



Quality Engineering Transformation

A ‘Shift-Left’ approach to mitigate quality, security and usability risks in software engineering

Leveraging ExtenSURE — Persistent’s Intelligent Product Engineering Framework





Application development continues to evolve, becoming more complex and distributed with the increasing adoption of DevSecOps, AI-ML and edge computing. Combined with ever-increasing cyber security threats, quality engineering is also transforming.

Today's quality engineering function must enable accelerated application, product, and service delivery, and be an enabler, rather than a barrier, to digital transformation.

To achieve this, testing becomes a continual function, not the final stage of the development cycle. Here we discuss ExtensURE, Persistent's Intelligent

Product Engineering Framework designed for the new world of software development.

Leveraging our **30 years of experience** in partnering with the world's largest software product companies and software-driven enterprises, ExtensURE embodies a **'Shift Left' approach** to quality engineering. It requires software engineering teams to maintain a testing mindset across the entire software development lifecycle and beyond, making sure that **reliability, security, and accessibility issues** are **identified** and **fixed** much earlier in the development life cycle.

Traditional QE Isn't Working

Traditional tried-and-tested approaches to testing and quality are falling short in today's evolving environments.

A siloed approach: Typical QE teams are siloed from development teams. This structure focuses on optimizing the subcomponents and ends up deviating from the real goal to optimize the user experience.

Slowing down overall engineering velocity: Traditional QE has primarily been manual, standing in the way of accelerated development and operations processes.

Expensive: Traditional QE takes up considerable engineering resources and consumes 30% – 40% of the overall budget.

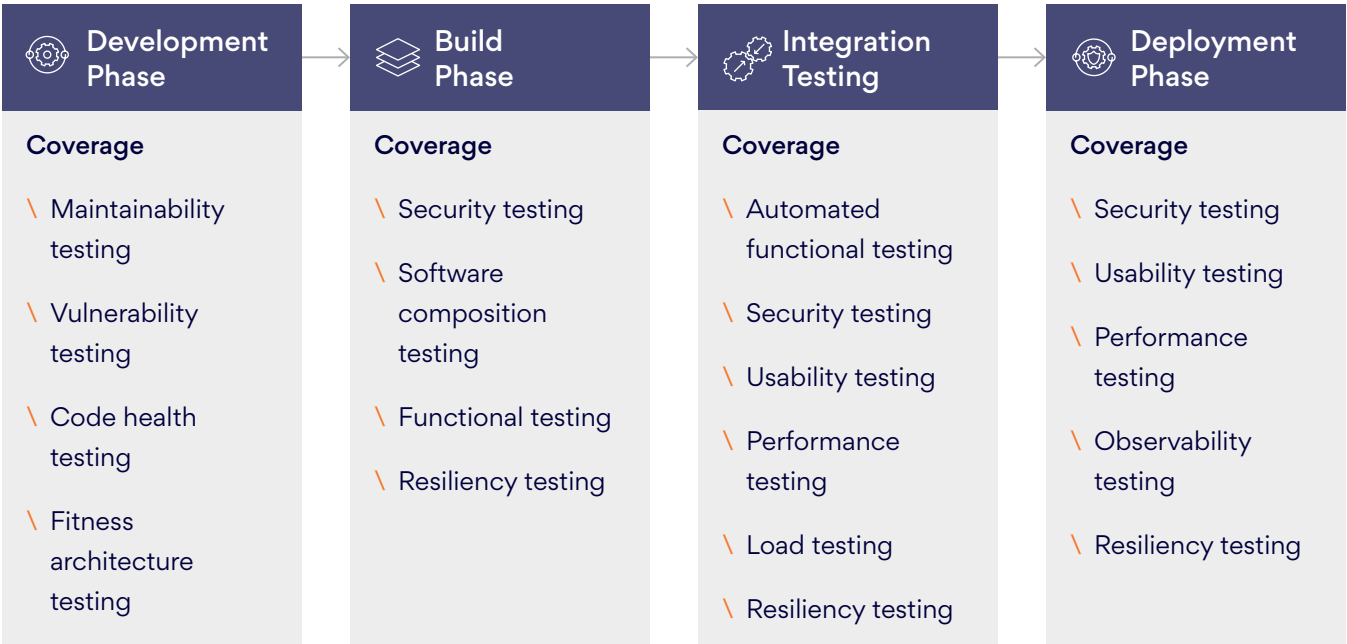
An afterthought: For decades, the testing approach was traditionally deferred to the end of a product cycle, by which time it is too late and can cause release delays and budget overruns.

