

The People Behind the Batch

What the skills surveys say, where batch knowledge piles up, and what each migration path needs from the people who hold it

Bill Ahlbrandt

President and CEO, Ahlbrandt Software

Abstract

Every few years a headline says the people who understand the mainframe are about to retire and nobody will be left to run it. I've worked on mainframe sort and batch for more than thirty years, and I wanted to see what the evidence actually says. The surveys agree that skills are hard to find. They disagree on how hard, most of them are paid for by vendors, and they describe the people who answered, which isn't the same thing as the workforce. There's real evidence on the other side too. The people answering the surveys are getting younger, employers are hiring, and training and code-explanation tools are easy to get. In batch, the scarcer knowledge is often in operations: JCL, scheduling, restart and recovery, utility control statements and installation defaults. That knowledge usually sits with a few long-serving people and is seldom written down. The paper shows how to measure that concentration and how to capture the knowledge, and what rewriting, converting or running the batch unchanged does to the skills a shop will need in 2030.

What I found

- The most-quoted figure, that **70 percent** of organizations struggle to find mainframe talent, comes from a vendor survey of 500 senior leaders. Other surveys report anywhere from about 39 to 62 percent, depending on who answered and how the question was asked.
- In the same survey, only **23 percent** said they lacked legacy programming language skills. More of them reported shortages in AI, cloud and integration skills.
- Surveys that ask about age show respondents getting younger, and employers say they're hiring. The bigger problem the evidence points to is **knowledge concentrated in a few people**.
- In batch, the knowledge that's hardest to replace is in **operations**: scheduler rules, restart procedures, utility control statements and installation defaults. Code-reading tools can't get it back, because most of it was never in the code.
- You can measure and reduce the risk now, from SMF records, scheduler exports and recorded walkthroughs. Each migration path changes **which** skills you need. None of them lets you skip capturing this knowledge first.

1. What April 2020 actually showed

On April 4, 2020, with unemployment claims coming in faster than New Jersey's systems could process them, Governor Phil Murphy used his daily briefing to ask for volunteers who knew COBOL. He described systems more than forty years old and asked how the state had got to a point "where we literally needed COBOL programmers".¹ More than 362,000 residents had filed claims in two weeks, and the state labor department reported a 1,600 percent increase in its usual volume of requests.² Within days IBM and the Open Mainframe Project had set up a forum to connect employers with COBOL programmers, a technical help forum and a free introductory COBOL course.³

The episode turned into shorthand for a skills crisis. The later analysis is more careful. A 2024 Brookings working paper found that COBOL became "a binding constraint" when systems were overwhelmed or benefit rules changed a lot. It also pointed to decades of piled-up code changes, front-end websites that couldn't cope, complicated new eligibility rules and thin adjudication staff.⁴ So the lesson is narrower than the headline. Reading COBOL was one part of it. The bigger part was that the few people who understood *that* system, its rules and its operating limits couldn't be multiplied when they were needed. That's a knowledge-concentration problem, and it's what this paper is about.

The US Government Accountability Office has seen the same pattern in federal agencies. In 2019 it reported that one agency had rehired retired employees to maintain COBOL systems, and that staff shortages at the Internal Revenue Service had "hindered the agency's efforts to modernize" its core tax processing.⁵ Its 2025 follow-up noted that agencies "may have to pay a premium for specialized staff or contractors", and that only three of the ten critical legacy systems it had named in 2019 had been modernized.⁶ Both points matter for what follows. A shortage of people raises the cost of running old systems, and it also slows down the projects meant to replace them.

2. What the surveys actually say

Exhibit 1 collects the main recent sources, with who paid for them and how many people they asked. Three cautions apply to all of them. First, most are commissioned or published by companies that sell mainframe software or services, and each one has a stake in the result. Second, they survey whoever chose to answer, so a change in who answers can look like a change in the workforce. Third, the questions differ. "Difficulty finding talent for modernization", "a skills gap" and "skills-gap pressure" aren't the same measurement.

