How to Build an Effective Remote Testing Competency

Published 30 June 2020 - ID G00726391 - 18 min read

By Analysts Jim Scheibmeir, Thomas Murphy, Joachim Herschmann

Initiatives: Agile and DevOps and 1 more

There is an acute need to master remote software testing. Application leaders must build a remote-software quality competency by optimizing access and collaboration, investing in test environments and keeping their promises to the business.

Overview

Key Challenges

- Remote testing requires leadership to create a culture that embraces greater levels of team autonomy and communication. Without such leadership, application leaders risk poor team morale, prolonged time to market and unnecessary rework.

- Application leaders and their teams often undervalue test environments as second class technology systems, and this lack of care and funding inhibits the flow of value.

- Application leaders frequently break/flareuctuate their release date promises to business peers, which erodes their professional credibility and trust.

Recommendations

As an application leader responsible for agile and DevOps software quality, you should:

- Optimize access, collaboration and information sharing by reducing dependencies with a SaaS (software as a service)-based test management platform to improve communications and decision management.

- Invest in test environments, by adopting cloud, SaaS and test orchestration technologies, until thriving remote testing leads to increased business agility.

- Keep your promises to the business by shifting your team’s focus from manual testing efforts to a concentration on automation. This is achieved with training, time, tools and by measuring the outcomes.

Introduction

The need to master remote software quality has become vital. Application leaders and their teams frequently treat test environments as second class technology systems. According to Gartner’s
2019 Software Quality Tools and Practices survey, an average of only 17% of the total spend on software quality was toward test labs and infrastructure in FY18. This lack of care and funding continues and is reducing the application leaders and their teams’ ability to modernize test practices and support remote work.

Beyond lacking a foundational and trustworthy test environment, your quality experts are missing the autonomy to be efficient. Gartner’s Agile in the Enterprise Qualitative Survey for 2019 found that a top challenge to the success of agile development is the ability to shift from a culture of control to one based on trust (56%). Application leaders must commit to making this shift a reality.

A core question to answer is how to build and improve remote testing competency and this will require investment in skills, tools and the test lab environment.

Application leaders should build an autonomy-culture of Pride in Product (PIP) and free your testers from manual testing toil (illustrated through phases over time in Figure 1).

This research note is complimented by a case study. Windstream Holdings, Inc., a telecommunications company, has utilized remote testing prior to COVID-19 and the pandemic has not eroded the quality of their services.

“Until you’ve broken it, you haven’t tested it.”

— Art Nichols, VP Architecture and Technology, Windstream Holdings, Inc.
Analysis

Optimize Access, Collaboration and Information Sharing

Utilizing a SaaS based test management platform, while enabling team courage and promoting autonomy, will help keep the testing effort efficient. Exploit this opportunity to embed SaaS in order to enable efficient testing irrespective of location. It is foundational to include a shared and accessible resource where simple questions, such as who is testing, where in the application, when and to what extent may be answered. Such questions could be posed across time zones and communication mediums. A hierarchical approach to controlling or facilitating knowledge transfer and communication will be overwhelmed/underdelivered. Rather, focus on specific enabling collaboration instrumentation and practice, such as:

- A test management system must integrate into the persistent chat systems that teams are already subscribing to.

- Create specific channels and virtual teams within the chat tools where the focus is software quality.

- Welcome the team members to tune out when over communication is inflicting disruption (you trust them to find defects and protect the user experience (UX), so you also trust them to tune in or out when they need to).

Application leaders will set goals and objectives, use metrics to measure outcomes and coach for great team behavior. Tools can facilitate remote testing, communication, and collaboration, to a degree. Also, allow the product that is being built to do some of the culture lift. Building a great product, including one high in quality, also builds team pride. Devised by Gartner, Pride in Product (PIP) is a measurement of employee sentiment toward a product that is in-market. For more information on PIP, please see “How to Prove the Value of User Experience Design.”

Application leaders must infuse trust into the teams. High-trust cultures have a consistent record of delivering better financial and mission-focused results while reducing employee turnover. ¹ To build more trust, reduce hierarchical control, and free team members to communicate and collaborate at the speed of the team and remote work, not at the speed of a bureaucracy or a checklist.

