

A WHITE PAPER FOR SOFTWARE ENGINEERING AND TECHNOLOGY LEADERS

QA Continuity:

The Risk Every Engineering Leader Inherits and Rarely Gets to Measure

Why the stability of your QA vendor's team affects release risk, audit readiness, and your credibility as a leader, with a practical framework for measuring it before your next renewal.



Executive Summary

For most software engineering leaders, the QA vendor relationship is managed the way a facilities contract is managed: reviewed at renewal, scored on rate and current staffing, and otherwise left alone. That approach misses the one variable that predicts whether a QA engagement will still be protecting the release calendar a year from now, which is continuity.¹

This white paper draws on SQA²'s work on QA vendor continuity, including the Continuity Score framework introduced on sqasquared.com,² to make the case that continuity is operational infrastructure. It has a direct, measurable connection to release risk, audit readiness, and an engineering leader's credibility with their superiors. The paper puts numbers on what continuity failures cost, and it includes the seven questions behind the Continuity Score, which any engineering leader can use to assess a current QA vendor before the next renewal.

The Hidden Cost of Losing Institutional Knowledge

When a new QA engineer joins an account, they inherit credentials, whatever documentation happens to exist, and a calendar full of knowledge-transfer meetings. What they do not inherit is judgment, the accumulated sense of where a system's edge cases live, which modules are fragile under load, and which defects have already been triaged and deliberately left alone. That kind of knowledge only transfers through time spent on the system, and while it is being rebuilt, two costs accrue that rarely show up on an invoice.

The first is engineering hours. Senior engineers on the client side, your own team, end up re-explaining systems they already explained to the last person who rotated off the account. The second, and the more expensive one, is defect risk: the ramp-up period is exactly when edge cases slip through, precisely because the person testing them does not yet know they are edge cases.

Call this the onboarding tax. It is paid every time a QA vendor rotates staff onto an account, and because it is paid in hours and near-misses instead of as a line item, most engineering organizations never total it up.

What the Onboarding Tax Adds Up To

The math is simple enough to run for a single team. The example below uses conservative assumptions, and every figure in it is meant to be replaced with your own.

The example assumes a 10-person QA team that loses 20% of its people every six months, which works out to four rotations a year.³ This is the same scenario as the stress test later in this paper. Each replacement works at about half productivity until fully ramped.⁴ During that period, a senior engineer on your side and a teammate on the existing QA team each spend five hours a week helping the replacement get up to speed.⁵ The example compares two ramp times. Two weeks is fast for most accounts. Six weeks matches the first answer to the first question of the Continuity Score, where a new engineer takes a month or more to become productive. Hourly costs are based on Bureau of Labor Statistics wage data for May 2025, the most recent available, with benefits and overhead added.⁶

Cost	Two-week ramp	Six-week ramp
Lost QA productivity per rotation	40 hours × \$70 = \$2,800	120 hours × \$70 = \$8,400
Senior engineer time per rotation	10 hours × \$92 = \$920	30 hours × \$92 = \$2,760
QA teammate time per rotation	10 hours × \$70 = \$700	30 hours × \$70 = \$2,100
Total cost per rotation	\$4,420	\$13,260
Annual onboarding tax	4 rotations × \$4,420 = \$17,680	4 rotations × \$13,260 = \$53,040

The gap between the two columns is more than \$35,000 a year for one team, and it comes entirely from how long replacements take to reach full productivity. That time depends on what the vendor has documented and who it can send, and both show up directly in the Continuity Score. Neither figure includes the defect risk that runs alongside the ramp, and a later section puts a price on it.

Why Continuity Keeps Failing

Staff turnover is a certainty. People get promoted, change employers, and move teams. What matters is what most vendor models do when staff change, and the usual answer is nothing. If everything a quality engineer has learned about a client's system lives only in that person's head, then that person leaving is a knowledge-loss event dressed up as a routine staffing update.

Subcontracting layers make this worse. A vendor that subcontracts testing work to a secondary firm, common practice in QA staffing, adds a layer of turnover risk the client never sees directly, along with a harder question: who else now has access to test data, credentials, and, in regulated environments, to PHI or cardholder data that passed through a test environment? Every additional subcontracting layer is a party your compliance team has to account for, whether or not anyone told them it existed. It is the same dynamic behind a statistic worth keeping in view: third-party involvement in data breaches jumped from 15% to 30% year over year, according to Verizon's 2025 Data Breach Investigations Report.⁷ A QA vendor's own subcontractors are part of that surface.

What a Continuity Failure Costs

The onboarding tax is the cost an engineering leader can see if they go looking for it. The larger costs come from what slips through while a team is rebuilding what it knew, and those costs are well documented outside of QA.

