

EXECUTIVE PAPER · LEGACY SYSTEMS

Extend Legacy Systems Before You Replace Them

Boards approve rewrites to escape old systems and often end up paying for both. Executives who fund the system that already works usually get better results for less money.

AUTHOR

Chris Watkins

SERVICE

Legacy Systems

FOR

CEO, CFO · 12 min read

AT A GLANCE

The problem. Old systems cost money to run and frighten the people who depend on them, so leadership teams approve replacement programs. Many of those programs run years late and far over budget, and the old system keeps running the whole time. Recent public cases include a $135 million quarterly sales hit at a food manufacturer and a government-owned retailer whose replacement budget grew from £180 million to £1.1 billion before it was abandoned.

Why it persists. A rewrite looks like a single clean decision with a finish date. The risk sits in the years between approval and switch-off, when the company pays for two systems and the new one has not yet earned anything.

The recommendation. Treat the working system as an asset. Document it, put a stable interface around it, and add new capability at the edges. Replace it piece by piece when the business case holds, with a funded retirement plan that has milestones and a switch-off date.

The Decision in Front of You

Somewhere in your company there is a system everyone complains about and nobody can turn off. It might be an order platform written twenty years ago, a heavily customized ERP, or a billing engine whose original authors have retired. A vendor, a consultant or your own team has proposed replacing it.

The choice on the table looks binary: keep paying to run the old thing, or fund a new one. A third option usually costs less and carries less risk. Keep the old system running, make it understandable, wrap it so new work can be built around it, and retire it in stages.

The stakes are large. US federal agencies report that about 80% of their IT spending, which exceeds $100 billion a year, goes to operating and maintaining existing systems [1]. Private companies carry the same pattern on a smaller scale. The money is already committed to the old system. The decision is whether the next dollar buys a bet on a replacement or buys steady improvement on something that already works.

What's Happening

Old systems are expensive, and that part is true

The case against legacy systems is real. In July 2025 the US Government Accountability Office reviewed 69 federal legacy systems and singled out 11 as most in need of modernization. Those 11 range from 23 to 60 years old and cost about $754 million a year to operate and maintain [1][2]. Eight of them use

languages such as COBOL and assembly, and GAO notes a dwindling pool of people with the skills to support them [2]. Seven have known security vulnerabilities that cannot be fixed without modernization [2].

GAO ran a similar exercise in 2019 and found the same pattern: outdated languages, unsupported hardware and missing modernization plans [3]. Six years later, two of the 11 systems in the 2025 review still had no modernization plan at all, and only three had plans with milestones, a description of the work, and a plan for retiring the old system [1][2].

Replacements fail in expensive and public ways

GAO's own background material shows how hard full replacement is. The Department of Veterans Affairs abandoned three attempts to replace its health records system after 2001, citing planning problems, high costs and long timelines. Its fourth attempt, begun in 2017, paused deployments in April 2023 [2].

The private sector has its own recent examples.

Lamb Weston, 2024. The frozen potato maker moved its North American systems to a new ERP at the start of its fiscal third quarter. The company reported that the transition cut net sales by about $135 million in that quarter and reduced net income by about $72 million after tax [4]. Of a $95 million hit to adjusted EBITDA, about $55 million came from orders it could not fill [4]. It lowered its full-year sales guidance at the same time, citing the ERP impact along with weaker restaurant traffic [4].

Zimmer Biomet, 2024 and 2025. The medical device maker went live on a new SAP system in North America on July 4, 2024. In a lawsuit filed in September 2025, it alleges that it could not ship or receive product, issue invoices, or produce basic sales reports through the third quarter of 2024 [5]. It claims more than $172 million in damages, and says implementation fees grew from an agreed $69 million to $94 million [5][6]. Deloitte, the implementer, denies the claims and has moved to dismiss; the case was unresolved at the end of 2025 [6].

Birmingham City Council, 2022 to 2026. England's largest local authority replaced a 1999 SAP system with Oracle Fusion and went live in April 2022. Its external auditor reported in February 2025 that the project would cost at least £90 million more than budgeted, and that the council would not finish reimplementing the system until 2026 or later [7]. The auditor found the go-live risk was poorly understood and that the program had customized the new system to mirror old processes [7]. The old SAP system had cost about £5.1 million a year to run [7].