Exhibit 1. Recent evidence on mainframe skills, with sponsors and samples

Source	Sponsor and fieldwork	Sample	Skills findings
Kyndryl, State of Mainframe Modernization 2025⁷	Kyndryl (services vendor); Coleman Parkes	500 senior business and IT leaders, four regions	70% report difficulty finding skilled talent for modernization; 23% cite legacy programming language skills; 39% say retiring staff take skills with them; 74% use external providers
BMC Mainframe Survey 2025^{8,9}	BMC (software vendor); 31 Mar–15 Apr 2025	More than 1,000 practitioners and stakeholders	66% of respondents millennial or Gen Z (37% in 2018); Gen Z 15% (1% in 2018)
BMC Mainframe Survey 2026¹⁰	BMC; 24 Mar–20 Apr 2026	More than 1,300	Published release reports no workforce figures; 40% would let AI recommend code-management actions, 23% would let it complete them
Arcati Mainframe User Survey 2025¹¹	Arcati / Planet Mainframe (trade publication); Nov 2024–Jan 2025	Over 200 respondents, as reported	62% cite a skills gap; 70% have worked in mainframe roles for 20+ years, 27% for 31–40; most-wanted skill is system administration and performance tuning (62%)
Arcati Mainframe User Survey 2026¹²	Planet Mainframe	Not stated in published summaries	About 39% report some skills-gap pressure; largest experience groups now 6–10 and 11–20 years

Source	Sponsor and fieldwork	Sample	Skills findings
Futurum, Global Mainframe Skills Report 2024 ¹³	Commissioned by IBM, Broadcom and 21CS; Feb–Mar 2024	750 companies, 200 universities, 200 students	91% of employers plan to hire within two years; 65% of university leaders see more skilled talent than five years earlier
US GAO, 2019 and 2025 ^{5,6}	Official audit body	Ten, later eleven, critical federal systems	Dwindling COBOL and assembler skills; premiums for specialist staff; shortages slowed modernization

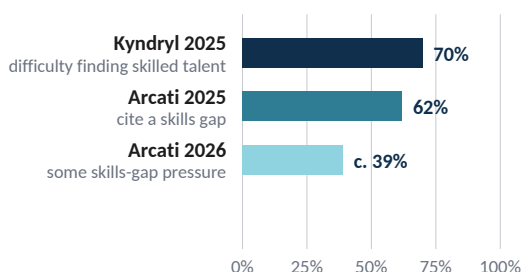
Percentages as published by each source. Survey figures describe the people who answered. They aren't a census of the workforce.

Read together, the sources support a modest conclusion. In most surveys a majority of respondents report some trouble with skills, but how big the problem looks depends a lot on how it's measured. Arcati's headline figure dropped from 62 percent in 2025 to about 39 percent in 2026, and its 2026 authors put the shift in experience levels down to their sample: "Changes in experience distribution reflect participation patterns rather than evidence of industry-wide workforce disruption."^{11,12} The same caution goes for BMC's encouraging age figures. They show who answered a survey. They aren't a census of mainframe staff.

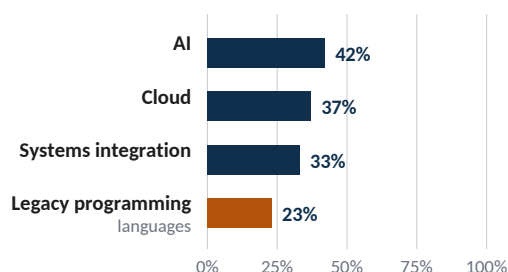
The detail is more useful than the headlines. Kyndryl's respondents reported their biggest shortages in AI (42 percent), cloud (37 percent) and systems integration (33 percent), and only 23 percent cited legacy programming language skills.⁷ Asked why mainframe skills were hard to find, 46 percent said newcomers didn't have them, 42 percent that employees didn't want to learn them, and 39 percent that staff were retiring and taking the skills with them. Arcati's respondents put system administration and performance tuning ahead of advanced programming as the skill they needed most (62 against 55 percent), and nearly a third held two or more technical roles.¹¹ Neither survey asked about batch operations directly. Both point away from programming languages as the heart of the problem.

Exhibit 2. The skills-gap figure depends on who you ask, and legacy languages come last on the list

A. The headline figure depends on the survey



B. Kyndryl 2025: where the shortage is felt



Panel A compares questions worded differently, so don't read it as a trend. The Arcati 2026 figure is reported as approximately 39 percent. Panel B: share of Kyndryl respondents naming each area as a skills shortage (n = 500). Sources: Kyndryl 2025; Arcati 2025 and 2026.^{7,11,12}

3. The other side of the story

The surveys also have evidence against the idea of a vanishing workforce. In BMC's 2025 survey of more than 1,000 people, 15 percent of respondents were Gen Z, against 1 percent in 2018, and millennials and Gen Z together made up 66 percent, against 37 percent.^{8,9} In the Futurum study, commissioned by three vendors, 91 percent of 750 employers planned to hire for new mainframe positions within two years, and 65 percent of university leaders said there was more skilled talent available than five years earlier. The report also concluded that mainframe skills were no harder for employers to find than cybersecurity or AI skills.¹³

Training is out there, and people use it. IBM reported in 2020 that its Master the Mainframe program had reached 4,286 students from more than 600 US schools the year before,³ and in 2024 that its skills accelerator had helped 83 employers train more than 440 system administrators and application developers in 13 countries.¹⁴ The Open Mainframe Project's free COBOL Programming Course, started during the 2020 crisis, is still maintained by volunteers; its repository had about 3,600 stars and 722 forks in September 2026.¹⁵ Stars aren't learners, and no

completion figures are published, but they do show continued interest.