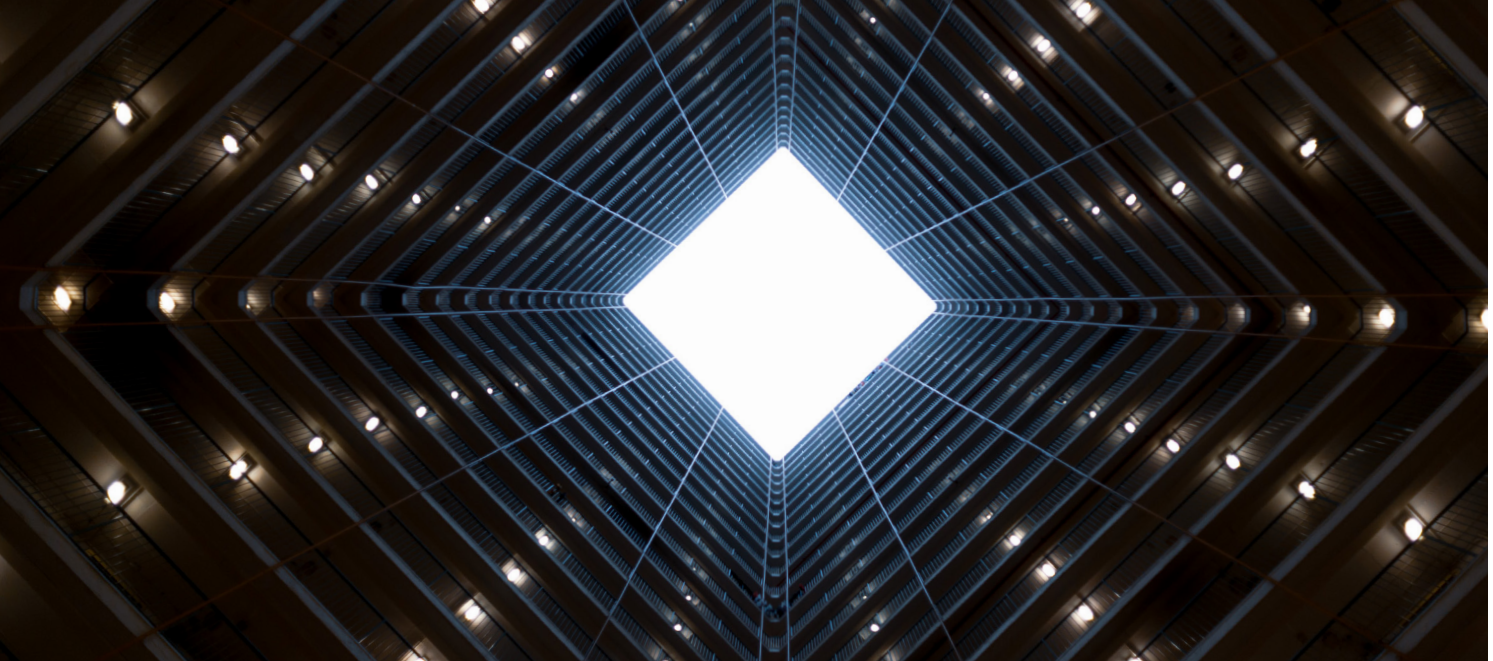
Testing Needs to 'Shift Left'

Software engineering teams must maintain a testing mindset and rigor as early as the design phase, through deployment and even beyond to production — monitoring production data and logs and creating a feedback loop to engineering teams. An automated 'Shift Left' approach to quality engineering focuses on

reliability, security, and accessibility at all stages of the design, development, and deployment process.

It ensures accelerated software releases and predictability in product releases by eliminating nasty surprises and helping to prevent cost overruns.