Build courage and curiosity within your remote testers. Testers will and should prevent an application release from damaging your brand. They must feel safe and respected so that if they determine a risk or they lack confidence in the testing, they can voice their concern without retaliation. Google has found that psychological safety is a top characteristic of a successful team. ² In order to maintain a high level of psychological safety, you must maintain a tight culture where desired behaviors are coached, rewarded and constantly maintained. More guidance on
building psychological safety is found in “Create Psychological Safety in IT to Boost Team Performance.” The following table highlights example behaviors to amplify/mute.

**Table 1: Leadership Do’s and Don’ts of Remote Testing**

<table>
<thead>
<tr>
<th>Behaviors to Amplify</th>
<th>Anti-pattern Behaviors to Mute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Empower your teams to seek out risk in an application,</td>
<td>Do not micromanage how testing is done.</td>
</tr>
<tr>
<td>and attack it how they know best.</td>
<td></td>
</tr>
<tr>
<td>Celebrate finding bugs (it beats customers finding them)</td>
<td>Do not hide bugs in a spreadsheet where various stakeholders are void of the</td>
</tr>
<tr>
<td></td>
<td>history and the nuggets of learning from failure. Rather, predict future</td>
</tr>
<tr>
<td></td>
<td>areas that may require additional testing.</td>
</tr>
<tr>
<td>Utilize SaaS based test management platforms that</td>
<td>Do not be a central point of communication and decision authority.</td>
</tr>
<tr>
<td>democratize information and ease central communication</td>
<td></td>
</tr>
<tr>
<td>bottlenecks.</td>
<td></td>
</tr>
<tr>
<td>Grant your team the autonomy to determine which quality</td>
<td>Do not make a one-size fits all test strategy and force it upon each effort.</td>
</tr>
<tr>
<td>characteristics matter most for an application and its</td>
<td></td>
</tr>
<tr>
<td>users’ experiences.</td>
<td></td>
</tr>
<tr>
<td>Review the agile testing manifesto.</td>
<td>Do not write a 30 page test plan</td>
</tr>
<tr>
<td>Provide opportunities and resources for formal and peer-</td>
<td>Do not expect non-technical testers to be proficient with common open-source</td>
</tr>
<tr>
<td>to-peer training for your non-technical testers to learn</td>
<td>test automation tools as you may expect from a software developer.</td>
</tr>
<tr>
<td>technical skills, such as SQL for test data, scripting</td>
<td></td>
</tr>
<tr>
<td>for infrastructure configuration and common languages for</td>
<td></td>
</tr>
<tr>
<td>test automation.</td>
<td></td>
</tr>
<tr>
<td>Utilize persistent chat and online collaboration tools.</td>
<td>Do not mandate that remote testers be “always on” during working hours.</td>
</tr>
<tr>
<td>Take an employee-centric coaching approach for your</td>
<td>Do not attempt to limit your team’s mentorship or coaching to your guidance/</td>
</tr>
<tr>
<td>testers, and leverage your broader network including</td>
<td>experience only.</td>
</tr>
<tr>
<td>teams and leaders.</td>
<td></td>
</tr>
</tbody>
</table>

Source: Gartner 2020

**Invest in Test Environments**
Application leaders must invest in test environments (compute, data and tools) that enable trusted test results. As opposed to lamenting over whether a defect is a defect, or just the result of poorly maintained test configurations or thrashed test data, invest in repeatable test environment provisioning technology. This will enable teams to work in new ways, including the ability to work remotely.

In Gartner’s 2019 Software Quality Tools and Practices survey, among those that have automated software testing, 25% of respondents have only on-premises testing tools (illustrated in Figure 2 below). Remote testers rely on access to technology and solid testing environments for many reasons including:

- To build confidence in new software releases
- To inform decisions toward application release
- To collaborate with customers on acceptance
- To assure system resilience

Application leaders must invest in technologies that enable remote testing at speed. Investment includes time and influence with partners such as infrastructure and operations (I&O) Leaders.

**Figure 2: Remote Testing Requires SaaS/Cloud-Based Technology**

One type of technology that application leaders may invest in to enable remote testing includes device clouds and cross browser testing platforms. These offerings enable teams to test on many different form factors, device manufactures, OS and browser versions. Maintaining this level of
runtime variation is tedious within self-managed labs and overly expensive to distribute adequately to individual testers working remotely.

Cloud resources, container technologies and test data management benefit development teams and testers with reusable lab components, orchestrated provisioning and self-service. In the 2020 Gartner Container Adoption Strategy survey, 56% of respondents stated application development productivity is a top three technical benefit of container adoption (supporting modern application architectures was a top benefit for 52%).

