.

An organization that has not built the trust, data quality, and personalization of the earlier stages cannot safely deploy agents. An organization that has built those foundations finds that agents close the loop on work that has always been done by people.

Use case

Continuous system health checks

A scheduled health-check agent runs on its own cadence, exercises each component, compares the results against historical baselines, and either confirms that everything is normal or escalates with a narrative explanation. Unlike traditional monitoring, the agent decides what to look at next based on what it finds. Investigating anomalies the way an engineer would, rather than ticking through a fixed checklist.

Use case

End-to-end fix triage

Consider a flow that today consumes a full afternoon of a senior engineer's time: a component flags an issue, the engineer identifies the vendor responsible, checks the vendor's support portal for a known fix, downloads the fix if one exists, stages it for installation, and – if no fix is available – opens a support ticket with the right logs and diagnostic information attached. An agent can execute every step of that flow autonomously:

- ▶ Detect the issue and classify it against known signatures
- ▶ Identify the responsible software vendor from configuration metadata
- ▶ Navigate the vendor's support site and search for a matching fix
- ▶ If a fix exists, download it and prepare it for installation in the appropriate environment
- ▶ If no fix exists, open a support ticket with logs, diagnostics, and a written summary of what was already tried
- ▶ Update the internal change management system and notify the on-call team

The engineer's role shifts from executing the workflow to supervising it. The work still gets done, but it is no longer the work of people; it is the work of systems that people oversee.

What changes at this stage

- ▶ **Operational footprint:** fewer people doing repetitive work, more people focused on design, governance, and exception handling.
- ▶ **Skills profile:** engineers spend more time defining policies, guardrails, and review criteria for agents, and less time executing those policies by hand.
- ▶ **Vendor relationships:** the agent becomes a heavy consumer of vendor support channels, and vendors that expose machine-readable interfaces gain an advantage.
- ▶ **Risk model:** the failure modes change – agents fail differently from humans, and governance must evolve to match.

Market Implications

It is tempting to view this journey as an internal IT transformation. It is more than that. The same forces that reshaped the SaaS market over the last several years such as natural-language interfaces displacing GUIs, agents displacing scripted workflows, personalization displacing one-size-fits-all tooling, are now arriving on the mainframe.

The implications for the mainframe software market are significant. Tools that compete primarily on the depth of their domain knowledge will increasingly compete on how well that knowledge can be exposed to and composed by GenAI. Vendors whose products are accessible only through proprietary GUIs and undocumented interfaces will struggle. Vendors who treat their products as composable building blocks; discoverable, documented, agent-friendly will thrive.

Customers, in turn, will increasingly evaluate mainframe software not just on its native capabilities but on how well it fits into their GenAI environment: how easily it can be reached by a skill, how cleanly it produces machine-readable output, how it behaves when an agent rather than a human is on the other end of the interface.



We expect GenAI to disrupt the mainframe software market on the same arc and at a similar magnitude as it has already disrupted SaaS. The window for vendors and customers to position for this shift is now, not in two years.

Looking ahead

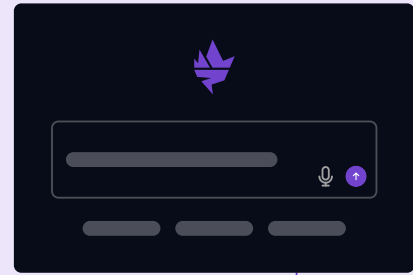
The most interesting part of this journey is that it is not finished. As underlying models continue to improve, and as the discipline of building GenAI into operational environments matures, new use cases will emerge that are not yet practical today. Six months from now, the boundary between Stage 3 and Stage 4 will look different. The set of tasks that can be delegated to an agent will be larger. The set of decisions that can be made without a human in the loop will expand, and the governance around those decisions will deepen.

This is why we frame GenAI on the mainframe as a journey rather than a product. The customers traveling with us are not just adopting a new tool; they are professionalizing a new capability, and they are setting the standard for what the next generation of mainframe operations will look like.

Where to Start

For organizations that are early on this journey, the path forward is concrete. Begin by identifying the use cases GenAI can help you to automate, save you time and bring better efficiency. Install and test the GenAI tools available to leverage the latest frontier models (LLMs).

The journey is real, the destination is closer than most customers expect, and the cost of delay rises with every quarter. The customers furthest along are not waiting for the technology to settle. They are shaping it.



Contact us to learn more
or schedule a live demo at

contact@geniez.ai