The first is the cost of a late defect. A widely cited IBM Systems Sciences Institute estimate puts the cost of fixing a defect found in testing at roughly 15 times the cost of catching it in design, and a defect found in production at roughly 100 times.⁸ The exact multipliers vary by study, but the direction does not, and Capers Jones's published research on software defect economics shows the same steep rise in repair cost the later a defect is found.⁹ A QA engineer who is still learning where a system's edge cases live is more likely to let a defect move one stage further down that curve, and each stage multiplies what it costs to fix.

A defect that reaches production can also cost the business directly through downtime. In ITIC's 2024 Hourly Cost of Downtime survey, 97% of large enterprises said a single hour of downtime costs them more than \$100,000, and 41% put that hour at \$1 million to more than \$5 million. In regulated industries, including banking and financial services, the average hourly cost exceeded \$5 million.¹⁰

The last cost comes from access. Subcontracting layers widen the set of parties with access to test data and credentials, and every person who rotates through an account is someone whose access has to be granted, tracked, and revoked on time. IBM's 2025 Cost of a Data Breach Report puts the global average cost of a breach at \$4.44 million and the average time to identify and contain one at 241 days.¹¹ At that pace, an access gap left behind by a departed contractor could go unnoticed for most of the following year.

None of these figures measures continuity directly. They put a price on the failures that continuity is meant to prevent, which is why continuity belongs in a renewal conversation alongside rate.

Redefining Continuity

The instinct is to solve for continuity by demanding that the same three names stay on the account for the life of the contract. That is not realistic, and it misreads what continuity means. Continuity means that when the faces on the account change, you do not start over.

“The team can change, as long as a change on the team costs the client nothing.”

That distinction matters because it turns continuity from a staffing promise, which any vendor can make and few can keep, into a structural property of how a QA function is built. That means documented system knowledge that outlives any individual, a bench deep enough to absorb a departure without pulling someone in cold, and a transition process that has been used often enough to be routine.

A Framework for Evaluation: The QA Vendor Continuity Score

Most QA vendor renewals are evaluated and re-signed without anyone testing for continuity directly. Rate gets negotiated, skills get checked against the current roadmap, and coverage numbers get reviewed. The question that predicts whether any of that holds up six months from now rarely gets asked: what happens to this account when someone leaves?

SQA² built the QA Vendor Continuity Score to make that question concrete. It is seven questions about your current QA vendor, and each one looks at what the vendor has built to survive a staffing change. For each question, note which answer best describes your vendor today. To get your score and what it means, take the scorecard at sqasquared.com/whats-your-qa-vendors-continuity-score.² It takes about two minutes, and your results are emailed to you with a breakdown of what drove them.

- 1. When someone new joins your account from the vendor, how long until they're productive?**
 - A** A month or more. Our engineers essentially retrain each one.
 - B** A few weeks, with some hand-holding.
 - C** Days, because they onboard from documented account knowledge.
- 2. Where does your vendor's knowledge of your system live?**
 - A** In individual people's heads.
 - B** Scattered notes and tickets, nothing maintained.
 - C** A maintained knowledge repository owned by the engagement.
- 3. If someone rotates off mid-release, what happens?**
 - A** We scramble. Releases have slipped because of it.
 - B** The vendor backfills, but the replacement starts near zero.
 - C** A defined transition. Knowledge is already captured, and the replacement often has prior exposure to our platform.
- 4. Who employs the people on your account?**
 - A** Honestly not sure. There may be subcontractors involved.
 - B** A mix of vendor employees and subcontractors.
 - C** The vendor's direct employees, with no layers.
- 5. Is QA your vendor's core business?**
 - A** No, it's one service line inside a broader staffing or IT firm.
 - B** It's a major practice among several.
 - C** Yes. QA is all they do.
- 6. If you needed two more Quality Engineers next month, who would show up?**
 - A** Whoever's available on the bench.
 - B** QA people, but new to our systems.
 - C** People who've already worked our account or platform.
- 7. Who is accountable, by name, for your quality outcomes?**
 - A** Nobody. Accountability is spread across resources.
 - B** An account manager, but not for quality outcomes specifically.
 - C** One named owner who answers for the release.

A vendor built for continuity can answer all seven questions easily, because the answers describe how it already operates.

A useful companion to the score is a stress test you can run without a vendor in the room. Assume 20% of the QA team turns over in the next six months. In SQA²'s experience, that is a plausible scenario for a vendor-staffed team, because people rotate between accounts as well as leave the company. Then ask three questions:

- Which releases on the current roadmap would slip?
- Who, specifically, absorbs the departing engineers' knowledge, and how?
- How long will it take their replacements to reach full productivity on this account?

If those three answers do not exist yet, the scorecard has already told you what you need to know.

What This Means for Engineering Leadership

Continuity risk rarely announces itself as a QA problem. It shows up as a release slipping with no clean explanation. It shows up in an audit, when the QA process cannot produce a consistent paper trail because the person who built it left eight months ago. It shows up as a defect that reaches production and traces back to a gap nobody flagged, because the engineer who would have flagged it had already rotated off the account.

