

GAPVELOCITY AI RESEARCH

The State of Legacy Modernization Demand 2026

What 1,105 organizations told us about the code they are trying to leave behind

68%

of organizations that inquired about legacy modernization in the past twelve months named Visual Basic 6 — a language that shipped in 1998 and lost vendor support in 2008 — as the technology they need to replace.

58%

of legacy modernization demand originated outside the United States, across more than 120 countries.

Published August 2026

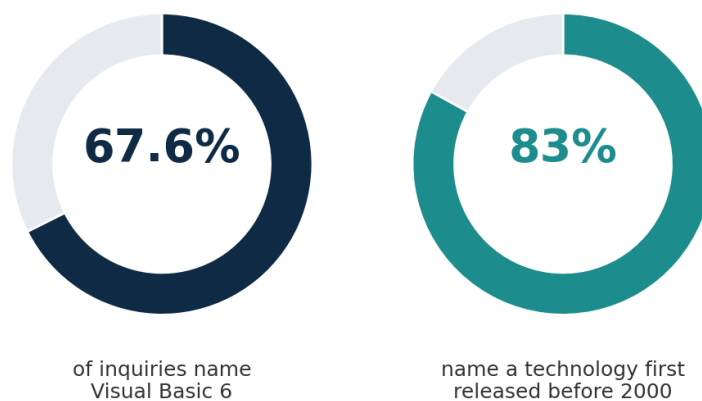
Sample period: 1 August 2025 – 21 August 2026

N = 1,105 self-reported modernization workloads; N = 9,636 inbound contacts (geography)

Key findings

Every figure in this report is drawn from first-party inbound demand data. Underlying counts are published in full, and sample limitations are stated in the final section so that anyone can check the arithmetic or reweight the sample.

- **Visual Basic 6 is the modal legacy workload.** 67.6% of organizations that inquired about modernizing a legacy application between August 2025 and August 2026 identified VB6 as the source technology (747 of 1,105). VB6 appeared 12 times as often as the next single technology.
- **Legacy is older than the industry assumes.** 83% of identified workloads run on a technology first released before the year 2000. The wider Visual Basic family (VB6, VB.NET, Office VBA) accounts for 73.4% of demand.
- **Legacy modernization is a global problem.** 58% of inbound modernization demand originated outside the United States, spanning more than 120 countries (N = 9,636).
- **Microsoft Access is the fastest-growing source of demand.** Access inquiries grew 6.3x year over year, faster than any other tracked technology, while total identified workloads grew 3.3x.
- **The long tail is underserved.** Demand spans 15 named source technologies. Clarion, a 4GL released in 1986, drew more inquiries than Delphi or Classic ASP — and has effectively no automated migration market.
- **Counting buyers and counting code gives different answers.** Across 467 scoped applications, the median PowerBuilder codebase is eight times the size of the median VB6 codebase and the median Clarion codebase more than ten times. Weighted by code volume, VB6's share falls from 68% of organizations to roughly 30%.



Source: GAPVelocity AI inbound demand data. N = 1,105.

Figure 1. Two numbers that describe the 2026 legacy modernization demand mix. N = 1,105.