The tools have changed too. IBM's watsonx Code Assistant for Z will explain selected COBOL, PL/I, REXX or assembler code, or a JCL step, in plain language.¹⁶ BMC markets a similar assistant as a way for newer staff to ask what a subroutine does without waiting for a senior colleague.¹⁷ Respondents are still cautious. In BMC's 2026 survey, 40 percent would let AI recommend code-management actions, and only 23 percent would let it complete them.¹⁰ Explanation tools make code cheaper to read. They can only explain what has been written down, though.

4. Where the scarce knowledge sits in batch

A batch cycle is a lot more than a set of programs. There's the JCL that strings them together, the scheduler that decides when each job can start, the utilities that sort, copy and reorganize data between programs, the installation settings that control how those utilities behave, and the procedures people follow when something fails at three in the morning. The COBOL is the best-documented part of this stack, because the source code is the documentation. The further down the stack you go, the less is written down and the fewer people know it.

Utility behavior is a good example. DFSORT reads its installation defaults from ICEPRMxx members in PARMLIB, which can set different options for each of eight environments, including batch JCL, program-invoked sorts, TSO and four time-of-day slots.¹⁸ So whether equal keys keep their input order, what happens on overflow and whether an error ends a step can depend on a member no application developer has ever seen, set years ago by a system programmer who may have retired since. The job looks the same in the JCL either way. Other sort products have their own installation option tables, and the same goes for storage-management routines, scheduler calendars and exits. Exhibit 3 maps the layers.

Exhibit 3. A skills map of the batch stack

Layer	Who usually holds the knowledge	Documentation typically present	Recoverable by reading code?
Application programs (COBOL, PL/I, assembler)	Application developers; often a wider pool	Source code; comments of varying quality; specifications often out of date	Largely: explanation tools work here
JCL and procedures	Production control and operations analysts	The JCL itself; the reasons for DISP, COND and data set choices rarely	Partly: what a step does; why it does it, rarely
Scheduler definitions and calendars	Schedulers and production control	Scheduler database; business reasons for dependencies and holidays seldom	Exportable, but exceptions and manual holds are tacit
Utility control statements (sort, copy, IDCAMS)	Operations, system programmers or developers	Members scattered across libraries and in-stream; rarely cataloged	Yes once collected, but meaning depends on defaults
Installation defaults and exits	System programmers; often one or two people	PARMLIB members and exit source; the history of changes seldom	No: not in the job at all
Restart and recovery	Senior operators and on-call staff	Run books uneven and often out of date	No
Operating norms	Long-serving operators	Seldom written: normal run times, volumes, month-end and year-end variants, known false alarms	No, though SMF history shows timings

My assessment, from experience with batch inventories. This is the typical picture. Well-run sites will have more documentation in the lower rows.

5. Key-person risk in operations

Software engineers call this the “bus factor” or “truck factor”: the number of people who would have to leave before nobody could maintain a system any more. It can be measured. A 2016 study of 133 popular open-source projects on GitHub estimated that 65 percent had a truck factor of two or less.¹⁹ Nobody has published an equivalent study of mainframe operations, and open-source projects aren't batch shops. It's still a reminder that knowledge piles up in a few heads unless somebody works against it, even in communities built around shared code.

Operations knowledge is especially likely to concentrate. It's learned from incidents, so the person who knows how to restart a job stream is usually whoever was on call the last time it failed. It's used rarely, since year-end and disaster recovery come once a year or less. And it sits with long-serving staff. In Arcati's 2025 survey, 70 percent of respondents had worked in mainframe roles for more than 20 years and 27 percent for 31 to 40 years.¹¹ Those numbers describe the respondents, who aren't the whole workforce, but somebody with more than 30 years in by 2025 will often be within a few years of retirement by 2030.

Outsourcing moves the risk somewhere else. It doesn't make it go away. Kyndryl found 74 percent of organizations using external providers for modernization work.⁷ Where operations are outsourced, the question becomes which people at the provider hold the knowledge, and what the contract says about handing it over.