Position your team so they utilize and scale orchestration and management layers to obfuscate the infrastructure. Along with infrastructure as code and immutable components, the orchestration layer allows non-technical testers to drive the provisioning of test environments. Business experts working remotely must be able to allocate test environment resources through intuitive portals, or rely upon continuous integration/continuous delivery (CI/CD) pipelines which notify them with demo site URLs once built and adequately provisioned.

A robust test environment strategy (thus a part of your test environment business case/investment) must support and encompass solutions against these common detriments:

- Disparate configurations of resources
- Manual deployment of applications to test
- Missing or inadequate test data
- Missing integration dependencies
- Lack of visibility, scheduling or notification

A test environment strategy should resonate with your CI/CD technologies and practices. Table 2 indicates example technologies that enable such a strategy.

<table>
<thead>
<tr>
<th>Brief Capability Description</th>
<th>Example Vendor/Technology</th>
<th>Example Open Source</th>
<th>Brief Product Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device Clouds</td>
<td>Amazon Web Services (AWS), BrowserStack, Experitest, pCloudy, Perfecto, Sauce Labs</td>
<td></td>
<td>These example device farms enable teams to test on many different form factors, device manufactures, OS and browser versions.</td>
</tr>
</tbody>
</table>

Table 2: Example Technologies That May Underpin a Robust Test Environment Capability
<table>
<thead>
<tr>
<th>Feature</th>
<th>Vendors</th>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask production data</td>
<td>Actifio, Broadcom, Delphix, IBM, Informatica</td>
<td>ARX, Fogger</td>
<td>Technology examples for masking sensitive production data for use in test environments.</td>
</tr>
<tr>
<td>Control Application Release Flow (Feature Flags)</td>
<td>Cloud Bees, GitLab, LaunchDarkly, Optimizely, Split.io</td>
<td>Unleash, FF4J</td>
<td>Example technology to control flow in test environments through feature flags and orchestrate provisioning and execution through CI/CD.</td>
</tr>
<tr>
<td>Generate Synthetic Data</td>
<td>GenRocket, Hazy, K2View</td>
<td></td>
<td>Example vendors and technologies to generate, version control and share synthetic test data.</td>
</tr>
<tr>
<td>Service (Dependence)Virtualization</td>
<td>Micro Focus, Parasoft, Tricentis</td>
<td>Hoverfly, WireMock</td>
<td>Open source technology examples for virtualizing production services as end-to-end integrations in dev or test environments.</td>
</tr>
<tr>
<td>Remote Debugging</td>
<td>Microsoft VSCode, Undo LiveRecorder</td>
<td></td>
<td>Example types of technology that can assist development and test teams debug applications remotely.</td>
</tr>
<tr>
<td>Test Environment Orchestration</td>
<td>CodeFresh, Plutora, Quali</td>
<td></td>
<td>Example provider of scheduling, allocation or orchestration capabilities for test environment resources.</td>
</tr>
<tr>
<td>Test Report Collection and Analysis</td>
<td>Allure Framework, ReportPortal</td>
<td></td>
<td>Allure Framework and Report Portal are open source examples of tools that can collect test results from many sources, integrate with common dev tools, and create graphical reports of the state of application quality.</td>
</tr>
</tbody>
</table>
Keep Your Promises to the Business

Software development features experience aging and bottlenecks as they travel through a value stream on their way to release. Utilizing automation and analyzing testing measurements for trends over time allow your teams to predict roadblocks and accelerate through them. These types of enablement, prediction through measurement and automation of tedious activities, help you keep release scope and timing promises to your business peers.

Some testing remains manual, such as exploratory. Much of the quality checks you do today should also be automated. In the 2019 Gartner Software Quality Tools and Practices survey, respondents indicated that 55% of their testing was automated. Automated tests should exist beyond the user interface (UI) layer. The same survey indicated API testing adoption at 61% and that static analysis was utilized by 59% of respondents. These levels of automation represent skill. You can build such talent, as well as source it during hiring. Utilizing remote testing extends the hiring pool by allowing your organization to source talent outside an immediate locality. Along with opening the skills market, physical extension of the hiring pool helps with building a great diverse team.

