

EXECUTIVE PAPER · INFRASTRUCTURE

If It Lives in One Person's Head, It Is a Risk

Boards price customer concentration and supplier concentration. Few price the engineer who is the only one who can restart billing. Here is how to find that exposure and retire it.

AUTHOR

Chris Watkins

SERVICE

Infrastructure

FOR

CEO, COO · 13 min read

AT A GLANCE

The problem. Most companies run at least one critical system that only one person fully understands. When that person is unreachable or gone, recovery slows from hours to days. Experienced technical staff are retiring and leaving, and the knowledge goes with them.

Why it persists. The dependency is invisible while the person is present. Their speed makes them look like an asset on every review, and documentation never wins a sprint. Incident reports describe the technical fault and rarely mention who was missing.

The recommendation. Put key-person dependency on the risk register with a dollar figure. Require every critical system to pass a handback test: a second trained person, a runbook that someone else has executed, credentials held by the company, and a timed drill each year. Fund the work as continuity spending and report progress to the board.

The Decision in Front of You

Every operations leader knows the name. It belongs to the engineer who built the order integration, the administrator who set up the payment gateway, or the contractor who is the only person who has ever patched the warehouse system. When something breaks at 2 a.m., the plan is to call that person.

The decision is whether to treat that arrangement as a continuity risk the board sees priced. This paper argues it should. A business that would not accept 60% of revenue from one customer is often accepting 100% of recovery capability from one employee, with no contract, no notice period that matters during an outage, and no backup.

Outages last longer when the one expert is unavailable, and a departure can turn into months of reverse engineering. Concentrated access adds a security exposure, since one compromised or exhausted person can become the way in.

What's Happening

Experienced staff are leaving, and few companies capture what they know

Uptime Institute's December 2025 staffing analysis found enterprises were more than five times as likely as colocation providers to lose an employee with over ten years of tenure, and it attributed many of those departures to retirement [1]. Nearly a quarter of respondents named staff retention as their single greatest challenge, and formal mentorship offerings at enterprises fell 12 percentage points in a year [1].

A 2025 survey of 1,000 professionals by APQC found 92% of organizations do not consistently capture knowledge from employees nearing retirement, and 58% of C-suite respondents reported serious concern about knowledge loss [2]. The study was sponsored by a knowledge-management software vendor, so read the figures as directional.

Government systems show where this ends. The U.S. Government Accountability Office reported in July 2025 that 11 critical federal legacy systems ranged from 23 to 60 years old, and eight of them used outdated languages, including COBOL and assembly code at Treasury [3]. GAO noted a dwindling number of people with the skills to support those languages. Agencies spend about 80% of more than $100 billion in annual IT spending on operating and maintaining existing systems [3]. Private companies carry smaller versions of this in ERP customizations and scripts nobody else has read.

The open-source supply chain carries the same risk, at scale

Most companies also depend on software maintained by one or two people outside their walls. A 2024 Tidelift survey of more than 400 maintainers found 60% were unpaid and 60% had quit or considered quitting their maintenance work [4].

xz Utils, March 2024. XZ Utils, a compression library in most Linux distributions, had for years one maintainer, Lasse Collin. In June 2022 he told a public mailing list he was dealing with long-term mental health issues and that a newer contributor, Jia Tan, might take a bigger role [5]. By 2023 Tan was merging code. In February 2024, versions 5.6.0 and 5.6.1 shipped with a backdoor that targeted remote login on Linux servers, rated 10.0, the maximum severity, by NIST's National Vulnerability Database [5][6]. Andres Freund, a developer at Microsoft, found it while chasing a small performance slowdown and disclosed it on March 29, 2024 [5][7]. Experts said it was caught before the affected version was widely deployed [5]. Freund said the industry got unreasonably lucky [5]. The attacker had posed as a helpful volunteer to an overloaded single maintainer and earned trust over months before inserting the code [5].