QE Transformation Leveraging ExtenSURE — Persistent’s Intelligent Product Engineering Framework

Transforming QE with a continuous ‘shift left’ approach primarily requires an automation-first approach by leveraging our AI-ML expertise coupled with best-in-class tools for a strong testing

mindset across every stage in the software product engineering lifecycle. Our approach introduces comprehensive testing as early as the design phase and continues post-deployment.

Functional Testing	Security Testing	Observability Testing	Resiliency Testing	UI Testing
<ul style="list-style-type: none">\ Codeless test authoring\ Visual inspection\ Automate anything (desktop, web, mobile, PoS etc.)	<ul style="list-style-type: none">\ Static application security testing\ Dynamic application security testing\ Software composition analysis\ Production application monitoring	<ul style="list-style-type: none">\ Application map\ CPU utilization\ I/O speed\ Response times\ Request rates\ Failed requests\ Live metrics	<ul style="list-style-type: none">\ Chaos engineering\ Load time\ Response time\ Wait time\ Error rate\ Throughput\ Latency\ Database testing	<ul style="list-style-type: none">\ A11y accessibility\ User flows\ User adoption\ Page views\ Customer satisfaction

QE rigor across every stage of the SDLC

Automated Functional Testing Using AI-ML

The biggest challenge and delay of function testing is the sheer volume of test cases that need to be written manually, maintained, and executed several times every day, week, month, while making sense of the test results for developer groups to act on them quickly.

ExtenSURE addresses this gap by leveraging AI-ML.

Codeless test authoring: Moving away from writing spreadsheet-based test cases, our approach leverages codeless test writing for functional test cases, which also accelerates creating automated test scripts to get test results faster. It records behavioral test cases with better coverage and accuracy as ML-based modeling captures every piece of data from DOM and CSS.

Automated regression testing: The framework auto highlights failure test cases and suggests One Click Cure (OCC) to recast test cases rapidly. This feature

significantly accelerates the regression testing process, further optimizing speed by combining with adjacent test areas, such as parallel browser testing, performance testing, etc.

Zero maintenance and improved defect tracking:

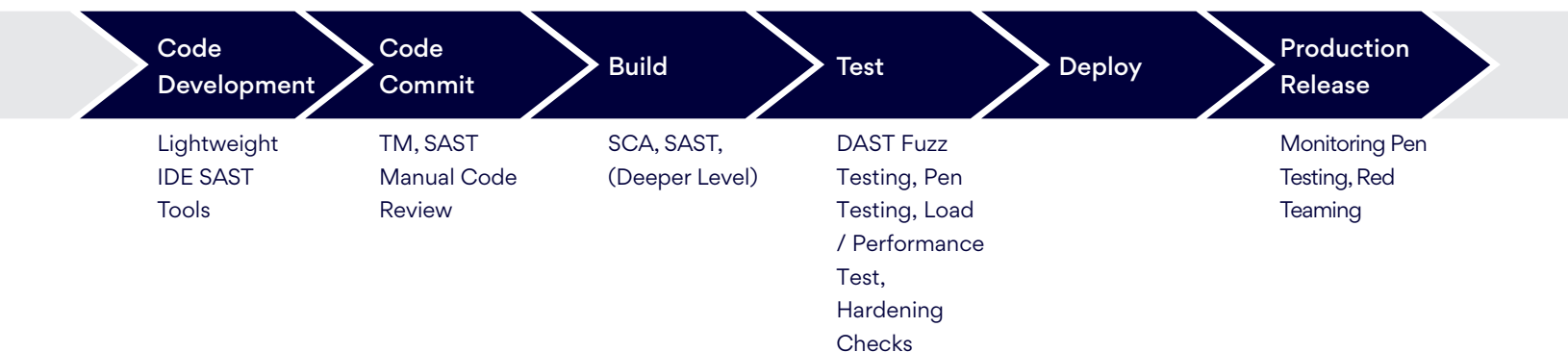
Any change in requirements requires manual updating of test cases. The framework speeds this process by recognizing application changes with details like “what was expected” and “what is found” and highlights them using advanced visualization techniques.