A simple test

For each application group, ask: **how many people could restart its critical batch path unaided, at month-end, tonight?** Count names, not teams. A count of one is a key-person risk, whatever the org chart says. You can ask the same question about the installation defaults, the scheduler calendars and the year-end procedures.

6. Getting it written down

Most of this knowledge can be captured without a lot of effort, and much of the evidence is already on disk. The goal is modest. Take what only one or two people know and put it in a form other people can check and use. Exhibit 4 is a checklist. The inventory method is covered in more detail in *Changing Your Sort Engine Without Changing Your Batch*, and using recorded inputs and outputs as a test corpus in *Same Input, Same Output*, both earlier papers in this series.

Exhibit 4. A knowledge-capture checklist for batch operations

What to capture	How	How you know it's done
What actually runs	SMF type 30 job and step records, types 14 and 15 for data sets and type 16 for DFSORT, over a full cycle including month-, quarter- and year-end	Catalog of jobs, steps, programs, utilities and data sets, with run frequency
Scheduling rules	Export scheduler definitions; list calendars, manual holds and special resources; note the reason for each dependency that isn't obvious	Dependency map reviewed by the scheduler owner
Utility control statements	Collect every SYSIN member and in-stream deck used by a step; link each to its jobs; add a one-line purpose	Control-statement library under version control
Installation defaults and exits	Run the ICETOOL DEFAULTS report (or the equivalent for other products); list active exits and parmlib members	Defaults report kept with each change record
Restart and recovery	For each critical-path job: restart point, clean-up, data to restore, escalation	Run book used successfully by someone other than its author
Rare procedures	Record walkthroughs of month-end, year-end and a rehearsed recovery; transcribe and index them	Indexed recordings linked from the run books

What to capture	How	How you know it's done
Expected results	Keep representative inputs and outputs of each critical job as a regression set	Corpus that can be replayed and compared byte for byte
Concentration	Count named people per application group, as in the box above; pair and rotate staff until each count is at least two	Named cover recorded and reviewed each quarter

ICETOOL DEFAULTS reports the merged PARMLIB and ICEMAC default values for each environment.¹⁸

AI explanation tools help here, if you're careful with them. They can produce a first description of a program or a JCL step in minutes, and a long-serving colleague can correct it much faster than they could write it from scratch. The correction is the valuable part. An unreviewed explanation of code that depends on an undocumented default can be confidently wrong.

7. What each migration path does to the skills you need

The routes off the mainframe, or within it, are compared in *Rewrite, Convert or Run Unchanged*. From the skills side, each one changes which knowledge you need and when you need it.

Rewrite

Rewriting replaces COBOL, JCL and utility steps with code in a new language and a new scheduler. In the long run it moves the skills requirement to a bigger labor market. In the short run it needs the scarce knowledge *twice*: once to pin down exactly what the old batch does, including the behavior set by defaults and exits, and again to design restart, recovery and scheduling on the new platform. The GAO's finding that staff shortages slowed modernization at the IRS is exactly this risk.⁵ A rewrite that starts after the key people have left has to rebuild their knowledge from the outputs.

Convert

Automated conversion translates JCL, programs or control statements into another syntax. It shortens the specification work, but what comes out is new material nobody at the shop wrote, and the team has to learn it. The operations layer still has to be rebuilt, and the defaults the converter assumed have to be checked against the ones the site actually used. The skills requirement moves to the converted code and the converter's conventions.

Run unchanged

Rehosting onto a platform that accepts the existing JCL and control statements, with a replacement sort engine and other utilities that read the same syntax, keeps most of the upper layers intact. Existing documentation and what the staff already know stay valid, and the new skills you need are the ones for the target platform. The trade-off is plain. You still need people who can read JCL and sort control statements, so this path keeps the need for those skills going for longer. Installation defaults still have to be captured, because the replacement has to be configured to match them.

Stay

Staying on z/OS keeps every layer as it is and makes the capture work in section 6 the whole program. For a lot of shops that's the right answer, and section 3 suggests people can be hired and trained. What you can't hire is thirty years of knowledge about one particular batch cycle.

8. Bottom line

- **There's a skills problem. The evidence for a cliff is weak.** Most surveys report difficulty, but how much depends on the sponsor, the sample and the question, and there's real evidence of new people coming in and of hiring.

