OpenText Evolve Software Testing For Stellar Application User Guide

Contents

OpenText Evolve Software Testing For Stellar Application

Specifications:

- Product Name: Software Testing Evolution

- Features: Performance testing, Functional testing, Automation, Intelligence
- Benefits: Improved efficiency, accuracy, speed, application resiliency, reliability

Product Information:
The Software Testing Evolution product focuses on improving application resiliency, reliability, and speed through performance and functional testing. It emphasizes the importance of software testing in ensuring that applications meet expected standards of quality and functionality.

Product Usage Instructions

Automation and Intelligence:
The product introduces automation and intelligence to streamline testing processes, improve efficiency, and enhance accuracy.

Best Practices:
Follow best practices such as collaboration, integration, and continuous improvement to achieve high-performing applications.

Introduction: Harness the speed of change
For organizations to move and innovate quickly to meet market and customer demands, software development needs to keep pace with desired agility and speed. Unfortunately, software development practices may be hurting, rather than helping, operations. Software testing, a critical part of software development, is often fraught with inefficiencies. It is frequently plagued by legacy tools, manual processes, staffing shortages, testing conducted too late in the development lifecycle, and an overall lack of harmony. When testing is not optimized for efficiency and is conducted in isolation, there is risk of time, money, and resources being wasted, software deployments delayed, and customer trust eroded if user experiences don't deliver as promised. There is good news however: we're in the midst of a software testing evolution. Tools are generating much-needed integration, collaboration, automation, and intelligence—resulting in improved efficiency, accuracy, and speed. Let's explore what's possible with the latest in technology for performance and functional testing, best practices to deliver high-performing applications, and what's needed to make software development more accessible, scalable, and cost-effective.

The importance of software testing

Software testing is the process of evaluating, verifying and validating that an application does what it is supposed to do. It's about gathering as much insight and information as possible and running various test scenarios to pinpoint issues that could impact functionality, performance, security, and the overall user experience. The importance of software testing cannot be underestimated. For example, in June 2024, a faulty software update from cybersecurity vendor, CrowdStrike, led to widespread global outages, impacting airlines, banks, and emergency services and raising questions about the company's software testing. When testing is done right, companies can save significant development and support costs. They are able to quickly identify and address issues tied to functionality, architecture, security, scalability, and design before a product goes to market.

Five ways software testing elevates the software development lifecycle

1. Supports on-time software releases
2. Ensures quality and performance
3. Reduces risk with early issue identification
4. Verifies usability
5. Drives continuous improvements

Evolve software testing for stellar application resiliency, reliability, and speed

Six testing best practices

There are many different types of software testing—each with their own objectives and strategies—which play an integral role in ensuring the final product meets expected standards of quality and functionality.

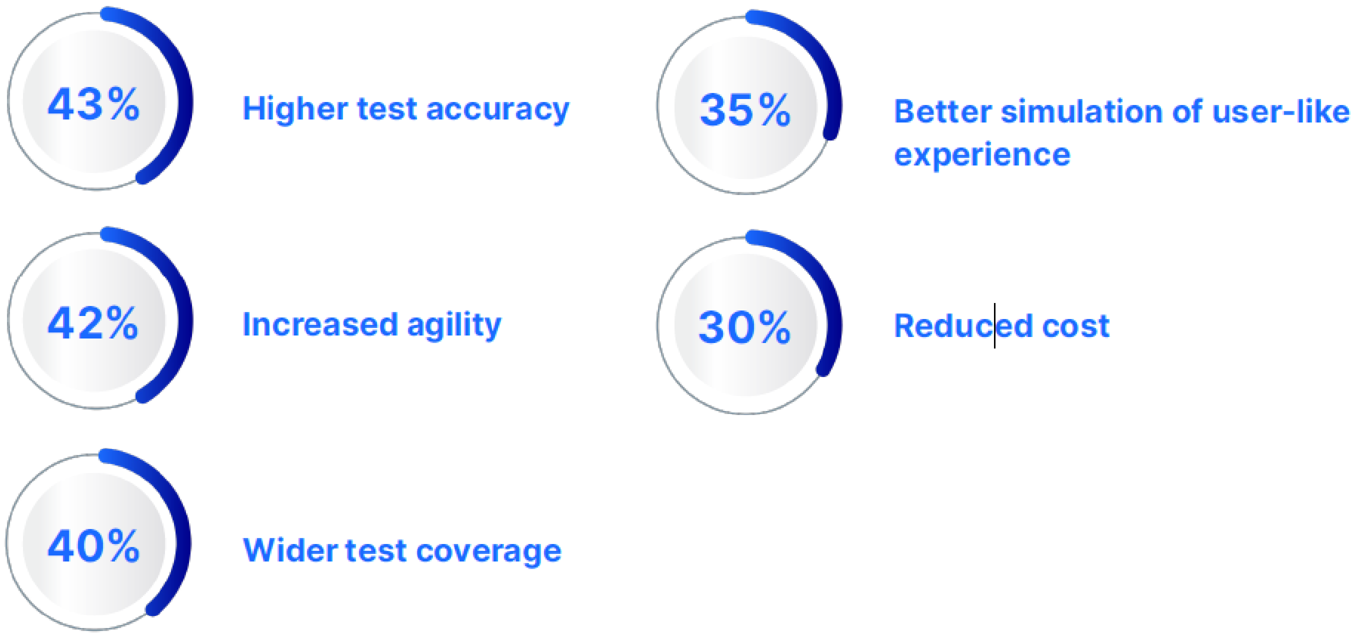
Here are best practices that should be applied to testing processes to support the overall software development process:

