



# Automate Testing for Point-of-Sale Systems with Keysight Eggplant

# Enhance the Retail Experience with Continuous Testing

Retail point-of-sale (POS) systems continue to evolve from a customer convenience to an operations efficiency and market-shaping necessity. Today's POS systems need to interact seamlessly across a wide range of device configurations, serving a wider range of POS application features, and interacting with a wider range of cloud-based back-office systems. And in retail, response times are measured in milliseconds-to-seconds for every transaction.

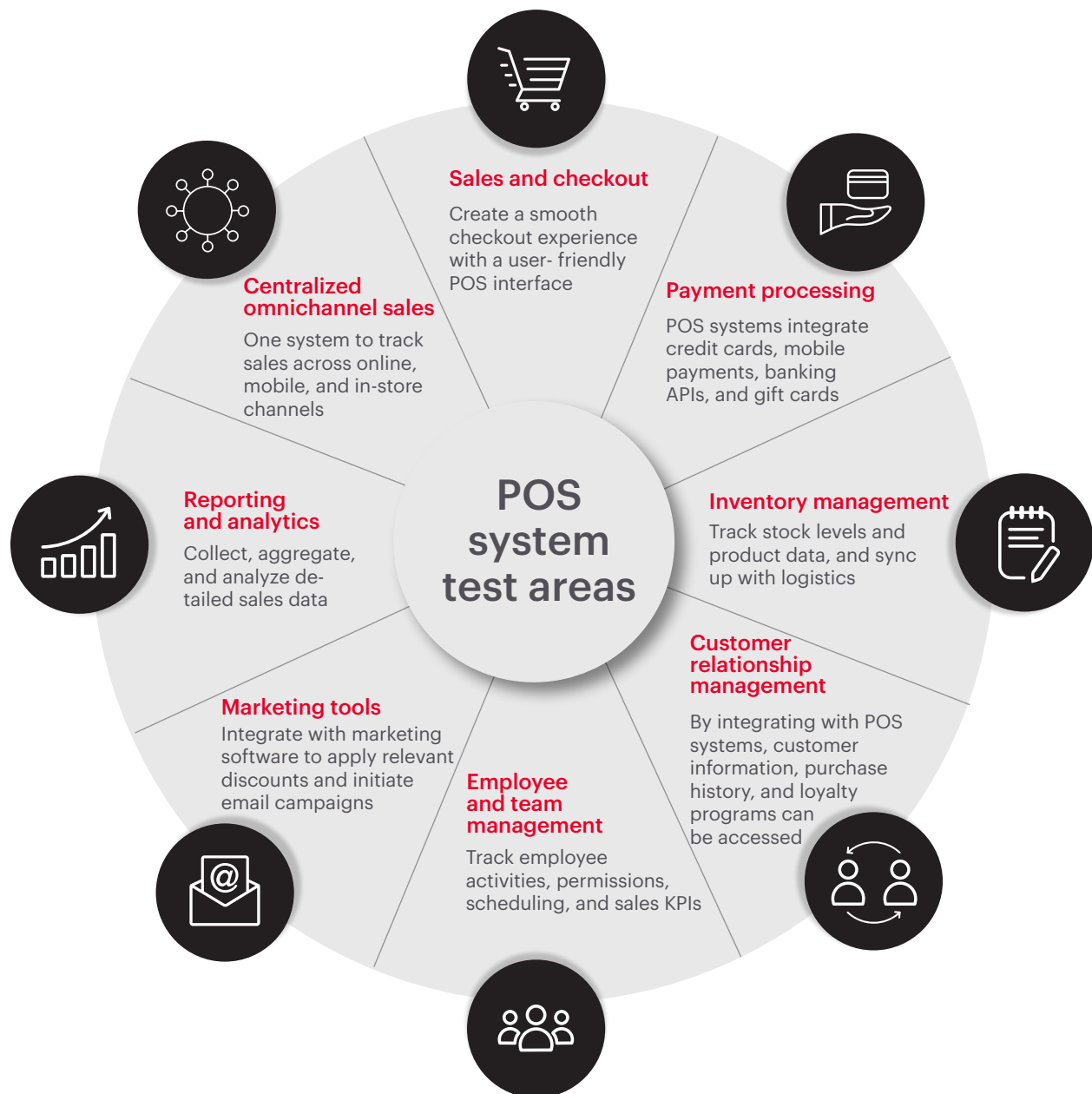
**Millions of possible tests can be reduced to a few hundred execution tests**