- **COBOL isn't the scarcest skill.** Respondents rank it below AI, cloud and integration. In batch, the scarcer knowledge is operational and specific to each site.
- **The risk is concentration.** Scheduler rules, restart procedures, control statements and installation defaults are often known by one or two people and written down nowhere.
- **You can measure it and reduce it now.** SMF records, scheduler exports, the defaults report and recorded walkthroughs cost very little next to any migration.
- **Capture comes before any migration decision.** Rewrite, convert, run unchanged or stay: every path depends on knowing what the batch does today, and the people who know are the input to all four.

About the author

Bill Ahlbrandt is President and CEO of Ahlbrandt Software, which he founded in 1993. Before that he was a systems programmer, Director of Information Security and Director of R&D at Acxiom. He has been working on mainframe sort and batch for more than thirty years, and his company's sort engine runs mainframe sort jobs on Linux and other platforms for banks, insurers and government agencies.

Full disclosure: my company sells sort software for platforms other than the mainframe. It reads mainframe sort control statements unchanged, which is one of the migration paths in section 7. You need the capture work in this paper whichever path you choose.

References

1. CNBC, "New Jersey needs volunteers who know COBOL, a 60-year-old programming language", 6 April 2020 (remarks by Governor Phil Murphy, 4 and 6 April 2020).
2. *The Register*, report on New Jersey's appeal for COBOL volunteers, 5 April 2020 (claims volumes as stated by the state).
3. IBM, press release, "IBM and Open Mainframe Project Mobilize to Connect States with COBOL Skills", 9 April 2020.
4. M. A. Navarrete, "COBOLing Together UI Benefits: How Delays in Fiscal Stabilizers Affect Aggregate Consumption", Hutchins Center Working Paper 98, Brookings Institution, September 2024, pp. 4–5.
5. US Government Accountability Office, *Information Technology: Agencies Need to Develop Modernization Plans for Critical Legacy Systems*, GAO-19-471, June 2019.
6. US Government Accountability Office, *Information Technology: Agencies Need to Plan for Modernizing Critical Decades-Old Legacy Systems*, GAO-25-107795, 17 July 2025.
7. Kyndryl, *2025 State of Mainframe Modernization Survey Report*, research by Coleman Parkes, released 9 September 2025 (n = 500), p. 10. Vendor-sponsored.
8. BMC, press release, "20th Annual BMC Mainframe Survey Reveals Confidence, Generational Shift in Views on Mainframe Development", 16 September 2025 (n > 1,000; fieldwork 31 March–15 April 2025). Vendor-sponsored.
9. N. Tardy, "Mainframers Are Getting Younger: BMC Survey", TechChannel, 22 January 2026 (age bands from the 2018 and 2025 BMC surveys).
10. BMC, press release, "Trust Powers AI on the Mainframe", 1 September 2026 (n > 1,300; fieldwork 24 March–20 April 2026). Vendor-sponsored.
11. Arcati / Planet Mainframe, *Arcati Mainframe Navigator 2025: User Survey* (fieldwork November 2024–January 2025); and Planet Mainframe, "Top 10 Key Findings from the 2025 Arcati Mainframe Survey", February 2025. Trade publication; vendor-supported.
12. Planet Mainframe, A. Hendley, "Mainframe Workforce Signals Are Shifting in 2026", 29 January 2026; and "Arcati 2026 | Workforce and Operating Constraints", 19 February 2026.
13. The Futurum Group, *2024 Global Mainframe Skills Report: Insights from Industry and Educational Experts*, commissioned by IBM, Broadcom and 21CS (fieldwork 20 February–3 March 2024; 750 companies, 200 universities, 200 students). Vendor-commissioned.
14. IBM, press release, "IBM Launches Mainframe Skills Council with SHARE and Other Key Organizations to Support New Generations of Mainframe Talent", 6 March 2024.
15. Open Mainframe Project, *cobol-programming-course* repository, GitHub, accessed 25 September 2026; and S. Srinivasan and M. Bauer, "COBOL Programming Course 2024 Milestones & More", 6 March 2025.
16. IBM Documentation, *watsonx Code Assistant for Z 2.x*, "Code explanation" overview, accessed September 2026.
17. BMC, "Bridging the Knowledge & Skills Divide with AI-Powered Assistants", BMC blog, 24 October 2025. Vendor source.
18. F. L. Yaeger, *What's New in DFSORT*, IBM DFSORT Team, October 2010 (ICEPRMxx members, installation environments, ICETOOL DEFAULTS).
19. G. Avelino, L. Passos, A. Hora and M. T. Valente, "A Novel Approach for Estimating Truck Factors", *Proceedings of the 24th International Conference on Program Comprehension (ICPC)*, 2016.