The Post Office: a rewrite that became a five-year extension

The UK Post Office case shows how a replacement program turns into the most expensive way to keep a legacy system alive.

The Post Office set out to replace its Horizon branch system, in service since 1999, with an in-house build called New Branch IT. The budget grew from £180 million to £1.1 billion, and the target date slipped from 2025 to as late as 2030 [8]. A 2024 government review rated the program "currently unachievable," citing weak governance, poor code quality, low test coverage and a team without large-scale delivery experience [9]. Reviewers also found the program had been framed as a copy of Horizon instead of a business change with clear outcomes [9].

In May 2025 the Post Office dropped the in-house build and opened a £492 million procurement for outside suppliers [10]. Meanwhile it kept extending the old contract: a one-year extension to March 2026 cost £63 million, bringing the total value of the Horizon contract since 1999 to £2.44 billion [11].

SIDEBAR: THE TWO-SYSTEM TAX

Every replacement program carries a hidden line item: the old system's full running cost for every month the new one is late. Birmingham's business case assumed the legacy SAP platform would cost £46 million over nine years [7]. The Post Office paid £63 million for a single year of Horizon while its replacement was rethought [11]. Each month of schedule slip adds a month of this tax.

Why the Usual Approach Fails

Rewrites fail for reasons that have little to do with the technology chosen.

Start with the specification. The old system is the specification. Twenty years of edge cases, regulatory fixes and customer workarounds live in its code and nowhere else. A team that sets out to copy it has to rediscover all of that, usually by breaking something in production. Birmingham and the Post Office both fell into this trap, rebuilding old processes in new software [7][9].

Value also arrives late. A rewrite delivers nothing until cutover, so the business funds years of spending with no return and no early signal of whether the program is on track. GAO notes that missing or incomplete plans raise the risk of overruns, delays and outright failure [3]. When the bad news arrives, it arrives all at once, close to go-live.

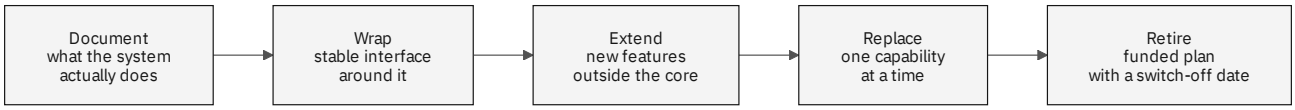
Culture compounds both problems. Large programs build momentum, and people stop reporting problems. Birmingham's auditor described a culture in which bad news was unwelcome or hard to raise [7]. The Post Office review found leadership and the delivery team did not trust each other [9].

None of this means old systems should run forever. GAO's central recommendation is that agencies write modernization plans with milestones and a disposition plan for the old system [1]. GAO is asking for a retirement schedule, which can be met one capability at a time.

The Framework: The Extend-First Ladder

The Extend-First Ladder treats the legacy system as a working asset and climbs toward retirement one step at a time. Each rung delivers value on its own, so the program can stop at any rung without wasting what was spent.

Exhibit 1. The Extend-First Ladder: each rung pays for itself before the next is funded



Source: author's framework, drawing on the strangler fig pattern as described in AWS Prescriptive Guidance [12] and GAO's modernization plan elements [1].

Document. Write down what the system does, who depends on it, and which business rules live only in its code. Capture its inputs and outputs in automated tests that record today's behavior. This is the cheapest risk reduction available, and every later step depends on it.

Wrap. Put a stable interface in front of the system, usually an API layer, so other systems talk to the interface instead of to the old code directly. New channels, reports and partners connect to the wrapper. The old system stops being a bottleneck for every new project.

Extend. Build new capability outside the core: a customer portal, an analytics feed, an AI assistant that reads the system's data. The business gets new features this quarter without touching the riskiest code.