Why this report exists

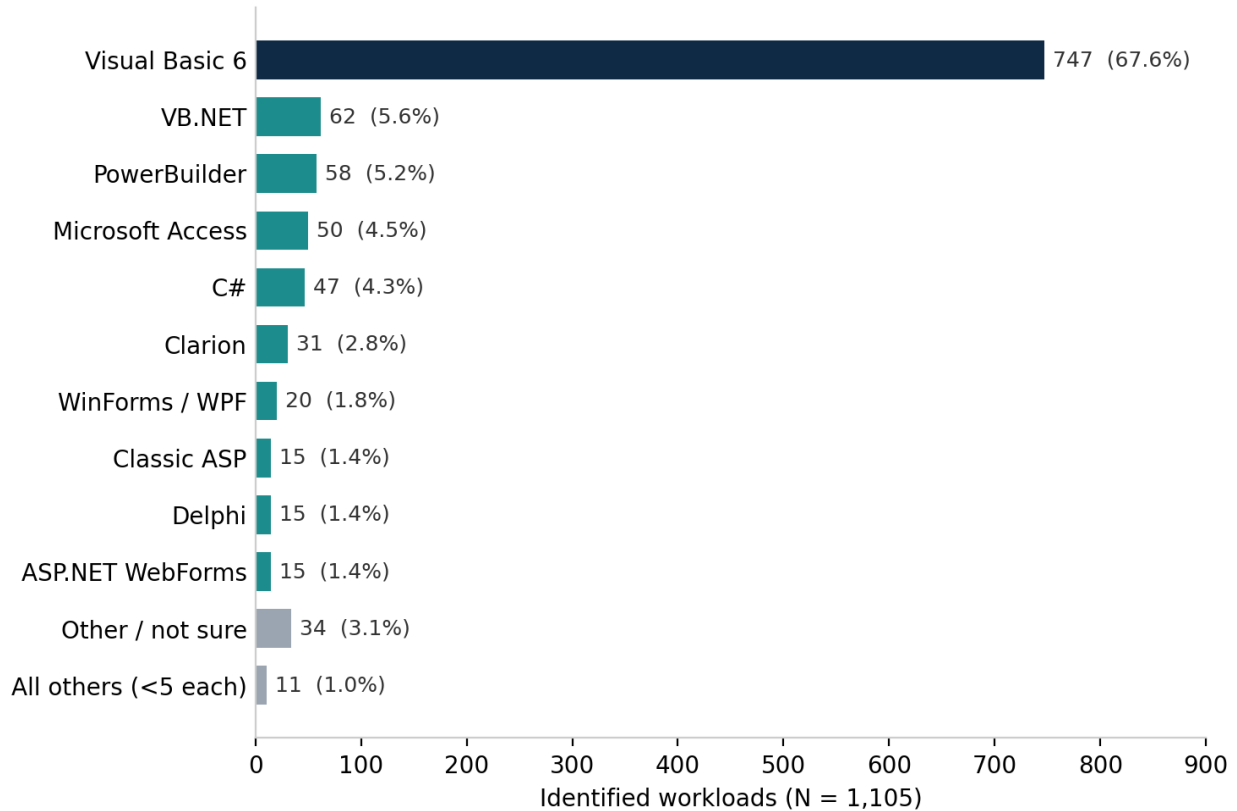
The size of the world's legacy code estate is estimated constantly and measured almost never. The figures in circulation — billions of lines of COBOL, trillions of dollars of technical debt — are extrapolations from surveys of enterprises that already know they have a problem, and those surveys rarely ask which technologies are actually running.

This report uses a different instrument. It is built from the self-reported source technology of every organization that approached GAPVelocity AI about modernizing an application over a twelve-month period. It is a demand-side sample rather than a census, and it carries a skew toward Microsoft-adjacent technologies — and toward Visual Basic 6 in particular, for reasons of the publisher's own history that are set out in the limitations. Within those limits, it answers a question the estimates cannot: when an organization decides to move off legacy code in 2026, what is it moving off?

Two terms recur and are not used interchangeably. Modernization is the outcome: an application running on a supported, maintainable stack with an architecture its owners can continue to develop. Migration is one method of reaching that outcome — the automated or assisted transformation of existing code to a new platform — and it is used here only in that narrower sense. Every inquiry in the sample is a modernization inquiry; how many will be met by migration rather than by rewrite or replacement is not something intake data can show.

1. Visual Basic 6 is the Center of the Market

Visual Basic 6 shipped in 1998. Mainstream support for the development environment ended in 2005 and extended support ended in 2008. Microsoft's current position is that the VB6 IDE is unsupported, that the core runtime files remain supported only for the lifetime of supported Windows versions, and that organizations should migrate VB6 applications to modern technology.¹ Eighteen years after the last patch, VB6 accounts for 747 of 1,105 modernization inquiries — 67.6% of everything in the sample.



Source: GAPVelocity AI inbound demand data, 1 Aug 2025 – 21 Aug 2026. N = 1,105.

Figure 2. Self-reported source technology of legacy modernization inquiries, August 2025 – August 2026. N = 1,105. Categories with fewer than five records are grouped.

No other lineage comes close. The second-largest single technology, VB.NET, accounts for 5.6%; PowerBuilder for 5.2%. Counting the wider Visual Basic family — VB6, VB.NET and Office VBA — raises the share to 73.4%. The chart needs an 800-record axis because of one technology; every other bar sits below 63.