Ingress NGINX, November 2025 to March 2026. Ingress NGINX routes outside traffic into Kubernetes clusters. In November 2025 the Kubernetes project announced its retirement. The maintainers wrote that only one or two people had done development work, on their own time after work hours and on weekends, and that even announcing the wind-down failed to attract more help [8]. Best-effort maintenance ended in March 2026, with no security patches after that [8]. An industry report cited by The New Stack found half of Kubernetes users still ran it in 2025, a year after a critical flaw rated 9.8 was disclosed in it [9]. Each of those users inherited a migration on a schedule set by two volunteers' capacity.

npm, September 2025. An attacker phished the account of a single maintainer of 25 popular JavaScript packages, including two called chalk and debug, which together see hundreds of millions of downloads a week [10]. The malicious versions tried to steal data and rewrite cryptocurrency wallet addresses in browsers [10].

A case worth knowing: the UK air traffic failure

On August 28, 2023, the UK's national flight planning system for air traffic control failed. The Civil Aviation Authority's independent review, published in November 2024, estimated more than 700,000 passengers were affected, about 300,000 of them through cancellations [11]. The trigger was a flight plan with duplicate waypoint names, a combination the system had never encountered [12]. Press coverage put the cost to the aviation industry at about £100 million [12].

Much of the delay traced to staffing. The senior engineer able to fix it was rostered on call from home on one of the busiest days of the year. The engineer was reached 34 minutes after the failure, could not log in remotely, and arrived on site about three hours and 15 minutes after the incident began [12]. The system's manufacturer was not contacted for more than four hours, and once called it found a fix within 30 minutes [12]. The review's 34 recommendations include NATS reviewing its contingency and engineering resource management [11].

SIDEBAR: ATTRITION NEVER APPEARS IN THE ROOT CAUSE

Lorin Hochstein, a reliability engineer who writes about incident analysis, observed in November 2025 that staff departures almost never appear in incident write-ups [13]. Public reports exist to reassure customers that a technical fault is fixed. Internal reviews look for technical causes. A team that lost its most experienced engineer six months earlier will record the failed configuration change and omit why nobody caught it, so the risk stays out of the data leadership reviews.

Why the Usual Approach Fails

Most companies respond to key-person risk by asking the expert to write things down. That request rarely succeeds.

The expert is rewarded for speed. Every hour spent documenting is an hour not spent on the project leadership is watching. Documentation has no customer, so nobody checks it, and a wiki page written once decays as the system changes while people go on trusting it.

Credentials concentrate quietly. The person who built a system often holds the only administrator account, the only hardware token, or the only relationship with the vendor's support engineer. NATS showed how much time a login problem and an unmade vendor call can cost [12].

The standards already expect more. NIST SP 800-34 Rev. 1, the U.S. federal contingency planning guide, says recovery teams should be sized to stay viable when some members are unavailable, with a designated alternate for each team leader [14]. It says staff should be trained well enough to carry out their recovery roles without the plan document in hand, at least once a year, and that contact lists need more frequent review than the rest of the plan [14]. The guide dates from 2010 and remains NIST's

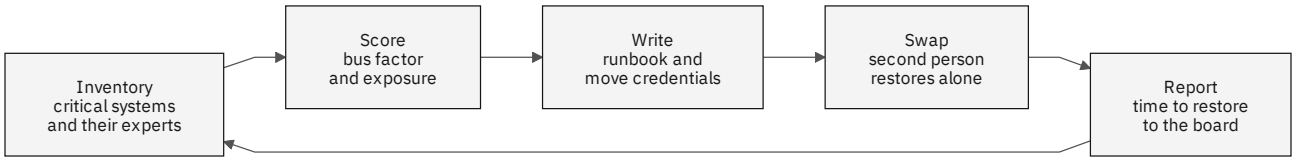
current version. ISO 22301:2019, the international business continuity standard, requires a documented management system that is maintained and improved [15], and auditors expect current evidence that the people named in the plan are competent to carry it out [16].

The Framework: The Handback Test

A system is handback-ready when the company could take it back from its current owner tomorrow and keep running. The test has five checks, and a system passes only when all five are true.

1. **A second person.** At least two named people can operate and restore the system, and they do not share a manager, a shift, or a vacation calendar.
2. **A runbook someone else has run.** Restore, restart, and failover steps are written down and have been executed by the second person, with the author watching and not touching the keyboard.
3. **Company-held access.** Every credential, token, certificate, and vendor support contract sits in a company vault or account. No production access depends on one person's phone or personal email.
4. **A known escalation path.** The runbook names the vendor, the contract number, and the point at which to call them. NATS lost hours before calling a manufacturer who fixed the problem in 30 minutes [12].
5. **A timed drill.** Once a year the second person restores the system while the primary owner is unreachable, and the time to restore is recorded.