Leverage Developer Testers: Being a tool-led functional testing approach, the framework also reduces the dependency on a dedicated quality assurance team, allowing software engineering teams to leverage developer testers to improve speed and scale as the application grows.

Security Testing Using the Secure SDLC Approach

The ExtenSURE QE transformation framework forces development and security teams to incorporate security best practices and tools throughout the software development lifecycle. This avoids the

need to remediate security vulnerabilities in post-production, which escalates costs compared to addressing them in the initial stages of the development lifecycle.



1\ Static Application Security Testing (SAST):

The framework enforces SAST in early stages, i.e., code development, code commit, and build stages of the development phase, to provide early warnings correlated to code health and hotspots for the source code.

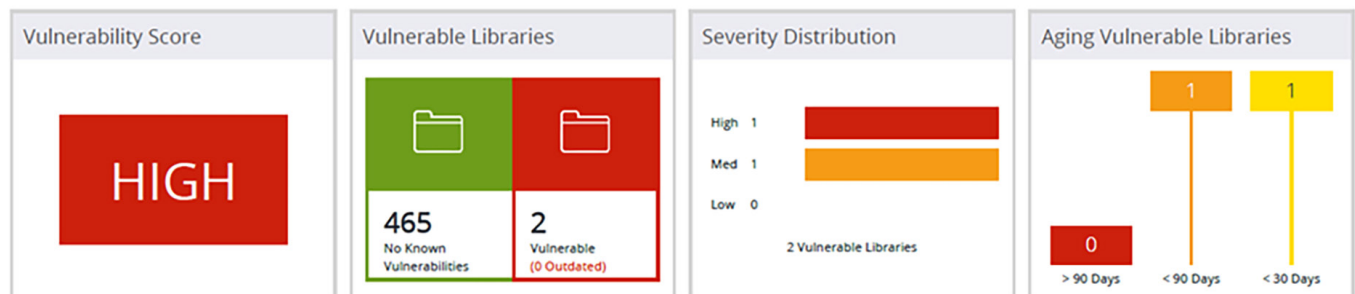
2\ Dynamic Application Security Testing (DAST):

The framework enforces robust DAST during the integration testing phase by leveraging OWASP ZAP (Zed Attack Proxy) to highlight low, medium, and high risks with suggestions to fix them.

3\ Software Composition Analysis (SCA):

The extensive use of open-source libraries and Docker images brings added risks of open library vulnerabilities, license compliance risks (MIT, Apache, GPL), and security vulnerabilities. The framework leverages WhiteSource and CAST highlights to perform a detailed software composition analysis, fixing issues early in the DevOps pipeline.

Security



4\ Production Application Monitoring (APM):

The framework prioritizes monitoring of the production application by leveraging Azure Application Insights to monitor production applications 24*7 for security issues and API failures to ensure robust application performance after deployment.

Home > extensurebotdev >

Insecure form data transmission detected (preview) ...

extensurebotdev

😊 Send a smile 😞 Send a frown ↗ Help ↗ Ask the community ↗ Suggest an idea

Detection Properties

Rule name	ⓘ	Potential security issue detected (preview)
When	ⓘ	5/6 5:30 AM - 5/7 5:29 AM
Number of affected forms	ⓘ	1
Most affected form	ⓘ	POST /api/messages

Detection Analysis

Number of affected users	ⓘ	2 (100.00% of all users)
--------------------------	---	--------------------------