A VB6 application written by a plant controller and a PowerBuilder system built by an enterprise IT department are one record each, and the second may contain many times the code of the first. By volume of code, PowerBuilder, Clarion and the 4GL tail hold a considerably larger share of the estate than their inquiry counts suggest. What the counts establish is which stack the buyer is trying to leave; the size of what they are leaving is a separate question, answered in Section 7.

Why the concentration is higher than most models predict

Two structural features of VB6 explain its persistence, and both cut against the way legacy estates are usually estimated.

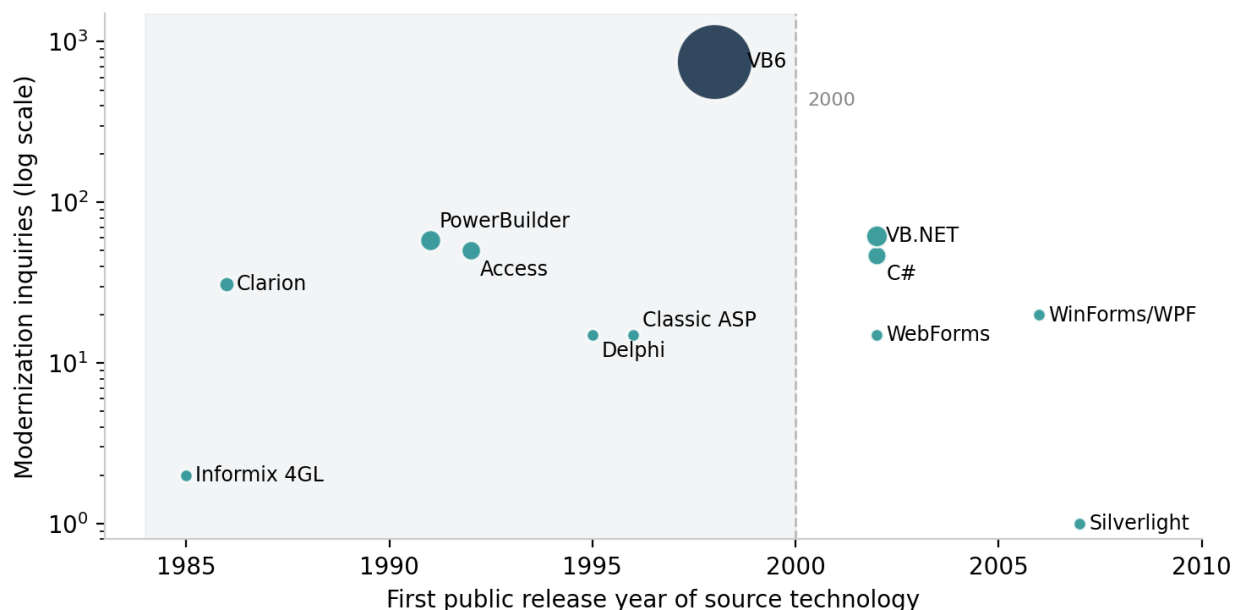
- **VB6 applications were built close to the business.** They were frequently written by domain experts in finance, operations and manufacturing rather than by professional development teams, to handle scheduling, pricing, inventory and costing. Years of fixes and exceptions turned the code into a working record of how the business operates.

- **A working VB6 application produces no pressure to move.** The forcing function is almost always external — a Windows rollout, an acquisition, a compliance audit, a failed 32-bit dependency, or the retirement of the one developer who understands the system. That makes demand lumpy and largely invisible until it surfaces as an urgent project.

A supported runtime is only one part of a working application. Many VB6 systems depend on abandoned installers, 32-bit database drivers, COM components and third-party ActiveX controls. Runtime support does not restore support for an abandoned control, or make the original development environment maintainable.

2. Legacy is an age and architecture problem

Plotting each technology's first release year against the volume of modernization demand shows how heavily the sample tilts toward the 1980s and 1990s. Technologies first released before 2000 account for roughly 83% of identified workloads.



Source: GAPVelocity AI inbound demand data, N = 1,105. Release years are first public release of each technology.

Figure 3. Technology age versus modernization demand. Bubble area is proportional to inquiry count; vertical axis is logarithmic. N = 1,105.

The exceptions on the right of the chart are as instructive as the cluster on the left. C# accounts for 4.3% of inquiries and WinForms or WPF for a further 1.8%. Neither language is old, and both are supported. What makes organizations want to move off these technologies is the architecture: thick desktop clients, WebForms postback models, direct database coupling, old framework dependencies and deployment models that limit further development.

