Gartner Research

Decentralize Quality Assurance Testing Teams to Drive Productivity

Jim Scheibmeir, Joachim Herschmann, Thomas Murphy

17 January 2024

Gartner

Decentralize Quality Assurance Testing Teams to Drive Productivity

Published 17 January 2024 - ID G00802862 - 8 min read

By Analyst(s): Jim Scheibmeir, Joachim Herschmann, Thomas Murphy

Initiatives: Software Engineering Practices; Build and Deliver New Digital Products/Experiences to Drive Business Results; Software Engineering Technologies

Modern software quality implies testing earlier, and more often, but centralized testers frequently remain a bottleneck for many software engineering teams. By distributing testers to product teams, software engineering leaders can free developers from unplanned work and release bottlenecks.

Overview

Key Findings

- Centralized testers are facing overwhelming demand for their time and skills. In fact, Gartner's 2022 Software Engineering Leaders Role Survey found that 35% of high-performing organizations expect increased demand for the quality assurance (QA) engineer role.
- While platform engineering is becoming a popular approach to improve software engineering, most organizations fail to include a "test" capability, which reduces the ability to scale software quality practices and tooling.
- Distributed testers allow product teams to have autonomy and own their quality, but they then miss out on shared learning.

Recommendations

- Embed quality engineers into product teams by starting with a pilot involving a single tester.
- Pave the way for integrated and streamlined development and delivery tools by leading a platform engineering approach that supports test tooling and testing activities.
- Support the acceleration of knowledge transfer and conformance to best practices by enabling your engineers with a community of practice, and budgeting for online learning platforms.

Introduction

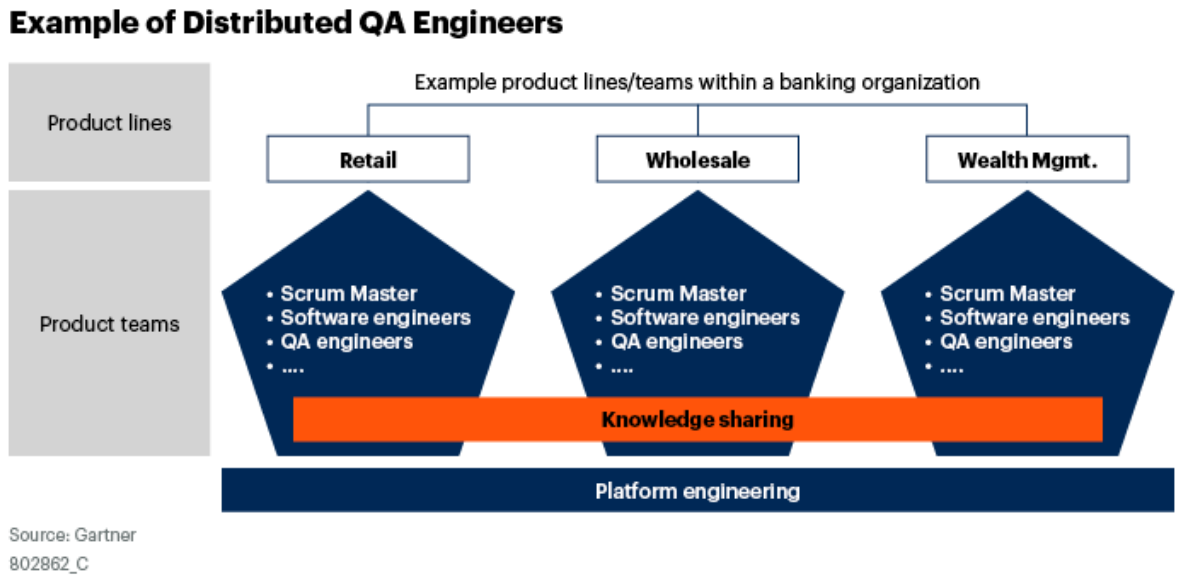
Despite the growing acknowledgment that testing needs to be a component of software engineering, rather than a bolt-on follow-up, software engineering leaders are confronted by the organizational momentum of centralizing testers into a silo of their own. Engaging centralized testers typically requires procedures for prioritization that fall out of in-sprint testing and automation deadlines. Centralized testers are frequently allocated across different products, leading to context switching time loss and cognitive overload when over-allocated.