Distribution of insecure data transmissions

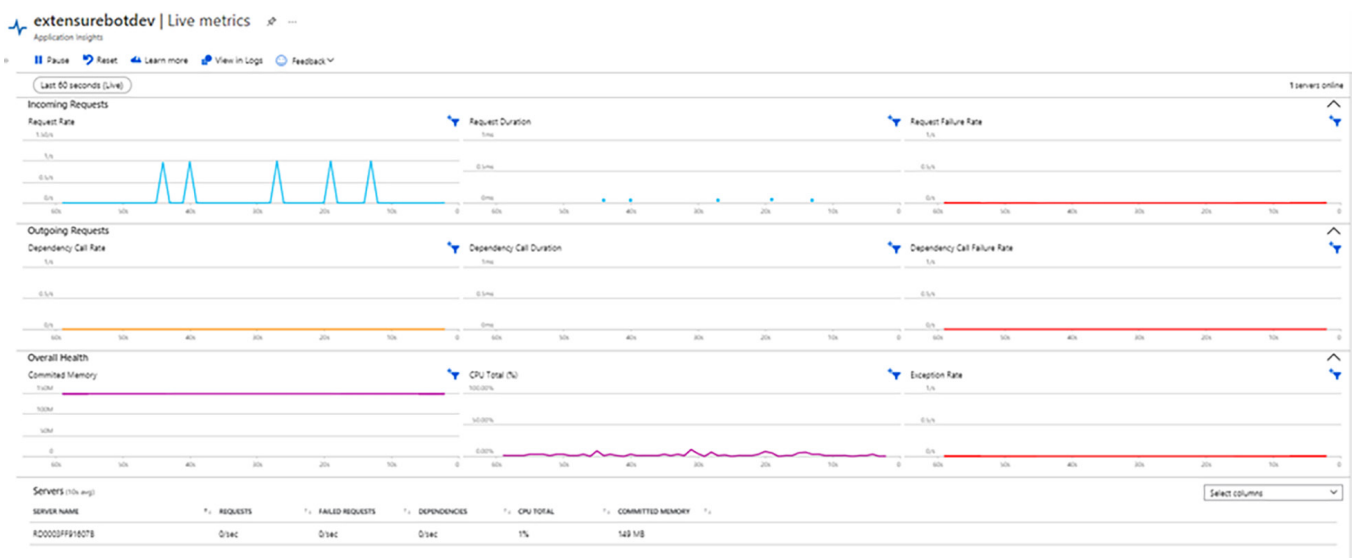
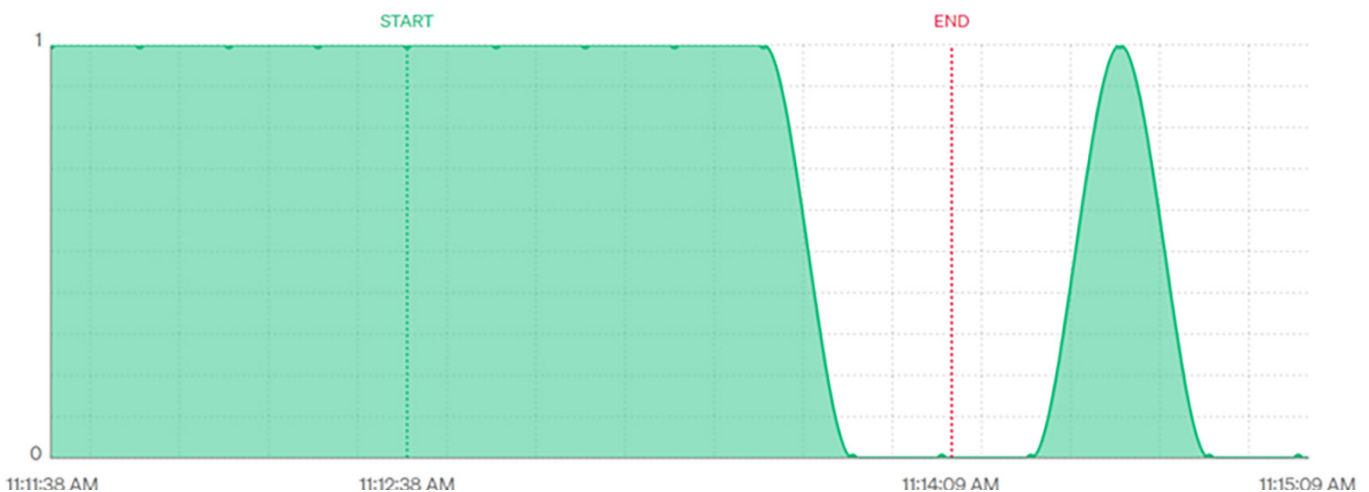


Resiliency and Observability using ‘Chaos Engineering’

The framework uses principles of chaos engineering to stress-test applications and infrastructure for unknown unknowns. It simulates extreme failures and edge cases, preparing for issues under scenarios such as:

- Application behavior when CPU utilization reaches 100% — Does it auto-scale or crash?
- Application behavior when 30 – 40% packets drop for some microservices — Slowness observed and possible impact on NPS.
- Application behavior when a critical infrastructure pod is suddenly shut down.