1. Make testing top of mind: Shift testing from an afterthought to a priority.
2. Be proactive: Implement a strategy and discipline to conduct tests early and often.
3. Share insights and learnings: Analyze metrics to promote best practices and areas for improvement across design, development, and testing teams.
4. Increase collaboration: Enable seamless team access to testing operations, schedules, and results.
5. Harmonize testing tools: Ensure testing tools work together and are tightly integrated.
6. Reduce manual steps: Automate where possible.

The evolved approach: Introducing automation and intelligence
Bringing automation and AI to software testing is a proven way to increase effectiveness, efficiency, and coverage.

- 60% of companies said improving product quality was among their organization's reasons to automate software testing1
- 58% said their organization was influenced by a desire to increase deployment speed2

After automating software testing, organizations report:3



1. Gartner, Automated Software Testing Adoption and Trends, 2023
   GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.
2. Ibid.
3. Ibid.

Performance testing: Why it's important

Performance testing determines the stability, speed, scalability, and responsiveness of an application under different workloads. Requiring deep technical skills and involvement across multiple teams, performance testing is commonly thought of as complex and daunting. Far reaching, it typically includes load testing, stress testing, scalability testing, endurance testing, and more. It is essential to validate the production performance of applications before release into a live environment to identify potential software issues—all of which can negatively impact the user experience:

- Long or poor application response times
- Slow load times
- Limited scalability for increasing user loads
- Performance bottlenecks
- Underused and/or overused resources (CPU, memory, bandwidth)

Performance testing generates massive amounts of data, traditionally requiring time-consuming, manual involvement. By bringing automation to this complex process, issues can be identified faster, adding consistency and repeatability to testing processes—delivering continuous improvements.

Performance testing: Common gaps and challenges

The performance testing phase of the software development cycle is vital, but often easier said than done.

Common challenges that hinder testing effectiveness and reach include:

Limited collaboration
Siloed activities lead to duplication of efforts by developers, performance engineers, and analysts.

Application complexity
High volumes of technologies and services, combined with gaps in coverage, can force teams to selectively choose what and where to test.

Data overload
Staff can struggle to conduct root cause analysis, making it more challenging to pinpoint issues and accurately interpret performance.

Unrealistic network conditions
Lack of ability to simulate real-world environments and anticipate real-world problems, such as seasonal demand.

Steep learning curve
Requirements for various test design and scripting tools impact fast adoption and ease of use.

Rising costs
The maintenance of test assets and infrastructure costs increase, placing pressure on human resource and tooling budgets.