Conversely, distributing testers to product teams aligns ownership of the activities of testing and software quality directly with the team building the software; however, the structure makes knowledge sharing and conformance to best practices a concern.

How should software engineering leaders free the bottleneck of testing and democratize testing tools and best practices?

To answer this question, we focus on distributing testing activities to product teams, using platform engineering to free testers from supporting their own frameworks, and supporting knowledge sharing of software quality practices (as illustrated in Figure 1).

Figure 1: Example of Distributed QA Engineers



Analysis

Embed Quality Engineers Into Product Teams by Starting With a Pilot

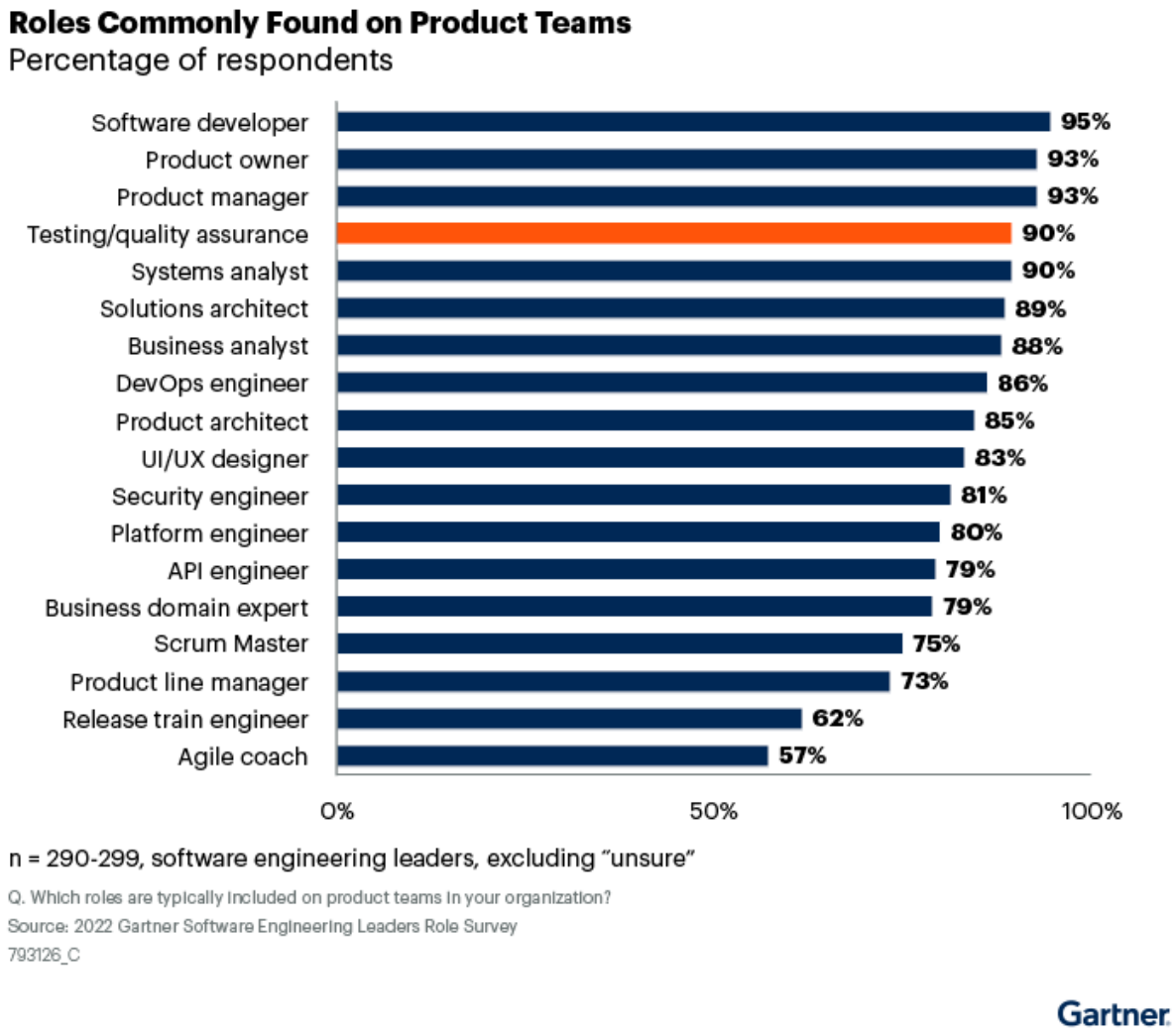
In Gartner's 2022 Software Engineering Leaders Role Survey, testing/QA roles were frequently found on product teams (90%), as Figure 2 shows. Gartner client inquiry related to distributing testers is a continuing inquiry trend, as are the problems associated with centralizing testers:

- Organizations won't budget to grow the central team in great numbers.
- The centralized testing team will create required yet wasteful processes and rules for their engagement — such as completing forms, attending prioritization meetings, and describing the product and project.
- When the centralized team of testers says "no" to helping others due to other priorities, testing has essentially become distributed to the product teams without adequate best practices.
- Product teams using agile and DevOps practices will view traditional centralized test centers of excellence (COEs) as bottlenecks and far removed from their business context.

If you have a centralized group of testers today, start this distributed testing initiative with a single pilot, integrating a sufficient number of testers to make the product team independent of the centralized silo. Invite these testers to leadership meetings so that they can inform your peers on the early struggles they experience, and evaluate the outcomes. (More on measuring outcomes and software quality metrics can be found in Quick Answer: What Metrics Should We Use to Assess and Improve Software Quality?)

The continued rollout of distributing testers should benefit from the experiences of your first movers. Rollout cadence should be influenced by the success of the pilot, not by arbitrary deadlines. Acknowledge early in the endeavor that this effort is working against silos and the frequent desire of leaders to organize by a type of activity.

Figure 2: Testing/QA Roles Are Commonly Found on Product Teams



The National Board of Medical Examiners is an example of an organization that has distributed its software testers and embedded them into its product teams.

Case in Point: Testers Embedded Within Product Teams (NBME)

The National Board of Medical Examiners (NBME) offers services for medical instruction students and educators. NBME has two product categories around business capabilities and key steps in the customer journey. Each product category has between one and four product lines. Each product line has a dedicated product team. NBME uses the following resource management approach:

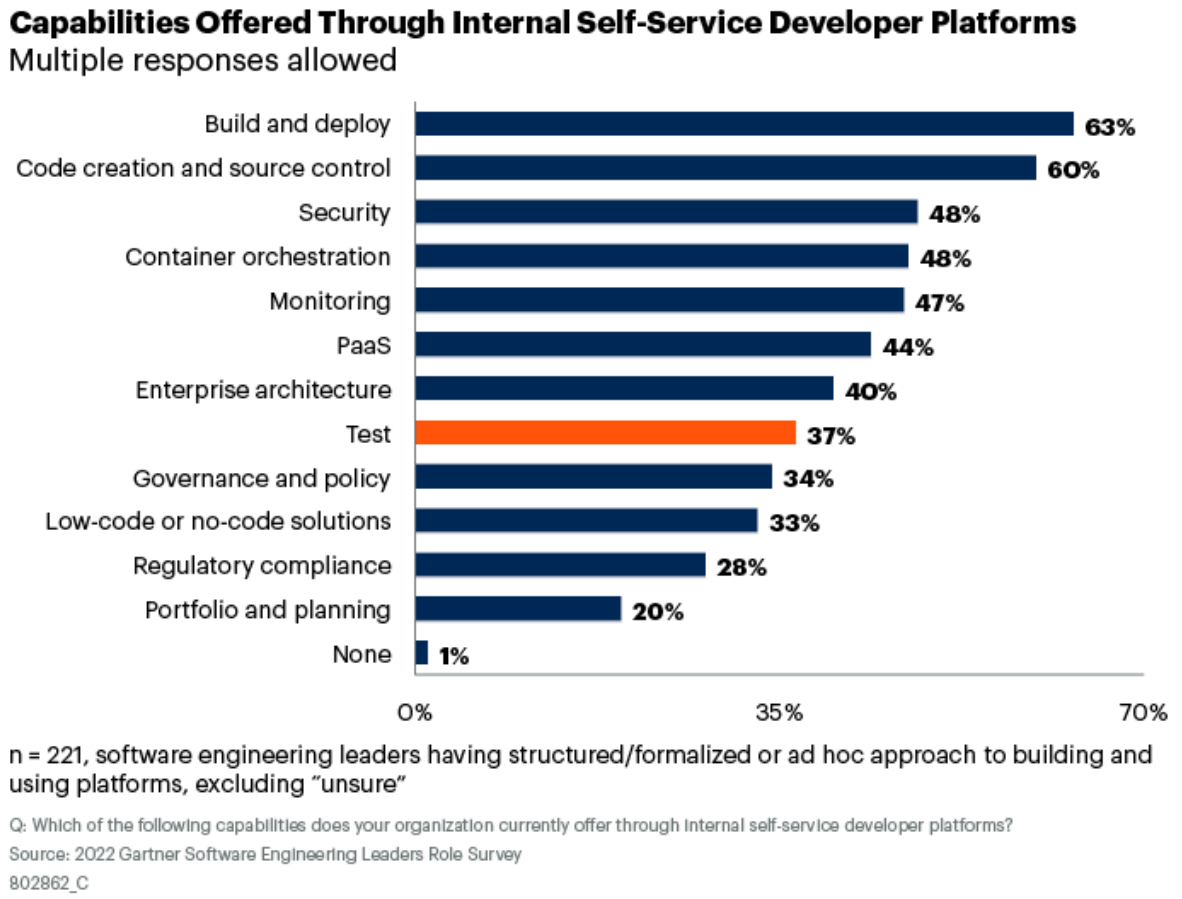
- Dedicated resources include software engineers and quality engineers.
- Shared resources such as Scrum Masters, solutions architects and UI/UX designers support a maximum of two product teams.
- Product team members are coached to have T-shaped skill sets and flex across roles to manage peaks in demand (T-shaped skill sets go broad across software engineering practices and deep in one type of activity).
- Product team members are internal full-time equivalents whenever possible.

Seeing the importance of having dedicated testers, at NBME, the quality engineer is one of only two fully dedicated roles on product teams, while other roles such as Scrum Master and solution architect are shared across teams. An organization chart illustrating NBME product teams and other collateral can be accessed in Product Management Team and Organizational Structures Library.

Pave the Way for Integrated and Streamlined Development and Delivery Tools

Platform engineering is the discipline of building and operating self-service developer platforms for software development and delivery (see Top Strategic Technology Trends for 2024: Platform Engineering). In Gartner's 2022 Software Engineering Leaders Role Survey, 88% of respondents viewed platform engineering as critical to support software engineering goals. The top three selected goals for pursuing platform engineering were improving software delivery speed (43%), improving software delivery quality (40%) and improving software reliability (27%). However, in another finding from the same survey, only 37% of respondents had a testing capability supported through internal self-service developer platforms in their organizations despite quality and reliability being two of the top three goals (see Figure 3).

Figure 3: Only One-Third of Respondents Test Through Internal Self-Service Developer Platforms



Testing tools have obvious places to fit into your platform engineering approach. Engineers, both developers and testers, benefit when they can select from scripts and other testing artifacts that are reusable, shared and readily available. While eliciting and prioritizing pain points experienced by your engineering teams (see more within How to Start and Scale Your Platform Engineering Team), the following components will enable easier consumption of quality best practices:

- Service virtualization and mocks for unit, API or UI tests.
- Test data representative of production (e.g., synthetic, production copy or masked data — for more examples, see 3 Steps to Improve Test Data Management for Software Engineering).
- Containers for test runners or configured integrations to hosted services for execution of automated tests and the mobile devices and browsers they run on.

