

Is the “Iron Triangle” Dead in Software Development?

Philip Lew and Heather Wilcox

philip.lew@xbosoft.com and heatherwilc@yahoo.com

Abstract

There is an old saying in software and in life: You can have it Good, Fast, or Cheap. Pick two. Over our combined 50 years in the software industry, we have seen many companies try to violate this “Iron Triangle” when developing software products. For the most part, they have failed.

This paper is not based on scientific studies or data; rather, it reflects a recognizable trend in modern software quality: the quality bar is dropping. Pressures of time and cost are leading many organizations to quietly redefine what “acceptable” quality means. This shift shows up in the choices executives make—whether to outsource, accelerate delivery, or lean on AI-driven development—all while believing they can still achieve good, fast, and cheap.

As software quality managers and leaders, we need to decide whether to participate in this trend or push back. There is no single solution. Instead, organizations must be aware of the tradeoffs and deliberately choose from a range of possible practices such as outsourcing strategies, DevOps practices, shift-left testing, risk-based approaches, or managing technical debt. Essentially, technical leaders must decide how they will balance economic interests with the long-term health of their products and the trust of their customers.

Biographies

Philip Lew is CEO of XBOSoft, a software quality assurance and testing services company. With over 25 years of experience, he has advised organizations worldwide on improving software quality through better processes, metrics, and leadership. Philip is a frequent keynote speaker and workshop leader at PNSQC, SOFTEC, STAREAST, STARWEST, and the QA&Test Conference, as well as other international quality forums. His current focus is on integrating AI into testing practices and helping teams balance speed, cost, and quality without compromising customer trust.

Heather Wilcox is a senior software quality professional with extensive experience in testing, development, and management. Over her career she has led teams in both co-located and distributed environments, with a focus on practical strategies for sustaining quality under real-world constraints. Heather brings a strong interest in how outsourcing, DevOps, and emerging technologies like AI reshape the Iron Triangle. She regularly contributes to industry discussions and collaborates on thought-leadership papers to advance quality practices across the profession. In her spare time, Heather enjoys fiber arts, equestrian sports, and training donkeys.

Introduction

There is an old saying in software and in life: You can have it Good, Fast, or Cheap. Pick two. Over the authors' combined experience of 50 years in the software industry, they have seen many companies try to violate this "Iron Triangle" when developing software products. For the most part, these organizations have failed.

This paper is not based on scientific studies or data, but rather it is a reflection on a recognizable trend in modern software quality: the quality bar is dropping. Pressures of time and cost are leading many organizations to quietly redefine what "acceptable" quality means. This shift shows up in the choices executives make—whether to outsource, accelerate delivery, or lean on AI-driven development—all while believing they can still achieve good, fast, and cheap.

As software quality managers and leaders, we need to decide whether to participate in this trend or push back. There is no single solution. Instead, organizations must be aware of the tradeoffs and deliberately choose from a range of possible practices such as outsourcing strategies, DevOps practices, shift-left testing, risk-based approaches, or managing technical debt. Essentially, technical leaders must decide how they will balance economic interests with the long-term health of their products and the trust of their customers.

1 Defining and Understanding “Acceptable” Quality

What is “Acceptable Quality”? This is the \$64,000 question. In the past, acceptable quality has been generally defined as a complete solution that was free of Showstopper, Major, and Medium level defects. Minor defects were considered acceptable, but still a bit uncomfortable. However, in recent years, “Acceptable” has shifted to mean a Minimum Viable Product MVP that has been tested for Showstopper and major defects but with a myriad of medium and low priority defects that are not considered painful enough to resolve. This level of “acceptable” quality now often equates to a product that works but is riddled with technical debt and is rarely tested against non-functional requirements such as performance, security, usability, or accessibility.

In this new paradigm, even though the product ships and generates revenue, defects and product shortfalls lurk beneath the surface. For many executives, this is an acceptable trade-off, as long as the company is still making money. But for those of us in quality assurance, the idea of shipping code filled with defects waiting to be discovered by end users is troubling.

“Quality is free, but only to those who are willing to pay heavily for it.” (Crosby, P., 1979)

2 Why Isn't Quality Prioritized?

If this new lowered level of quality is so bad, why isn't testing prioritized? The simple answer is money.

Take the 737 Max 800. For decades, Boeing had a reputation for impeccable quality. The company traded on that reputation to build and sell the 737 Max 800, an incredibly popular mid-sized aircraft. But when problems emerged, FAA investigations revealed software issues that stemmed from prioritizing cost and speed over safety. (FAA, 2020)(Kitroeff, N., Gelles, D., & Nicas, J., 2019)

Another example of Money over quality is the 2024 CrowdStrike incident. A supposedly routine update to their Falcon sensor caused the infamous “Blue Screen of Death” on millions of PCs across the world, disrupting everything from individual PCs to airline, finance, and healthcare systems. After an internal investigation, CrowdStrike called out the need to improve their test processes. An article published shortly after the event states that, “Industry experts and analysts have since come out to say that the

practice of rushing through patches and pushing them directly to global environments has become mainstream, making it likely that another vendor could fall prey to this issue in the future.” (CIO Staff, 2024) In other words, they shorted testing to decrease time to market, thus saving money on test time. The cost? It is estimated at over 10 billion dollars. (CIO Staff, 2024)

Most software failures don't result in loss of life or billions of dollars like the examples cited above. However, poor quality still wastes users' time and money and can erode trust. In today's Software as a Service SaaS environment, where recurring subscriptions drive revenue, quality is tightly linked to customer loyalty. A substandard product might sell once, but renewal rates will suffer.