Functional testing: Why it’s important

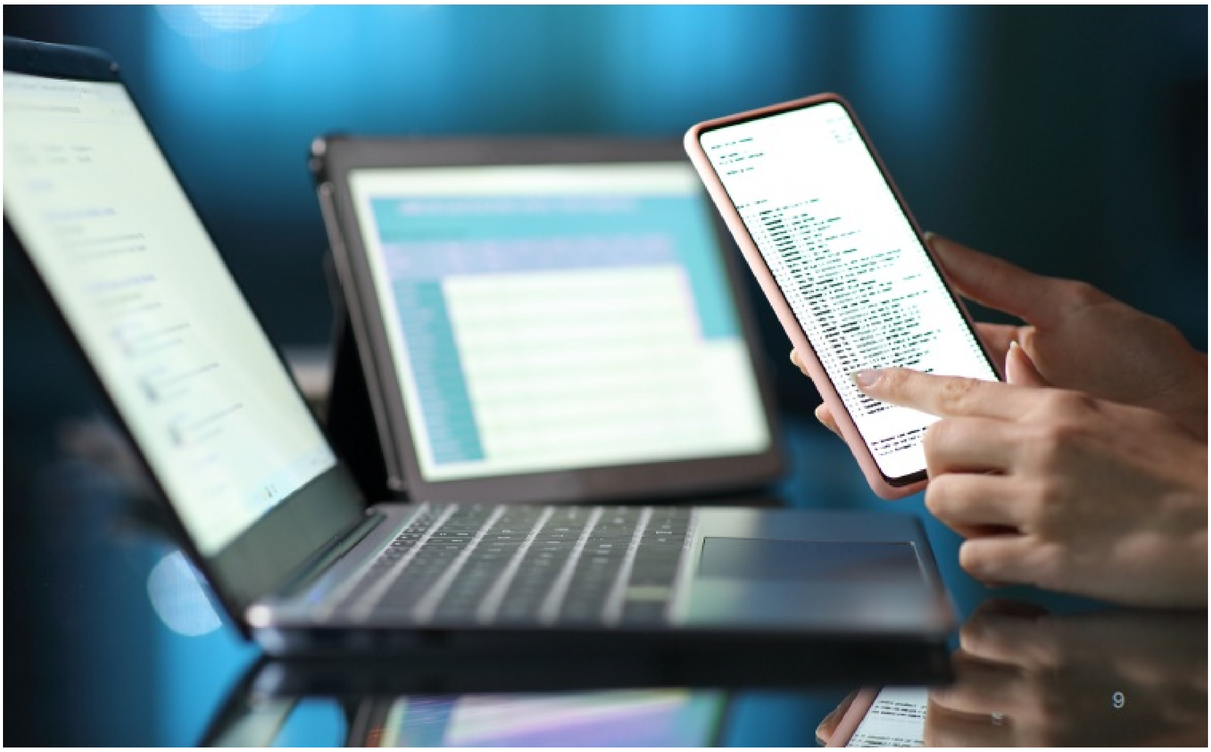
In the fast-paced environment of software development, functional testing is critical to ensure solutions perform as expected, per the application's functional requirements. In other words: verifying the features the application or software system is expected to have. For example, for a payment module, functional testing scenarios may include multiple currencies, processes for handling expired credit card numbers, and generating a notification on the completion of a successful transaction.

Functional testing is important to the software development lifecycle, delivering four key benefits:

1. Confirm end user outputs: Checks APIs, security, client/server communication, database, UI, and other key application functionalities.
2. Mobile testing: Ensures applications perform seamlessly across various devices and operating systems.
3. Identify and address performance gaps: Reproduces the user experience in a live environment to meet desired requirements.
4. Decrease risk: Improves product quality, eliminates bottlenecks, and boosts security.

Gain a complex picture of application security

Software testing helps unearth and resolve security vulnerabilities at various points throughout the software development lifecycle. Combining static analysis and dynamic analysis tools delivers improved visibility, boosting collaboration and remediation and minimizing risks to the software supply chain.



Functional testing:

Common gaps and challenges

Functional testing can be repetitive and time-consuming.

Introducing automation drives time and cost savings, improving test execution, visibility, and ROI by addressing six common challenges:

Wasted time

Limited machines and/or devices, automating the wrong things, and actions not aligned to business requirements.

Staffing shortages

Resource constraints make it difficult to balance and prioritize responsibilities among developers and testers.

Time-consuming test execution

Unreliable scheduling, too many test execution engines, and difficulty running tests in parallel.

Skills gaps

Current practices require technical know-how to leverage automation, minimizing business user involvement and input.

Tedious test maintenance

Duplicate test creation, tests resilient to frequent changes, and broken automation.

Infrastructure overhead

Multiple testing environments (browsers, mobile devices, etc.) and hardware support for test solutions (hardware, licensing, patching, upgrades).

OpenText: A partner for automated, AI-powered testing

As an automation and AI pioneer, we understand the importance of helping organizations embrace new ways of working, empowering teams to reimagine software development.