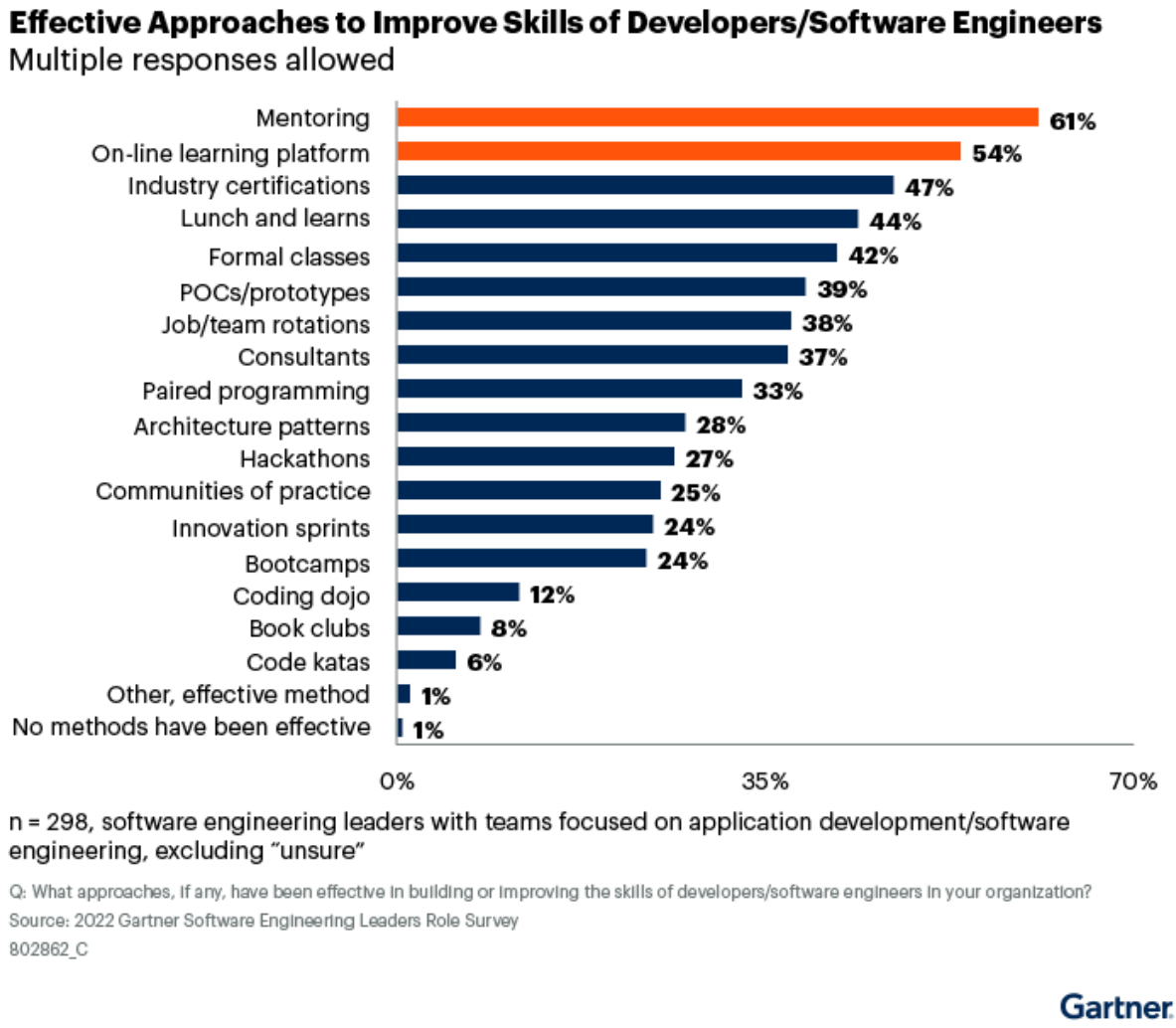
- Reuse of functional automated tests and integrations to cloud-hosted load testing generators.
- Application deployment capabilities that aid in repeatable and consistent deployments to preproduction environments, use of crowd-testing platforms, and production deployment capabilities including progressive releases and feature toggling.