Legacy is an architecture problem, not only an age problem. Modern code and modern architecture go hand in hand.

3. The Tail is Long, and it is Underserved

Below Visual Basic, demand fragments across fourteen further named technologies. The full distribution is reproduced here in its entirety, including the free-text and unsure responses.

Source technology	Records	Share
Visual Basic 6	747	67.6%
VB.NET	62	5.6%
PowerBuilder	58	5.2%
Microsoft Access	50	4.5%
C#	47	4.3%
Clarion	31	2.8%
WinForms / WPF (C#)	20	1.8%
ASP / Classic ASP	15	1.4%
Delphi	15	1.4%
WebForms / ASP.NET WebForms	15	1.4%
Other legacy technology (free text)	15	1.4%
Not sure	15	1.4%
Other (free text)	4	0.4%
SQL (any vendor)	3	0.3%
Java	3	0.3%
Informix 4GL	2	0.2%
VBA (Office macros)	2	0.2%
Silverlight	1	0.1%
Total	1,105	100%

Table 1. Self-reported source technology, all inbound modernization inquiries, August 2025 – August 2026. N = 1,105. Percentages may not sum to exactly 100 because of rounding.

Three observations that are not obvious from the headline number.

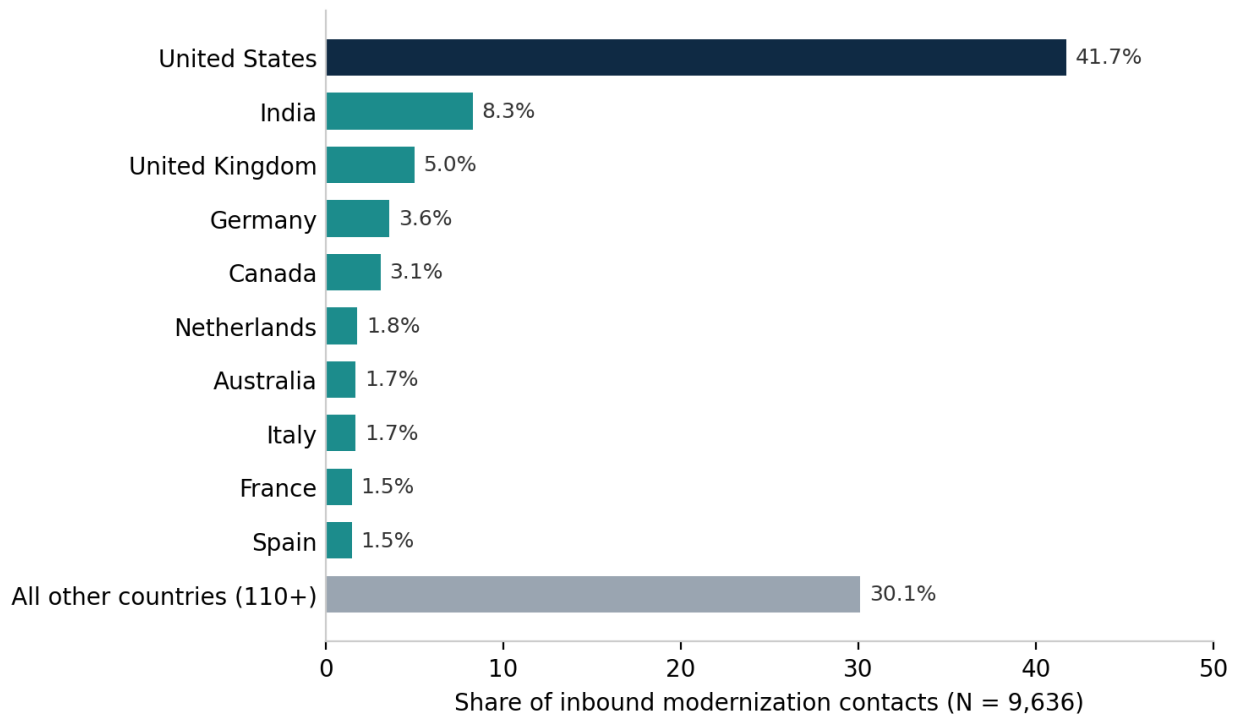
Clarion outranks Delphi and Classic ASP

Clarion, a 4GL first released in 1986 and unknown to most modernization vendors, drew 31 inquiries — more than Delphi and more than Classic ASP. Nearly a third of all inquiries (32%) name something

other than VB6, and the organizations in that group often face the hardest road. A business on Clarion or Informix 4GL confronts a market of effectively zero automated migration options, because tooling economics push every vendor toward the head of the distribution.