Accelerate software testing processes with a trusted partner who stands apart due to five key advantages:

1. **Deep experience and expertise**

   Take advantage of our deep understanding of software testing challenges and requirements. OpenText has a proven track record of delivering reliable testing tools trusted by leading enterprises around the world.
2. **Non-stop innovation**

   Get advanced testing solutions that integrate cutting-edge AI, machine learning, and cloud capabilities.
3. **Comprehensive testing toolset**

   Simplify and drive efficiency across the complete testing landscape with OpenText technology. Our tools support functional and performance testing, mobile testing, and test management.
4. **Proven, trusted support**

   Receive unparalleled support and be part of our vibrant user community. You and your team can quickly resolve issues and share best practices, enhancing your overall experience and productivity.
5. **Broad integration ecosystem**

   Use tools you are already familiar with. OpenText supports integrations across open source, third-party tools, and other OpenText solutions. You can also easily support multiple testing strategies across your software development lifecycle.

Get what you need for performance engineering

Expand on traditional performance testing approaches with OpenText and adopt a proactive, end-to-end testing and monitoring discipline: performance engineering. Leveraging automation and AI, we facilitate complex, enterprise-wide load, stress and performance scenarios, simulating real-world network and load conditions and supporting testing across any application type and protocol—in any software development environment. We make testing processes more agile, facilitate continuous improvement via constant feedback loops, and help organizations keep up with testing demands by leveraging built-in integrations across CI/CD, open-source tools, and third-party testing tools.

Elevate your team with a shared testing platform that tackles all your performance testing challenges:

Simple: Easy to use, with tests and scripts uploaded in minutes.

OpenText performance engineering solutions

- OpenText™ Enterprise Performance Engineering (LoadRunner™ Enterprise): A collaborative testing platform that reduces complexity, centralizes resources, and leverages shared assets and licenses.
- OpenText™ Professional Performance Engineering (LoadRunner™ Professional): An intuitive, versatile solution that saves organizations time, improves code coverage, and provides accurate results.
- OpenText™ Core Performance Engineering (LoadRunner™ Cloud): Conduct extensive performance testing without costly infrastructure.
- Smart: Predictive analytics, location-aware analytics, and transaction analysis provide real-time information, easily pinpointing the cause of problems and providing optimization recommendations.
- Scalable: Scale to more than five million virtual users for ultimate test coverage and use a cloud-based SaaS to scale dynamically and on demand.

Get what you need for functional testing
Transcend the boundaries of functional testing tools with an OpenText solution designed to meet the demands of modern software development. Our embedded AI capabilities accelerate functional testing design and execution, allowing teams to test earlier and faster for web, mobile, API, and enterprise applications.

As a result, organizations can:

- Save time, boost accuracy: AI-driven capabilities reduce script creation time and enable tests to be scaled across distributed architectures.

- Optimize coverage: Support any development methodology, including Agile and DevOps, for effective and streamlined testing processes.
- Minimize skills gaps: Involve business users (SMEs) in test automation processes, leveraging built-in model-based testing methodology.
- Gain insights: Leverage comprehensive reporting and analytics to quickly identify and remediate issues and inform decision-making.
- Address infrastructure overhead: Minimize your off-cloud footprint and enable testing from anywhere with a SaaS-based, self-contained integrated solution.

OpenText functional testing solutions

- OpenText™ Functional Testing: AI-powered test automation.
- OpenText™ Functional Testing Lab for Mobile and Web: Comprehensive mobile and device testing solution
- OpenText™ Functional Testing for Developers: Automated shift-left solution for functional testing.

Next steps: Achieve excellence in software quality and innovation
Discover how to enhance software testing for better app development and superior products.

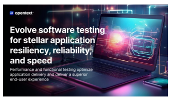
- Learn more about performance engineering
- Find additional information on functional testing

About OpenText
OpenText, The Information Company, enables organizations to gain insight through market leading information management solutions, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTEX) visit opentext.com.
opentext.com | X (formerly Twitter) | LinkedIn | CEO Blog

FAQ

- **Q: Why is software testing important?**
  A: Software testing ensures applications meet quality standards, identify issues early, reduce risks, and drive continuous improvements.
- **Q: What are the benefits of performance testing?**
  A: Performance testing helps evaluate application speed, reliability, and scalability under different conditions to optimize performance.
- **Q: How does functional testing contribute to software quality?**
  A: Functional testing verifies that each function of the application works correctly, ensuring overall software quality and reliability.

Documents / Resources

 opentext Evolve software testing for stellar application resiliency, reliability, and speed Performance and functional testing optimize application delivery and deliver a superior end-user experience	OpenText Evolve Software Testing For Stellar Application [pdf] User Guide Evolve Software Testing For Stellar Application, Evolve Software Testing For Stellar Application, Testing For Stellar Application, Stellar Application, Application

References

- User Manual