Test automation is an investment, and investments take time to produce worth. Focus on building out stable test environments before heavily investing in test automation. False positives from poor test environments will create waste and damage confidence and trust in the automated tests. Shifting to automated tests will offer the following positive outcomes:

- Enabling skills to be built in your team thus increasing the capacity, knowledge and capability
- Increasing team morale by automating tedious tasks and focusing on value adding activities such as exploratory testing
- Decreasing the duration of dev-test iterations when moving from serial manual tests to automated tests running in parallel
- Optimizing the cost of delivering a quality product

Test automation is frequently regarded as the automation of test-step execution. However, beyond the test steps are many more areas where tedious, error-prone and costly processes exist. Your test
automation strategy should not only be toward automating the test execution, but also in automating the peripheral activities associated with software quality. These peripheral activities include grabbing logs from dozens of remote runtime environments, and consolidating and analyzing them for error or outlying behavior. Measuring testing cycles and iterating through constraint identification and elevation is a key component to keeping your promises to business peers.

There is synergy and similarity between test automation technology and robotic process automation (RPA) technology, skills and use cases. RPA for testers condenses tedious activities around gathering log files, compiling reports and sending mass emails, to managing and maintaining a bot.

You must maintain an acceptable cadence and keep your promises to the business. So enable your team to reduce toil and build slack into the schedule through parallel and automated activities.

Table 3 below identifies example vendors that provide both RPA and test automation capabilities. For more information on the synergy and similarities of these technologies, please see “Assess Whether Test Automation Tools Could Help You With Robotic Process Automation.”

<table>
<thead>
<tr>
<th>Example Vendor</th>
<th>Product Websites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eggplant</td>
<td>Eggplant Robotic Process Automation (RPA)</td>
</tr>
<tr>
<td></td>
<td>Digital Automation Intelligence</td>
</tr>
<tr>
<td>LEAPWORK</td>
<td>LEAPWORK RPA</td>
</tr>
<tr>
<td></td>
<td>LEAPWORK Test Automation</td>
</tr>
<tr>
<td>Micro Focus</td>
<td>Micro Focus Robotic Process Automation</td>
</tr>
<tr>
<td></td>
<td>UFT One</td>
</tr>
</tbody>
</table>

Table 3: Example Vendors Supporting RPA and Test Automation Products
Case in Point: Windstream Holdings Inc. Remote Testing

Windstream Holdings, Inc., is a provider of network communications and technology solutions. The company provides data networking, core transport, security, unified communications and managed services to mid-market, enterprise and wholesale customers across the U.S. It also offers broadband, entertainment and security services for consumers, and small and medium-sized businesses primarily in rural areas in 18 states.

Since its formation as a company in 2006, Windstream has grown as an organization and as a business from many acquisitions; which has also brought the headaches and costs of managing many disparate test environments. In 2017, an architecture and technology group within Windstream started to deploy a test environment provisioning and scheduling system. This technology orchestrates the test environments on demand, reducing wait time. This technology also provides automation that refreshes the environments’ configuration and data, keeping them closer to production scenarios and builds more trust in the test results.

In a further departure from traditional testing, Windstream’s testers have, for many years, been geographically dispersed, working remotely. The nature of the remote testing requires more collaboration around test planning, communication of roadblocks, and teamwork to integrate dependencies and build confidence in Windstream services. This group utilizes persistent chat and other cloud collaboration tools, such as OfficeSuite HD, to share tools, test, and practice knowledge and maintain awareness.

Art Nichols, vice president within the architecture and technology group, believes that remote testers must have a voice and feel empowered. They aspire and prioritize self-service models where engineers, whether developers or testers, can move at the speed of automation. They use...
two metrics to measure for test success. Firstly, they trend test plan completion duration over time, to understand if bottlenecks in testing environments, processes or practices are slowing them down. Secondly, they track the number of defects found, by severity, during their testing. They celebrate finding bugs because it means the process is working. They also share defect findings with many other internal and external product teams, so that everyone can learn and improve. Automated test assets are shared, so each team can build more robust testing into their CI/CD pipelines. Art is a fan of robust testing.

The COVID-19 induced lockdown has proved the value of what they built. Today, they have fully remote testing capabilities, driven by CI/CD and exploratory testing techniques, and only one person physically maintains a multi-million dollar lab. Furthermore, the pandemic has not eroded the quality of their services.

“Until you’ve broken it, you haven’t tested it.”

— Art Nichols, VP Architecture and Technology, Windstream Holdings, Inc.