4. Legacy Modernization is a Global Problem

Across 9,636 inbound contacts in the same period, 58.3% originated outside the United States, distributed across more than 120 countries.



Source: GAPVelocity AI, all inbound contacts, Aug 2025 – Aug 2026. N = 9,636. Directional.

Figure 4. Geographic distribution of inbound modernization contacts, August 2025 – August 2026. N = 9,636. Country-name variants consolidated.

Country	Contacts	Share
United States	4,020	41.7%
India	795	8.3%
United Kingdom	486	5.0%
Germany	346	3.6%
Canada	294	3.1%
Netherlands	171	1.8%
Australia	167	1.7%

Country	Contacts	Share
Italy	167	1.7%
France	149	1.5%
Spain	144	1.5%
All other countries	2,897	30.1%
Total	9,636	100%

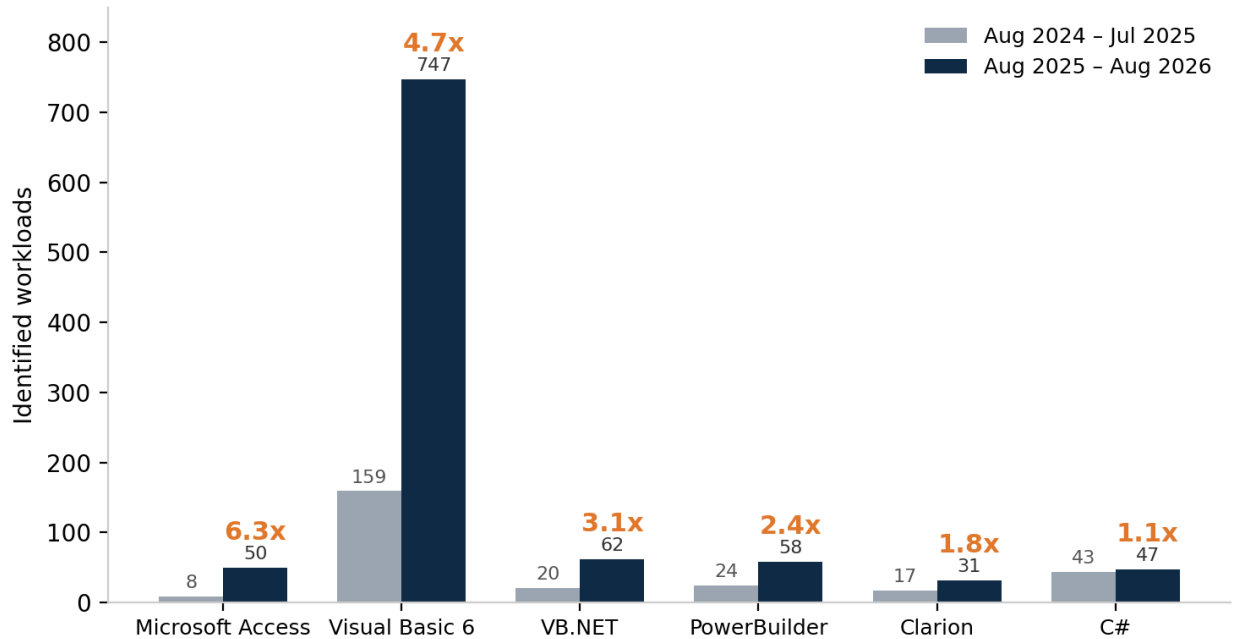
Table 2. Geographic distribution of all inbound contacts. N = 9,636.

Europe as a whole accounts for roughly 27% of contacts, and the distribution within Europe tracks enterprise software spend closely — the United Kingdom, Germany, the Netherlands, Italy, France and Spain in that order. India's 8.3% share reflects a mix of end-user organizations and systems integrators evaluating tooling on behalf of clients.

The distribution matters because legacy modernization is frequently framed as a problem of American small and mid-sized businesses running shadow IT. The demand data does not support that framing. Legacy burden appears to track the general installed base of enterprise software rather than clustering in any single region or company size.