Replace. When a specific capability justifies it, rebuild that capability behind the wrapper and route traffic to the new version. This is the strangler fig pattern, named by Martin Fowler after a vine that grows around a host tree until the tree is no longer needed [12]. AWS's guidance describes it in three steps: build the new component, run it alongside the old one with the old one kept for rollback, then remove the old function once traffic has moved [12]. AWS also notes its limits. It does not suit small, simple systems, it needs a place where requests can be intercepted, and the routing layer must be designed so it does not become a single point of failure [12].

Retire. Once most capability has moved, fund the switch-off as its own project with a date, an owner and a budget. GAO's three plan elements apply directly: milestones, a description of the work, and a disposition plan for the old system [1].

What the two paths cost

The following is an illustrative model for a hypothetical mid-size distributor whose order platform costs $1.2 million a year to run. Substitute your own figures.

Exhibit 2. Illustrative three-year cost: extending first costs less, even before a rewrite slips

Item (three years)	Full rewrite, on plan	Full rewrite, 50% overrun and one-year slip	Extend-first ladder
Legacy run cost	$3.6M	$4.8M	$3.6M
Build or change cost	$6.0M	$9.0M	$4.0M
Total cash	$9.6M	$13.8M	$7.6M
First new capability in production	Year 3	Year 4	Quarter 2
Legacy system retired	End of year 3	End of year 4	Core modules by year 3; full retirement planned for year 4 or 5

Source: author's illustrative estimate. Assumes $1.2M annual legacy run cost; a $6.0M rewrite delivered over three years; a ladder spend of $1.0M in year 1 for documentation, tests and an API wrapper, then $1.5M a year in years 2 and 3 to replace modules behind the wrapper. Legacy run cost is held flat for simplicity; in practice it falls as modules retire on the ladder path. The overrun case is a stress test; the Post Office budget grew more than sixfold [8].

The ladder does not finish retirement inside three years, and the table says so. Its advantage is that money starts returning value in the second quarter, each step is small enough to stop, and the company never has to bet the business on a single go-live weekend.

Costs, Risks, and Tradeoffs

Extending first is cheaper in most cases. It is also slower to reach a fully modern estate, and it can become an excuse to never finish.

Exhibit 3. Tradeoffs of the extend-first approach

Area	Cost or risk	Mitigation
Two architectures at once	Engineers maintain old code and new services side by side for years	Fund the retirement rung explicitly; track the share of traffic still on legacy each quarter
Skills	Old languages need scarce, expensive people, as GAO found [2]	Pair senior legacy staff with newer engineers while documenting; capture rules in tests
Wrapper risk	The interface layer becomes a bottleneck or single point of failure [12]	Build it with the same availability standard as the core; load test it
Permanent half-finish	Extension becomes the strategy and nothing retires	Write a disposition plan with a date and owner at the start [1]
Vendor support end	Unsupported hardware or software raises security exposure [2]	Isolate unsupported components behind the wrapper; prioritize them for replacement
When this is wrong	The system is small, its rules are simple and well understood, it cannot be wrapped, or the vendor is ending support with no extension available	Replace it directly, with a rehearsed cutover and a rollback path

Source: author's analysis.

How to Act

Exhibit 4. A one-year roadmap

Window	Actions	Owner	Milestone
First 30 days	Inventory systems that are old, expensive or blocking new work. For each, record annual run cost, the business processes it supports, and who understands it. Freeze approval of any full rewrite until this is done.	CFO with CIO or CTO	Inventory signed off with run cost and key-person risk per system
First 90 days	For the top two systems, document business rules and capture current behavior in automated tests. Design the wrapper interface.	Engineering leads with business process owners	Test coverage on the top 20 business transactions; interface design approved
First 90 days	Write a disposition plan for each system with milestones, scope of work and a target retirement date, using GAO's three elements [1].	CIO or CTO	Plans approved by the executive committee with a budget line for retirement
First 365 days	Put the wrapper into production. Route at least one new channel or report through it. Replace one capability behind the wrapper and switch it off in the old system.	Engineering leads	One legacy capability retired; new feature delivered without changing legacy code
First 365 days	Report quarterly on legacy run cost, share of transactions still on legacy, and retirement milestones hit.	CFO	Run cost trend and retirement progress reported to the board

Source: author's recommendation.