Number of Healthy Hosts



Example — Real-time application behavior when one of the VMs is shutdown



Usability Testing with an Accessibility-First Approach

In the United States, the Americans with Disabilities Act (ADA) Standards for Accessible Design require all electronic and information technology (such as websites, apps, and videos) to be accessible to people with disabilities, making companies put accessibility and usability at the center of their software

development processes. To address this requirement, the framework couples two critical components for usability — accessibility (A11y compliance) and usability — right from the design phase of the software engineering lifecycle to mitigate accessibility and usability issues early in the development process.

✓ x11	scrollable-region-focusable: Elements that have scrollable content must be accessible by keyboard
Resources for this rule More information about scrollable-region-focusable WCAG 2.1.1	
Path	.pushy
Snippet	<div class="pushy pushy--left" data-focus="#first-link">
How to fix	Fix ONE of the following: <ul style="list-style-type: none">• Element should have focusable content• Element should be focusable

Benefits of ExtenSURE's QE Transformation Approach

10x
Faster

Resulting from
automation



85%
Time Saved

From 'One-Click-Cure'
of broken test cases



5x
improvement

Improved testing
coverage from 10%
to 50%



80%
Cost Reduction

Reduced resource and
infrastructure cost

Client Success: Indian bank accelerates the launch of a modernized credit card platform through QE transformation

An Indian bank with a sizeable credit card portfolio faced enormous challenges due to its legacy applications.

To give the customer a seamless and modern user experience, the client partnered with Persistent to build a modern platform for credit card applications that allowed customers, sales agents, the operations team, and the bank underwriters to collaborate across the credit card application workflow. The platform also needed to integrate with existing bank systems and other 3rd parties' systems. Using Persistent's ExtenSURE QE transformation framework, the bank accelerated the testing process by:

- Using **codeless test authoring** and **Record and Play**, as part of the automated the complete credit card application workflow.
- Leveraging **Heal-all and One Click Cure** features to **reduce** manual efforts by **61.23%** to write and maintain test cases. Reduced time to write test cases from **26 mins to 9 mins** per case.
- Leveraging **computer vision validation** to build intuitive visual comparisons and identify failures.

The automated QE transformation framework accelerated the overall engineering velocity of the initiative allowing the bank to launch this platform ahead of its competitors.

Why Persistent



Product Engineering DNA

A 30-year legacy of leadership in software product engineering and digital transformation.



Partners in Innovation

Excellent track record with clients and innovation partner of choice for the world's leading software product companies.



Robust Execution Framework

Leveraging our signature methodology that combines the power of Design Thinking, Hackathons, Agile, DevOps, CI/CD and industry solution accelerators to build next-generation digital products with breakthrough operational efficiency.



Expanding Partner Ecosystem

Connectors, integrations and accelerators built on or built with leading platform partners such as AWS, Google Cloud, IBM, Mambu, Outsystems, Microsoft, RedHat and Salesforce.

About Persistent

With over 15,000 employees around the world, Persistent Systems (BSE & NSE: PERSISTENT) is a global services and solutions company delivering Digital Engineering and Enterprise Modernization. As a leader in software product engineering for 30+ years, we offer the expertise, operating models, and technology required for a unique competitive advantage.

India

Persistent Systems Limited
Bhageerath, 402
Senapati Bapat Road
Pune 411016
Tel: +91 (20) 6703 0000
Fax: +91 (20) 6703 0008

USA

Persistent Systems, Inc.
2055 Laurelwood Road, Suite 210
Santa Clara, CA 95054
Tel: +1 (408) 216 7010
Fax: +1 (408) 451 9177
Email: info@persistent.com



Persistent

www.persistent.com