The question becomes: Will companies that sacrifice quality for cost eventually pay the price? Boeing already has—going from a \$19 billion profit in 2018 to a \$2 billion loss in 2024. (FAA, 2020).

The Boeing and CrowdStrike examples illustrate a broader trend called out in the CIO article cited above: **the quality bar is lowering across the industry**. Executives under pressure to deliver faster and cheaper often don't fully realize how much quality has been compromised until a critical problem occurs. Essentially, Quality isn't important enough to prioritize until there's a massive failure. At which point, it is often too late. The fiscal and reputation damage can be overwhelming and potentially unrecoverable.

So, what are the options for maintaining quality while delivering quickly at an acceptable cost?

3 Outsourcing as the Next Big Thing Again

One of the most common ways organizations have attempted to balance speed, cost, and quality is through outsourcing.

In the late 1990s and early 2000s, outsourcing QA teams was a significant trend. Many companies replaced their internal quality teams with third-party offshore testing firms, while development remained in-house.

This created a painful lesson: having developers and QA separated by multiple time zones was not an effective way to ensure quality. Defect turnaround time stretched from hours to days. Offshore QA was inexpensive, but it wasted costly development time answering questions as offshore teams lacked context and understanding of the business and software. Ultimately, many firms limited offshore QA to lower-level testing—compatibility or internationalization.

The failure of complete offshore testing led to a new trend: colocation. Developers and QA were placed not only in the same building but often on the same team in the same room. Pairing and mobbing became fashionable, and “Shift Left” became the industry mantra. The pendulum had swung from one extreme to the other.

At the same time, 2001 brought us the Agile Manifesto which reinforced the need for closer collaboration with new methods such as scrum and kanban enshrining close collaboration via face to face and colocation. This was all upended during the pandemic as people could not go to their office to work together which then drove the acceptance of working remote and from home.

Because of new remote working tools such as Zoom for meetings, and Atlassian for collaboration on issue handling and project planning, the pendulum has again swung back toward outsourcing. Organizations under cost and schedule pressure are looking for perceived efficiencies. The difference this time is that choices are more extreme, and success depends heavily on how much an organization is willing to lower its quality bar.

But outsourcing is only one way organizations have tried to respond to the pressures of the Iron Triangle. There are many other approaches worth examining—each with benefits and risks. The important point is that any single one of these does not solve the problem. Leaders must make conscious choices that they believe provide the most value and the best quality for their organization.

4 Paths Forward: Possible Responses to the Declining Quality Bar

There is no single solution to the Iron Triangle dilemma. Each organization operates within its constraints of leadership priorities, budgets, timelines, and customer expectations. What matters is not pretending that the triangle can be escaped, but acknowledging the tradeoffs and deliberately deciding how to manage them. Below are several approaches to consider. None is a magic bullet or total solution, but each offers ways to resist the erosion of quality.

4.1 Outsourcing Strategies

As discussed earlier, outsourcing remains a popular way... but handing over core product development often leads to long-term quality risks. Organizations pursuing outsourcing have a variety of models to choose from, each with different tradeoffs:

- 4.1.1 **Third-Party Offshore Companies:** Vendors in low-cost countries promise rapid progress at minimal expense, but often produce exactly what is specified—nothing more, nothing less. (Kim, G., Humble, J., & Forsgren, N., 2018)
- 4.1.2 **Local Third-Party Outsourcing Domestic:** More expensive but culturally aligned; often used for specialized tasks like accessibility or cross-platform testing.
- 4.1.3 **Hybrid Outsourcing Local + Offshore:** A mix of local Project Managers, a few Senior Developers and Quality Engineers with offshore doing the bulk of the work. Vendors Promise “best of both worlds,” but quality depends on the offshore execution.
- 4.1.4 **Offshore Company Branches:** Employees have more stake than contractors, but distance and cultural issues remain.

4.2 Leveraging AI

AI has become the newest “outsourcing” option, with organizations experimenting by replacing junior engineers or quality analysts with AI-assisted tools. While these systems can generate code, test cases, or even documentation at impressive speed, the quality of the results is uneven and highly dependent on context. AI can amplify productivity, but without experienced staff to guide prompts, review critically, and integrate responsibly, the risk of a brittle or incomplete solution grows. Rather than a replacement for human expertise, AI should be treated as an accelerator paired with knowledgeable oversight.