Acronym Key and Glossary Terms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Glossary Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIP</td>
<td>Devised by Gartner, Pride in Product is a measurement of employee sentiment toward a product that is in-market. PIP is utilized to gauge employee sentiment and predict future behavior such as productivity and attrition.</td>
</tr>
<tr>
<td>RPA</td>
<td>Robotic process automation</td>
</tr>
<tr>
<td>SaaS</td>
<td>Software as a service</td>
</tr>
</tbody>
</table>

Evidence

1. “Business Case for a High-Trust Culture,” Activated Insights
2. “Google Considers This to Be the Most Critical Trait of Successful Teams,” Business Insider.

Gartner’s 2019 Tech Innovation Study

This study was conducted online from September 2018 through February 2019, with 273 innovation leaders in the U.S. and U.K. It explored effective approaches to innovation, including goals, enablers and inhibitors of innovation, and the importance of emerging technologies for
meeting future innovation needs. Study participants either led an innovation program directly or had direct knowledge of, and involvement with, their organization's innovation initiatives and strategies. A mix of industries at end-user organizations with revenue of at least $250 million participated, with the majority of respondents representing organizations with worldwide annual revenue of over $1 billion.

The study was developed collaboratively by Gartner analysts and members of the Gartner's Primary Research team who follow innovation for enterprise architecture and CIOs.

Gartner's 2019 Software Quality Tools and Practices survey

This Gartner study was conducted to explore the transitions in tools, practices and structure for software quality at organizations. This primary research was conducted online from 29 April through 4 June 2019, and included 251 respondents in North America, Europe, India and Latin America. Qualifying organizations span various industries except IT services, real estate, business services and wholesale. Organizations were screened for having annual revenue for FY18 to be greater than/equal to $50 million. Organizations were required to be using DevOps, or lean or agile approaches for application development. The sample represents organizations in the U.S. (n = 66), Canada (n = 25), the U.K. (n = 52), Europe (France, Germany, Ireland, Netherlands and Switzerland) (n = 48), India (n = 30) and Brazil (n = 30). Respondents were required to have a role that is primarily IT focused or be a fairly even blend of business and IT. They were also required to have involvement in software quality at their organization. Quotas were applied for countries/regions, industries, annual revenue and in-house/outsourcing of DevOps, agile or lean development efforts. The study was developed collaboratively by Gartner Analysts and the Primary Research team. Disclaimer: Results do not represent “global” findings or the market as a whole but reflect sentiment of the respondents and companies surveyed.

The 2019 Gartner Agile in the Enterprise Quantitative survey

This survey was conducted via an online survey from 3 June through 25 June 2019, with 130 members of the Gartner Research Circle in the form of a Gartner-managed panel composed of IT and IT-business professionals. The survey was developed collaboratively by a team of Gartner analysts and was reviewed, tested and administered by Gartner's Research Data and Analytics team. The results of this study are representative of the respondent base and not necessarily the market as a whole.

Gartner’s Container Adoption and Strategy survey (2020)

This survey was conducted online from 3 March through 18 March 2020 with 91 Gartner Research Circle members in the form of a Gartner-managed panel of IT leaders. In order to participate, a respondent’s organization had to be involved in container technologies. Sixty percent are currently using container technologies, 27% are piloting/testing and 12% have plans to adopt. Respondents came from across the world — with a majority from Europe (49%) and North America (33%). This
survey was developed collaboratively by a team of Gartner analysts and Gartner's Primary Research team.

Approved methodology: Gartner conducts social listening analysis leveraging third party data tools to complement or supplement the other fact bases presented in this document. Due to its qualitative and organic nature, the results should not be used separately from the rest of this research. No conclusions should be drawn from this data alone as it may not be entirely market representative. Social media data in reference is from 7 March 2018 through 7 March 2020 in all geographies (except China) and recognized languages.

Recommended by the Authors

Create Psychological Safety in IT to Boost Team Performance
How to Prove the Value of User Experience Design
How to Choose Your Best-Fit Vendor for Test Management
Market Guide for Container Management
Best Practices for Suddenly Remote Agile Teams
Market Guide for Data Masking
Assess Whether Test Automation Tools Could Help You With Robotic Process Automation
The Future of DevOps Toolchains Will Involve Maximizing Flow in IT Value Streams
DevOps Success Requires Shift-Right Testing in Production
Become an Agile Superhero: 8 Attributes for Success

Recommended For You

Innovation Insight for Chaos Engineering
Innovation Insight for Promise Theory
Magic Quadrant for Application Release Orchestration
How to Safely Begin Chaos Engineering to Improve Reliability
Critical Capabilities for Application Release Orchestration

© 2020 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information.