5. Microsoft Access is the fastest mover

Comparing the twelve months to August 2026 against the preceding twelve months shows which legacy stacks entered the modernization conversation fastest. Microsoft Access grew faster than any other tracked technology, from a small base, and it is the stack organizations are least likely to volunteer that they depend on.



Source: GAPVelocity AI inbound demand data. Total identified workloads grew 3.3x; multiples = relative acceleration.

Figure 5. Year-over-year change in identified workloads, technologies with $n \geq 20$ in the current period.

Source technology	Aug 2024 – Jul 2025	Aug 2025 – Aug 2026	Multiple
Microsoft Access	8	50	6.3x
Visual Basic 6	159	747	4.7x
VB.NET	20	62	3.1x
PowerBuilder	24	58	2.4x
Clarion	17	31	1.8x
C#	43	47	1.1x
All identified workloads	331	1,105	3.3x

Table 3. Year-over-year change in identified workloads, technologies with $n \geq 20$ in the current period.

PowerBuilder's absolute volume more than doubled while its share of the mix declined slightly, from 7.3% to 5.2%.

6. The AI Demo Illusion

Generative AI has changed the economics of code analysis and transformation. It has also exposed the limits of simple code translation.

Legacy technologies have three characteristics that make them harder for large language models to modernize:

- **Small public training corpora.** There is far less VB6, PowerBuilder or Clarion code on the open internet than Java or Python, and what exists is rarely paired with a modern equivalent.
- **Heavy coupling between business logic and a proprietary GUI framework.** The behavior of a VB6 application is spread across forms, event handlers, modules, COM references, control properties and deployment assumptions. Translating one procedure at a time leaves too much context behind.
- **No canonical mapping to a modern target.** There is no agreed answer to what a PowerBuilder DataWindow or a VB6 form should become. Target architecture decisions still require engineering judgment.

A credible AI-led approach therefore has to work at application scale: inventory hidden dependencies such as 32-bit COM components, preserve business behavior, handle known language and framework patterns consistently, generate maintainable target code that does not simply replicate legacy constraints, and test both compilation and semantic fidelity.

Claims about AI-driven modernization should be evaluated against the stacks that dominate real demand.

7. Counting organizations is not counting code

The counts above measure how many organizations are trying to leave each stack. But it's also important to note how much code each organization is carrying and the function/complexity of the applications they want to modernize.

Source technology	Applications sized	Median lines of code	Median relative to VB6	Share of 2026 inquiries	Implied share of code volume
Visual Basic 6	314	124,000	1.0x	67.6%	30.0%
VB.NET	14	300,000	2.4x	5.6%	6.0%
PowerBuilder	44	1,000,000	8.1x	5.2%	18.7%
Microsoft Access	18	210,000	1.7x	4.5%	3.4%
C#	8	570,000	4.6x	4.3%	8.8%

Source technology	Applications sized	Median lines of code	Median relative to VB6	Share of 2026 inquiries	Implied share of code volume
Clarion	12	1,300,000	10.5x	2.8%	13.1%
WinForms / WPF (C#)	14	275,000	2.2x	1.8%	1.8%
ASP / Classic ASP	10	642,000	5.2x	1.4%	3.2%
Delphi	9	1,000,000	8.1x	1.4%	5.0%
WebForms / ASP.NET WebForms	10	1,000,000	8.1x	1.4%	5.0%
Informix 4GL	3	7,000,000	56.6x	0.2%	5.0%

Table 4. Application size by source technology, scoped modernization opportunities, January 2023 – August 2026. N = 467. Median is used throughout because the distribution is heavily right-skewed. "Implied share of code volume" applies each technology's median size to its share of 2026 inquiries; it is an estimate, not a measurement. Rows with fewer than ten sized applications are indicative only.

The pattern is consistent and large. The median VB6 application scoped for modernization is about 124,000 lines of code. The median PowerBuilder application is 8x larger at 1,000,000 lines of code. The median Clarion application is 1.3 million, more than ten times larger, and the three Informix 4GL estates in the sample average 7 million. These are exactly the stacks that were built by enterprise IT departments and systems integrators for core operations, while VB6 was the language of the departmental application built close to the business. The count data and the size data describe the same market from two directions, and they agree.

