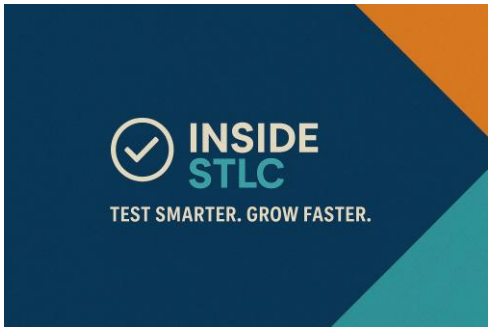
THE 5-POINT QA HIRING FRAMEWORK

25 Years of Hiring Decisions — Distilled into One Scorecard



Before You Read This

I used this framework in every QA interview I conducted for over two decades. It's not a tick-box exercise. It's the five lenses I used to answer one question: *Do I want this person on my team?*

Most candidates I rejected passed the technical screening. They knew the tools. They had the certificates. What they didn't have was what this framework measures.

Use it to prepare for your next interview — or to evaluate your own growth as a QA professional.

THE FIVE POINTS

Point 1: CURIOSITY & PREPARATION

What I was looking for: Did this person come in with questions — not answers?

The strongest candidates walked into the room having done their homework. They'd read our engineering blog. They'd explored our product. They'd noticed something specific about our tech stack or recent releases — and they wanted to understand it better.

Red flags:

- No questions, or only questions about salary, benefits, and working hours
- Generic answers that could apply to any company

- Zero awareness of what the company actually builds

Green flags:

- "I noticed your team recently migrated from on-premise to cloud — how did that affect your testing pipeline?"
- "I read your post about CI/CD challenges — what's been the biggest bottleneck in test execution?"
- Demonstrates they researched *you*, not just the role

How to prepare: Find one thing the company has shipped or published recently. Form one genuine question around it. That's it.

Point 2: BUSINESS IMPACT THINKING

What I was looking for: Can this person connect testing to what the business actually cares about?

Most testers report bugs. The testers I hired reported *risk*. They could tell me what a bug would cost if it shipped — in money, in trust, in users. They didn't just describe what was broken; they described why it mattered.

Red flags:

- Describes bugs only in technical terms (steps to reproduce, expected vs. actual)
- Cannot articulate the real-world impact of a defect they found
- Treats all bugs as equally important

Green flags:

- "This issue affects roughly 12% of checkout users — about 3,400 transactions per month at risk"
- Frames testing in terms of business outcomes: revenue, retention, compliance, reputation
- Knows which bugs need a war-room response and which can wait for the next sprint

How to prepare: Before your interview, prepare one story where you connected a testing finding to a business outcome. Use numbers if you have them.

Point 3: COMMUNICATION

What I was looking for: Can this person make risk understandable to anyone in the building?

This is becoming more valuable in QA, not less. The best testers I worked with could walk into a room with developers, product managers, and executives — and adjust their language for each audience without losing the substance.

Red flags:

- Overly technical explanations with no ability to simplify
- Blames developers or throws colleagues under the bus
- Cannot explain a testing decision without jargon

Green flags:

- Adapts their explanation based on who's asking
- Describes how they've handled disagreement constructively
- Can explain a complex testing concept to a non-technical stakeholder

How to prepare: Practice explaining your most impressive testing achievement in two ways: one for a fellow engineer, one for a product manager who doesn't know what a test case is.

Point 4: JUDGMENT

What I was looking for: Do they know what *not* to test?

This is the skill that separates testers I want on my team from testers who are going through the motions. When time is tight and scope is large, what do you cut? What do you keep? And can you explain why?

Red flags:

- "I test everything" — this is not a virtue, it's an absence of judgment
- Cannot describe a situation where they had to make trade-offs
- No framework for prioritising testing effort

Green flags:

- "I sat down with the product owner and asked: what are the three things that absolutely cannot break?"
- Describes cutting test cases deliberately, with reasoning
- Uses risk-priority thinking naturally, not as a buzzword

How to prepare: Have a story ready about a time you made a difficult testing trade-off under pressure. What did you cut? What did you protect? Why?

Point 5: THINKING vs. CHECKING MINDSET

What I was looking for: Do they think about quality — or just testing?

This is the hardest thing to measure and the most revealing. The candidates I hired saw their job as protecting the user experience, not executing test cases. They questioned requirements. They noticed when acceptance criteria missed edge cases. They treated testing as a thinking discipline.

Red flags:

- Describes testing purely as following scripts or checklists
- No examples of questioning a requirement or challenging an assumption
- Sees their role as "finding bugs" rather than "managing quality risk"

Green flags:

- "The acceptance criteria covered the happy path, but I noticed it didn't address what happens when the payment gateway times out mid-transaction"
- Describes proactively improving a process, not just executing one
- Talks about testing in terms of risk, user impact, and systems thinking

How to prepare: Think of one moment where you went beyond what was asked — where you noticed something that wasn't in the test plan but mattered. That's your story.

THE SCORECARD

Use this in your next interview prep. Rate yourself 1-5 on each:

Point	Score (1-5)	Evidence / Story to Tell
1. Curiosity & Preparation		
2. Business Impact Thinking		
3. Communication		
4. Judgment		
5. Thinking vs. Checking		

Scoring guide:

- 5: I have a specific, compelling story that demonstrates this clearly
- 3: I understand this and can talk about it, but need a stronger example
- 1: I've never thought about this — it's a development area

A Final Note

None of these five points require another certification. None of them require five more years of experience. They require a shift in how you think about your work — and how you talk about it.

The candidates who got hired weren't always the most technical. They weren't always the most experienced. But they walked in demonstrating these five things.

About the Author

I've spent 25+ years in software testing and QA leadership, hiring and building QA teams across multiple companies. I've interviewed hundreds of testers, SDETs, and QA engineers — and I've seen the same patterns repeat across every company, every team, and every hiring decision.

This framework is what I wish every candidate understood before they walked through the door.