Exhibit 1. The Handback Drill: knowledge counts only after a second person has used it



Source: author's framework, drawing on NIST SP 800-34 Rev. 1 guidance on alternates, training, and plan maintenance [14].

Documentation that only its author has executed is an untested backup. The swap drill reveals the missing step, the expired password, and the server nobody listed.

Putting a price on it

The model below is illustrative for a hypothetical regional distributor whose order integration is maintained by one senior engineer. Replace each assumption with your own.

Exhibit 2. Illustrative annual exposure from one key-person dependency

Assumption or result	Undocumented, one expert	Handback-ready, two people
Revenue through the system per operating hour	$40,000	$40,000
Share of that revenue lost or penalized during an outage	50%	50%
Incidents per year that need deep system knowledge	4	4
Probability the expert is unavailable when needed	20%	20%
Extra outage hours when the expert is unavailable	16	2
Expected annual outage cost from absence	$256,000	$32,000
One-time cost to rediscover the system if the expert leaves	$150,000	$20,000
Annual probability the expert leaves	15%	15%
Expected annual departure cost	$22,500	$3,000
Total expected annual exposure	**$278,500**	**$35,000**
First-year cost of mitigation		$34,400

Source: author's illustrative estimate. Assumes $400,000 of orders per 10-hour operating day; expected outage cost = incidents × probability × extra hours × $20,000. Rediscovery assumes 600 contractor hours at $250 per hour. Mitigation assumes 160 hours of the expert's time and 120 hours of a second engineer's time at $100 per loaded hour, plus four 8-hour quarterly drills for two people at the same rate.

On these assumptions the first year of mitigation costs about $34,000 and reduces expected annual exposure by about $243,500. The probabilities are the softest inputs. If the expert is unavailable half as often, the reduction is still about $131,500 a year.

Costs, Risks, and Tradeoffs

Exhibit 3. Tradeoffs of requiring handback-ready systems

Area	Cost or risk	Mitigation
Expert time	The people who know the most are pulled off project work to write and teach	Budget it as continuity spending; time-box each system to a few weeks
Second-person cost	Cross-training ties up two salaries on one system	Pair systems so each engineer is primary on one and secondary on another
Stale runbooks	Documentation rots and people trust it anyway	Fail any runbook not executed in the last 12 months; update after every incident
Morale	Experts can read the program as a threat to their job security	Frame it as freeing them from 2 a.m. calls; reward teaching in reviews
Open-source exposure	Your own runbooks do nothing for a single-maintainer dependency upstream	Inventory critical dependencies; fund or replace the thin ones before they retire
When this is wrong	A small firm on standard SaaS with no custom systems has little to hand back	Rely on vendor support contracts and keep admin credentials in a shared company vault

Source: author's analysis.

The approach also slows the organization in the short term. A team of five that spends a quarter cross-training will ship less that quarter. Show that cost on the same page as the exposure it retires.

How to Act

Exhibit 4. A one-year roadmap

Window	Actions	Owner	Milestone
First 30 days	List the systems that would stop revenue, payroll, or regulatory reporting. For each, name everyone who can restore it today. Find every production credential held by a single person.	COO with the CTO or IT director	Register of critical systems with a bus-factor count for each; single-holder credentials listed
First 30 days	Add key-person dependency to the enterprise risk register with an estimated annual exposure.	CFO or risk lead	Line item on the register, reviewed by the audit or risk committee
First 90 days	Move every single-holder credential and vendor contract into company-controlled accounts. Write runbooks for the top five systems. Name a second person for each.	CTO with system owners	Zero production credentials held by one person; five runbooks drafted
First 90 days	Run the first swap drill on the highest-exposure system with the primary owner unreachable.	Operations lead	Measured time to restore by the second person, with gaps logged
First 90 days	Inventory open-source and vendor components with one or two maintainers that critical systems depend on.	Engineering lead with security	Thin-dependency list with a replace, fund, or accept decision for each
First 365 days	Bring every critical system to a passing handback test. Add a "who was missing" question to every incident review. Report bus factor and drill times to the board twice a year.	COO with HR and the CTO	No critical system with a bus factor of one; every system drilled in the last 12 months

Source: author's recommendation.