Applying each technology's median size to the 2026 inquiry mix produces a rough picture of where the code actually sits. Visual Basic 6 falls from 68% of organizations to roughly 30% of implied code volume. PowerBuilder rises from 5% to roughly 19%, and Clarion from under 3% to roughly 13%. On this measure, the three enterprise 4GL-era stacks — PowerBuilder, Clarion and Informix 4GL — together represent more than a third of the code seeking a way out, from about 8% of the organizations asking. The estimate is crude: it assumes the size distribution of scoped opportunities holds for inquiries that never reached scoping, which almost certainly overstates the tail. Even halved, the conclusion stands.

Four implications

The usual estimates undercount the technologies that matter most

Estimates of legacy code volume are built from surveys of IT organizations about the systems they manage. The technologies concentrated at the top of this distribution — Visual Basic 6, Access, Clarion — are precisely those least likely to appear in a managed application inventory, because they were often built outside IT. The gap between estimated and actual legacy burden is likely to be widest exactly where demand is heaviest. Discovery work should include operations, finance and manufacturing teams, not only the systems already known to central IT.

The AI-migration thesis has not been tested against the real mix

The benchmark for AI modernization tooling should be the demand mix documented here, not the stacks that are easiest to demonstrate. VB6 in particular is a demanding benchmark: its behavior is distributed across UI, COM and deployment layers in ways that defeat procedure-by-procedure translation.

The long tail is a real market gap, and agentic AI is the first plausible answer

Roughly one in three inquiries names something other than VB6, and a meaningful share of those fall in technologies with fewer than a few dozen annual buyers worldwide. For those organizations, the realistic options have been a manual rewrite or continued deferral. This is the segment where general-purpose agentic tooling, rather than stack-specific converters, is most likely to change the economics.

What comes next: measuring the code, not just the demand

Demand data should lead to code-estate data. Section 7 gives a first, scoping-based answer. GAPVelocity AI intends to publish a companion report drawn from automated code assessments rather than sales scoping: measured application size by source technology, counts of third-party controls and COM dependencies, database access patterns, unsupported dependencies, common language patterns and migration outcomes. Those benchmarks would give modernization teams a basis for planning and give the industry a better picture of the legacy estate still in service — including whether the modal buyer and the modal codebase are the same thing.

Methodology

This report is based on first-party data from GAPVelocity AI's inbound demand between 1 August 2025 and 21 August 2026, a reporting period of 386 calendar days.

Technology findings (N = 1,105)

Source technology is self-reported by the requester on an intake form at the point of inquiry. Respondents select from a fixed list with a free-text alternative. Each record represents one inquiry about modernizing a specific application workload, not a completed engagement. An organization inquiring about more than one workload appears once per workload, so the headline figure is best read as 1,105 workloads. Duplicate picklist values arising from a mid-period form change were consolidated; "other" and "not sure" responses are reported separately rather than redistributed across named categories. The data has not been independently verified against source code.

Geographic findings (N = 9,636)

Country is drawn from all inbound contacts in the period, including free-tool downloads and event registrations. This is a broader and differently composed population than the technology sample, and it includes some third-party enrichment. It is reported as a distribution only.

Year-over-year comparison

The prior period is 1 August 2024 to 31 July 2025 (365 days); the current period is 386 days. The comparison is restricted to technologies with at least twenty records in the current period; this threshold was set before the results were examined.

Application size findings (N = 467)

Application size is drawn from GAPVelocity AI's opportunity records created between 1 January 2023 and 21 August 2026 for which an effective lines-of-code value was recorded during scoping — 467 of 807 opportunities in the period. Records under 1,000 lines and one record with an implausible value (over one billion lines, almost certainly a data-entry error) were excluded. Source technology is taken from the opportunity's recorded migration path. Roughly a quarter of these opportunities closed as won; the remainder were lost, stalled or remain open. Size figures are typically supplied by the prospect or estimated from a preliminary code inventory during scoping; about a third are round figures to the nearest 100,000 lines, so individual values are approximate and medians are reported in preference to means.

Engagement findings

Page-level metrics are taken from server-side web analytics on gapvelocity.ai for the same period, covering the forty highest-traffic pages and approximately 140,000 page views. They provide supporting context and are separate from the 1,105 workload records.

Technology release years

The "pre-2000" share uses the first public release of each named technology: Informix 4GL (1985), Clarion (1986), PowerBuilder (1991), Microsoft Access (1992), Delphi (1995), Java (1995), Classic ASP (1996), Visual Basic 6 (1998). VB.NET, C#, WebForms, WinForms/WPF and Silverlight are counted as post-2000.