For an engineering leader, that risk lands somewhere specific: on you. You are accountable for release velocity and for release safety at the same time, but you rarely have full visibility into whether the QA function underneath either one is actually stable. Continuity is one of the few variables you can assess directly, in a single conversation with a vendor, before it becomes an incident.

It is also one of the better answers to a question engineering leaders increasingly have to field upward: how do you know this is under control? “We track vendor continuity as part of our renewal process” is a materially stronger answer than “we’ve been happy with them so far,” and it is an answer that holds up in front of a CEO, a board, an auditor, or a CISO asking about third-party access to production-adjacent systems. It also gives QA a measurable, reportable risk posture, and that makes the function easier to defend in budget conversations.

Before Your Next Renewal

A continuity review does not require a new procurement process. It fits inside the renewal conversation you are already planning to have, and the Continuity Score gives it a structure.

1. **Score your vendor before the meeting.** Take the scorecard yourself, or with the engineering leads who work with the vendor day to day. If your answers differ from theirs, that difference is useful information on its own.
2. **Use your weakest answers as the agenda.** Every question where you chose the first or second answer becomes a question for the vendor, and you should ask them to answer each one specifically. A vendor with real answers will give you names, processes, and examples. A vendor without them will give you reassurance, and you will hear the difference.
3. **Ask one question the Score does not cover.** Find out how many people, across the vendor and any subcontracted layers, have touched your environment or your data in the past year. In a regulated environment, your compliance team needs that number, so it is worth knowing whether your vendor can produce it.
4. **Run the 20% scenario together, out loud.** Walk the vendor through the stress test from the previous section and see how specific their answers get.
5. **Write down what the vendor commits to.** Confident answers only protect you if someone holds the vendor to them at the next review, so treat them as part of the engagement record.

Nothing here requires distrust of your current vendor. The review treats continuity as a measurable property of the engagement, the same way you already measure defect escape rate or test coverage, because over the life of a contract it behaves exactly like one.

About SQA²

Continuity is built into SQA²'s engagement model. Embedded quality engineers are backed by a shared bench, system knowledge is documented so it survives any one departure, and client relationships have run five, eight, and in one case more than ten years without a restart.

SQA² also sets aside time every week to maintain an internal knowledge repository that captures what our engineers learn about each client's system. The repository becomes more valuable the longer an engagement runs, because every release and every staff transition adds to it. An engineer joining a five-year account starts with far more context than one joining a new account, and that context is why replacements on SQA² accounts typically reach full productivity in one to two weeks.

In a competitive RFP, one of the largest private banks in the United States selected SQA² as its QA partner. SQA² scored highest in every category the bank evaluated, including stability, accountability, ownership of quality outcomes, and rate. For a regulated institution, stability, accountability, and ownership decide whether a QA function holds up when the team changes, and the Continuity Score measures the same things. The result also shows that a vendor built for continuity does not have to cost more.

If you would like to walk through the Continuity Score against your own QA vendor, or talk through what a continuity-first QA engagement looks like in practice, we are glad to have that conversation.

sqasquared.com · Stability. Ownership. Judgment.

Notes and Sources

¹ SQA², "QA Continuity Is an Asset. An Overlooked One.," sqasquared.com/qa-continuity-is-an-asset-an-overlooked-one/

² SQA², "What's Your QA Vendor's Continuity Score?," sqasquared.com/whats-your-qa-vendors-continuity-score/

³ A 10-person team losing 20% of its people every six months loses two people every six months, or four a year.

⁴ At 40 hours a week, a two-week ramp is 80 working hours, and half productivity means 40 hours of lost work. A six-week ramp is 240 working hours, or 120 hours lost.

⁵ Five hours a week comes to 10 hours over a two-week ramp and 30 hours over a six-week ramp.

⁶ The QA engineer rate is the BLS May 2025 median hourly wage for software quality assurance analysts and testers (\$50.14), multiplied by 1.4 for benefits, payroll taxes, and overhead and rounded to \$70. The senior engineer rate is the median hourly wage for software developers (\$65.38), calculated the same way and rounded to \$92. Source: U.S. Bureau of Labor Statistics, "Occupational Employment and Wages, May 2025" (released May 2026).

⁷ Verizon, "2025 Data Breach Investigations Report" (third-party breach involvement statistic).

⁸ IBM Systems Sciences Institute, relative cost of defect repair by phase (widely cited estimate).

⁹ Capers Jones and Olivier Bonsignour, *The Economics of Software Quality* (Addison-Wesley, 2011).

¹⁰ ITIC, "2024 Hourly Cost of Downtime Survey."

¹¹ IBM, "Cost of a Data Breach Report 2025."