Support the Acceleration of Knowledge Transfer and Conformance to Best Practices

A common problem associated with distributing centralized testers to product teams is the loss of a consistent knowledge transfer and approach to software quality. When testers are centralized, they can easily share best practices. However, there are multiple methods to improve knowledge creation and sharing.

In Gartner's 2022 Software Engineering Leader's Role Survey, over half of the respondents considered mentoring (61%) and online learning platforms (54%) to be effective in building or improving the skills of developers/software engineers. Communities of practice were found to be less effective (25%), but they still offer the direct benefit of bringing distributed engineers together in a community (see Figure 4).

Figure 4: Effective Approaches to Improve Skills of Developers/Software Engineers



Organizations should avoid stereotyping distribution as a sacrifice to economies of scale and consistency in practices. Instead, utilize shared, accessible resources to support distributed testers within product teams. This could include definition of done templates across your engineering organization (see Technical Brief: Definition of Done). By adding testing and quality activities to the definition of done, you will secure quality and testing activities within the development process (see Quick Answer: What Should a Definition of Done Contain?). The definition of done should also work to bring the new teammates —the testers — into active participation as the team works through their backlog and delivers quality features and fixes to customers.

Evidence

2022 Gartner Software Engineering Leaders Role Survey. This survey was conducted to understand how organizations attract, hire and retain software engineering talent; improve and modernize developer skills; improve developer productivity; establish platform engineering teams; create platform teams; and incorporate design into software engineering. The survey was conducted online from November through December 2022. In total, 300 respondents were interviewed from the United States. Qualifying organizations operated in multiple industries (excluding IT software and public sector) and reported enterprisewide revenue for fiscal year 2021 of at least $250 million or equivalent, with 60% over $1 billion in revenue. Qualified participants were highly involved in managing software engineering/application development teams and the activities they perform. Disclaimer: Results of this survey do not represent global findings or the market as a whole, but reflect the sentiments of the respondents and companies surveyed.

Disclaimer: The organization (or organizations) profiled in this research is (or are) provided for illustrative purposes only, and does (or do) not constitute an exhaustive list of examples in this field nor an endorsement by Gartner of the organization or its offerings.

Recommended by the Authors

Some documents may not be available as part of your current Gartner subscription.

Top Strategic Technology Trends for 2023: Digital Immune System

Cool Vendors in Software Engineering: Improving Digital Resilience

Infographic: Sketching a Continuous Quality Strategy Over Coffee

Quick Answer: What Metrics Should We Use to Assess and Improve Software Quality?

Quick Answer: 10 Key Nonfunctional Software Quality Characteristics

3 Steps to Improve Test Data Management for Software Engineering

---

Actionable, objective insight

Position your organization for success. Explore these additional complimentary resources and tools for software engineering leaders:

Research

How to Use Generative AI to Boost Developer Productivity

Use GenAI tools to improve developer satisfaction, collaboration and flow.

Download Now

Webinar

Improve Developer Productivity, Retention and Morale With Platform Engineering

Optimize the developer experience and accelerate customer value delivery.

Watch Now

Roadmap

Build and Deliver Digital Products That Drive Business Results

Mature your organization to deliver superior digital products.

Download Now

Webinar

Exploit Generative AI for Happier, More Productive Software Developers

Discover how AI coding assistants can be trained and used.

Watch Now

Already a client?
Get access to even more resources in your client portal. Log In

Connect With Us

Get actionable, objective insight to deliver on your mission-critical priorities. Our expert guidance and tools enable faster, smarter decisions and stronger performance. Contact us to become a client:

U.S.: 1 855 811 7593

International: +44 (0) 3330 607 044

Become a Client

Learn more about Gartner for Software Engineering Leaders
gartner.com/en/software-engineering

Stay connected to the latest insights

Gartner