4.3 DevOps and “Shift-Left” Practices

DevOps and continuous delivery practices help reduce the delay between defect introduction and defect discovery, making quality issues visible earlier in the process. By embedding QA in design discussions, conducting code reviews, and implementing strong automation, organizations can prevent quality from being the first casualty of speed. Shift-left testing requires organizational cultural change and investment, but it creates tighter feedback loops that keep teams honest about the state of their product. In fast-moving environments, these practices act as safeguards against gradually lowering standards.

4.4 Quality Metrics That Matter

“You can’t manage what you don’t measure.” Unfortunately, many organizations still rely on vanity metrics such as the number of test cases executed or lines of code written—data points that say little about user experience or product reliability. More meaningful metrics, like defect escape rate, customer-reported issues, or churn tied to quality, connect engineering work directly to business outcomes. Choosing the right set of metrics helps executives understand the cost of lowering quality and gives teams a way to demonstrate the value of investing in it.

4.5 Building a Culture of Quality

Quality is not just a testing function—it is a cultural value that must be reinforced across the organization. When leadership rewards speed above all else, quality inevitably suffers, regardless of how skilled the QA team may be. Strong cultures set clear expectations that critical quality gates—security, accessibility, usability—are non-negotiable. Building such a culture takes time and consistency, but it ensures that protecting quality becomes everyone’s responsibility, not just the job of a testing group.

4.6 Risk-Based Testing and Prioritization

In a world of limited time and budget, it is impossible to test everything. Risk-based testing addresses this by focusing attention where defects would have the highest impact—on customer satisfaction, revenue, or safety. By classifying features or components by likelihood and severity of failure, teams can make smarter decisions about where to invest limited testing resources. This approach doesn’t eliminate risk but ensures that the riskiest areas are given the attention they deserve.

4.7 Managing Technical Debt

Technical debt is inevitable, but unmanaged debt compounds until it undermines both development speed and product stability. Treating it like financial debt—tracking it, calculating its “interest,” and deliberately paying it down—helps keep products sustainable. When ignored, debt slows every new initiative, creating the illusion of short-term speed while sacrificing long-term quality. Leaders who resist allocating time for refactoring or infrastructure improvements often discover that the bill eventually comes due, and the cost is far greater than planned.

None of these approaches alone “beats” the Iron Triangle. The key is awareness: quality will continue to erode if leaders do not protect it.

5 Conclusion: The Iron Triangle Isn’t Dead—It’s Distorted

The Iron Triangle remains as relevant as ever. Companies that claim to have “beaten” it usually aren’t defying the model—they’re redefining quality downward until the triangle appears conquerable. Under pressure to deliver faster and cheaper, organizations are quietly lowering the bar on what is considered “acceptable” quality.

We do not have a single solution to this problem. Our purpose is to raise awareness: the quality bar is slipping, often without executives or managers realizing it. The danger lies not in making tradeoffs, but in doing so unconsciously.

There are many possible responses. Some organizations may turn to outsourcing or AI; others may adopt DevOps, risk-based testing, stronger metrics, or deliberate management of technical debt. Each has strengths and weaknesses, and the right approach will depend on context. What matters most is that

leaders recognize what is happening and make intentional choices about how to protect quality in their organizations.

Ultimately, the Iron Triangle is not dead—it is distorted. Those that ignore the quality corner, risk normalizing “good enough” until quality becomes the weakest side of the triangle. When quality suffers, eventually so will your bottom line.

6 References

Barnes, M. 1969. The Iron Triangle of Project Management. Association for Project Management APM.

Crosby, P. 1979. Quality Is Free: The Art of Making Quality Certain. McGraw-Hill.

Drucker, P. 1995. Managing in a Time of Great Change. Harvard Business Review Press.

FAA 2020. Final Committee Report on the Design and Certification of the Boeing 737 MAX. Federal Aviation Administration.

Foote, B., & Yoder, J. 1999. Big Ball of Mud. University of Illinois, Computer Science.

Gartner 2021. Improving SaaS Renewal Rates Through Better Onboarding and Quality. Gartner Research Report.

Kim, G., Humble, J., & Forsgren, N. 2018. Accelerate: The Science of Lean Software and DevOps. IT Revolution Press.

McKinsey & Company 2023. The State of AI in Software Development. McKinsey Global Institute Report.

Ries, E. 2011. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses. Crown Business.

Zuckerberg, M. 2014. Move Fast and Break Things. Keynote at Facebook F8 Developer Conference.

Kitroeff, N., Gelles, D., & Nicas, J. 2019. Boeing Was 'Go, Go, Go' to Beat Airbus With the 737 Max. The New York Times.

CIO Staff: 2024. "CrowdStrike failure: What you need to know" posted August 1, 2024, <https://www.cio.com/article/3476789/crowdstrike-failure-what-you-need-to-know.html> (accessed Sep 7, 2025)