Questions to Ask Your Team

1. What does each of our oldest systems cost to run each year, including staff and contractors?
2. Who understands the business rules inside it, and what happens if they leave?
3. Do we have automated tests that describe what the system does today?
4. Could we put an interface in front of it so new work stops touching the old code?
5. What is the smallest piece of it we could replace first, and what would that return?
6. In our replacement business case, how many months of double running did we assume, and what happens to the case if that doubles?
7. Is there a written plan, with a date and an owner, for switching the old system off?
8. When was the last time someone on this program raised bad news, and what happened?

Closing

A working legacy system holds years of business knowledge that no requirements document captures. Rewriting it from scratch puts that knowledge, and the company's cash, at risk on a single go-live. Document the system, wrap it, extend it at the edges and replace it a capability at a time. Fund its retirement as a real project with a date. The company keeps earning from the old system while the new one proves itself, and the board can measure progress every quarter.

Sources

[1] U.S. Government Accountability Office. "Information Technology: Agencies Need to Plan for Modernizing Critical Decades-Old Legacy Systems." GAO-25-107795. 2025. https://www.gao.gov/products/gao-25-107795.

[2] U.S. Government Accountability Office. "Information Technology: Agencies Need to Plan for Modernizing Critical Decades-Old Legacy Systems." GAO-25-107795, full report. 2025. https://files.gao.gov/reports/GAO-25-107795/index.html.

[3] U.S. Government Accountability Office. "Information Technology: Agencies Need to Develop Modernization Plans for Critical Legacy Systems." GAO-19-471. 2019. https://gao.gov/products/gao-19-471.

[4] Lamb Weston Holdings, Inc. Form 8-K, Exhibit 99.1, third quarter fiscal 2024 results. U.S. Securities and Exchange Commission. 2024. https://www.sec.gov/Archives/edgar/data/1679273/000167927324000017/lw-20240225x8kxexx9913q24.htm.

[5] MassDevice. "Zimmer Biomet sues Deloitte for $172 million." September 2025. https://www.massdevice.com/zimmer-biomet-sues-deloitte-for-172-million/.

[6] The Journal Gazette. "Zimmer Biomet lawsuit against Deloitte Consulting continues to move through New York court." December 2025. https://www.journalgazette.net/business/zimmer-biomet-lawsuit-against-deloitte-consulting-continues-to-move-through-new-york-court/article_51a2bb79-4ff5-4b18-846d-ab266d6ade14.html.

[7] Computer Weekly. "Auditor: 2026 till Birmingham recovers from botched Oracle project." Coverage of Grant Thornton's February 2025 report to Birmingham City Council. 2025. https://www.computerweekly.com/news/366619557/Auditor-2026-till-Birmingham-recovers-from-botched-Oracle-project.

[8] Computer Weekly. "Fujitsu set for further £180m deal as Post Office Horizon replacement delayed." 2024. https://www.computerweekly.com/news/366587174/Fujitsu-set-for-further-180m-deal-as-Post-Office-Horizon-replacement-delayed.

[9] Computer Weekly. "Can the Post Office project to replace Horizon be rescued?" Coverage of the Infrastructure and Projects Authority and Public Digital reviews. 2024. https://computerweekly.com/news/366612938/Can-the-Post-Office-project-to-replace-Horizon-be-rescued.

[10] TechRadar Pro. "Post Office confirms end of in-house NBIT system, £410M procurement plan to go ahead." May 2025. https://www.techradar.com/pro/post-office-confirms-end-of-in-house-nbit-system-gbp410m-procurement-plan-to-go-ahead.

[11] The Register. "£136M government grant saves troubled Post Office from suboptimal IT." April 2025. https://www.theregister.com/2025/04/25/136_million_government_grant_saves/.

[12] Amazon Web Services. "Strangler fig pattern." AWS Prescriptive Guidance: Decomposing monoliths into microservices. https://docs.aws.amazon.com/prescriptive-guidance/latest/modernization-decomposing-monoliths/strangler-fig.md.

The views in this paper are the author's own and do not represent any employer. Scenarios are hypothetical. This paper is not legal, financial, or compliance advice.