Limitations

- This is a demand-side sample, not a census. It reflects organizations actively researching modernization, which is a smaller and different population than organizations still running legacy code.
- The sample is skewed toward Microsoft-adjacent technologies, and toward Visual Basic 6 in particular, for a specific historical reason. Microsoft funded and specified the original Visual Basic Upgrade Wizard, built by ArtinSoft and shipped inside Visual Studio .NET, and the publisher's lineage (ArtinSoft, Mobilize.Net, GAPVelocity AI) has been the default destination for VB6 migration questions for more than two decades. Organizations looking to leave VB6 are therefore more likely to find this publisher than organizations leaving any other stack, and the 67.6% VB6 share should be read as an upper bound for the wider market rather than a neutral estimate. Mainframe COBOL, SAP ABAP and mainframe-adjacent 4GLs are almost certainly under-represented relative to their true share of global legacy burden. The figures describe the composition of demand reaching one modernization provider, not the global installed base.
- Inbound volume is influenced by the publisher's own marketing activity, including event-driven spikes. Year-over-year comparisons should be read as relative acceleration within a growing sample. Future trend comparisons should use equal reporting windows and a stable category definition.
- The intake form's picklist changed partway through the period. Duplicate values were consolidated, but the change may have affected how respondents categorized themselves.
- Source technology is self-reported and unverified. Organizations running more than one legacy stack report only the one prompting the inquiry.
- The application size sample (Section 8) is drawn from scoped opportunities, not from inquiries, and over a longer period. Opportunities that reach scoping skew larger and more enterprise than inquiries as a whole, and size values are scoping estimates rather than measured counts. The "implied share of code volume" figures in Table 4 combine the two samples and should be treated as an order-of-magnitude estimate only.

Statistics in this report

For convenience, the principal findings are restated below as standalone statements, each attributable to this report.

- 67.6% of organizations inquiring about legacy modernization between August 2025 and August 2026 identified Visual Basic 6 as the source technology (N = 1,105).
- Visual Basic 6 appeared 12 times as often as the next single source technology.
- The Visual Basic family — VB6, VB.NET and Office VBA — accounted for 73.4% of legacy modernization demand.
- 83% of legacy modernization inquiries named a technology first released before 2000.
- 32% of legacy modernization inquiries named a technology other than Visual Basic 6; demand spanned 15 named technologies.
- 58% of legacy modernization demand originated outside the United States, across more than 120 countries (N = 9,636).
- Microsoft Access modernization inquiries grew 6.3x year over year — faster than any other tracked technology.
- Total identified legacy modernization workloads grew 3.3x year over year within the sample.
- C# and WinForms/WPF together accounted for 6.1% of legacy modernization demand despite being fully supported technologies.
- Across 467 scoped modernization opportunities (2023–2026), the median Visual Basic 6 application was about 124,000 lines of code; the median PowerBuilder application was 1,000,000 lines and the median Clarion application 1.3 million.
- Weighted by median application size, Visual Basic 6's share of legacy modernization demand falls from 67.6% of organizations to roughly 30% of code volume, while PowerBuilder rises from 5.2% to roughly 19%.

Source

¹ Microsoft, "Support Statement for Visual Basic 6.0 on Windows." The VB6 IDE is unsupported; core runtime files remain supported for the lifetime of supported Windows versions, with support limited to serious regressions and critical security issues. Microsoft recommends migrating VB6 applications to modern technology.

Citation

GAPVelocity AI. *The State of Legacy Modernization Demand 2026: What 1,105 Organizations Told Us About the Code They Are Trying to Leave Behind.* August 2026.

gapvelocity.ai/research/modernization-demand-2026

This report may be quoted and reproduced with attribution. The underlying counts are published in full above. Methodology questions from analysts and journalists are welcome at research@gapvelocity.ai.

About GAPVelocity AI

GAPVelocity AI is the AI modernization business of Growth Acceleration Partners. Its lineage runs through ArtinSoft, developer of the Visual Basic Upgrade Wizard shipped with Visual Studio .NET, and Mobilize.Net, developer of the Visual Basic Upgrade Companion. Its VELO platform applies agentic AI to the modernization of Visual Basic 6, PowerBuilder, Microsoft Access, VB.NET, WebForms, WinForms and other legacy stacks. Details on VELO and its supported source technologies: gapvelocity.ai/velo