Questions to Ask Your Team

1. Which systems would stop if one specific person was unreachable for a week?
2. For each critical system, how many people have restored it in the last year, and when?
3. Which production passwords, tokens, or vendor accounts belong to an individual instead of the company?
4. When did someone other than the author last follow one of our runbooks from start to finish?
5. Who on the team is within five years of retirement, and what do they know that nobody else does?
6. Which outside software do we depend on that is maintained by one or two volunteers?
7. In our last three incident reviews, did anyone ask who was unavailable and whether it mattered?

Closing

A company that discovered 100% of its revenue came from one customer would fix it before the next board meeting. Many companies carry the same concentration in the people who can restore their systems, and the exposure rarely appears on any report. Count it, price it, and require every critical system to pass a handback test. When the person who holds the knowledge leaves, the system should stay behind and keep running.

Sources

[1] Uptime Institute Intelligence. Staffing survey analysis. UII Update 446. December 2025. https://intelligence.uptimeinstitute.com/resource/survey-highlights-industry-staffing-crisis.

[2] APQC, sponsored by eGain. Press release on "Navigating the Great Retirement with KM & AI" survey. Business Wire. October 13, 2025. https://www.businesswire.com/news/home/20251013139432/en.

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

[4] Tidelift. "2024 State of the Open Source Maintainer Report." Press release via Business Wire. September 17, 2024. https://www.businesswire.com/news/home/20240917030299/en.

[5] Insurance Journal (Reuters). "Why Near-Miss Cyberattack Put US Officials and Tech Industry on Edge." April 8, 2024. https://amp.insurancejournal.com/news/national/2024/04/08/768370.htm.

[6] National Institute of Standards and Technology, National Vulnerability Database. "CVE-2024-3094." 2024. https://nvd.nist.gov/vuln/detail/CVE-2024-3094.

[7] Freund, A. Disclosure of the xz/liblzma backdoor. oss-security mailing list, Openwall. March 29, 2024. https://www.openwall.com/lists/oss-security/2024/03/29/4.

[8] Kubernetes SIG Network and Security Response Committee. "Ingress NGINX Retirement: What You Need to Know." Kubernetes Contributors blog. November 2025. https://kubernetes.dev/blog/2025/11/12/ingress-nginx-retirement.

[9] Vaughan-Nichols, S. J. Report on the Kubernetes retirement of Ingress NGINX. The New Stack. February 3, 2026. Citing Isovalent's 2025 networking report. https://thenewstack.io/kubernetes-to-retire-ingress-nginx/.

[10] Endor Labs. "Major supply chain attack compromises popular npm packages including chalk and debug." September 2025. https://endorlabs.com/learn/major-supply-chain-attack-compromises-popular-npm-packages-including-chalk-and-debug.

[11] UK Civil Aviation Authority. "Aviation regulator publishes independent review into August 2023 NATS flight planning system failure." November 14, 2024. https://www.caa.co.uk/newsroom/news/aviation-regulator-publishes-independent-review-into-august-2023-nats-flight-planning-system-failure/.

[12] ITV News. "Engineer working from home with password issue contributed to massive air traffic control failure." Coverage of the CAA independent review. November 15, 2024. https://itv.com/news/2024-11-15/remote-engineer-with-password-issue-contributed-to-air-traffic-control-failure.

[13] Hochstein, L. "You'll never see attrition referenced in an RCA." Surfing Complexity. November 2, 2025. https://surfingcomplexity.blog/2025/11/02/youll-never-see-attrition-referenced-in-an-rca/.

[14] National Institute of Standards and Technology. "Contingency Planning Guide for Federal Information Systems." NIST SP 800-34 Rev. 1. 2010. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-34r1.pdf.

[15] International Organization for Standardization. "ISO 22301:2019 Security and resilience: Business continuity management systems: Requirements." Edition 2. 2019. https://www.iso.org/standard/75106.html.

[16] ISMS.online. "ISO 22301 Clause 7: Support." Undated web page. https://www.isms.online/iso-22301/clause-7-support/.

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.