| Device testing                                                                                                                  | Application testing                                                                                                              | Back-office testing                                                                                                              |
|---------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"><li>• Hardware functions</li><li>• Customer GUI interfaces</li><li>• API interfaces</li></ul> | <ul style="list-style-type: none"><li>• Application functions</li><li>• Customer interactions</li><li>• Response speed</li></ul> | <ul style="list-style-type: none"><li>• Inventory recognition</li><li>• Customer recognition</li><li>• Fraud detection</li></ul> |

**Deploying safe and reliable retail point-of-sale across swipe, chip and pin, NFC, and mobile takes comprehensive software testing across a huge range of conditions and combinations – more automation, more recognition, faster response time, larger fraud attack surfaces.**

POS systems can range from simple credit card terminals to completely automated self-checkout solutions connected to continuously updated pricing systems, product optical recognition systems, customer record systems, inventory systems, and more. They can also deal with multiple languages, multiple sites, synchronized hardware and mobile device displays, complete ERP and cloud integration, and advanced features such as customer recognition and customization. They can also be just one part of the above, but regardless the test matrix can be huge to validate every feature, for every interaction combination.

Every retail store has a unique idea in mind for their approach to customer interaction as well as their own unique back-office systems. Pharmacies deal with regulatory and insurance issues as part of their POS design while gas pumps need to deal with a wide range of environmental conditions. The hardware each might select would have different application functions, graphical user interface (GUI) requirements, and application programming interface (API) specifications. The same is true for the application functions, customer interfaces and responses they need to collect, and the response speed they require. What they have in common is that every time they change features, upgrade, or update, the entire configuration must be tested end to end.



For retail POS systems of this size and scale, software testing involves three main areas:

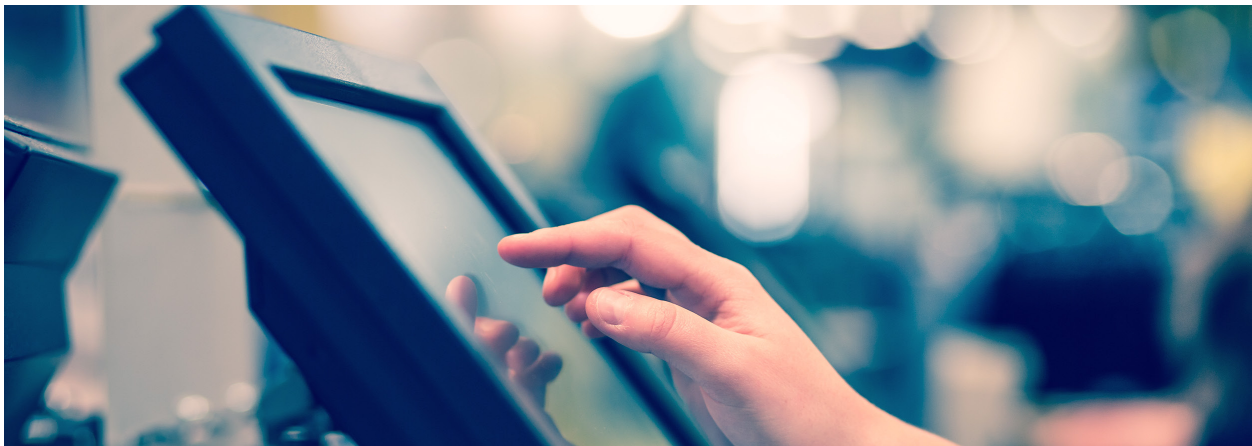
1. **POS hardware testing** — Most POS hardware has three main parts of its hardware configuration – a scanner/sensor, a customer/employee interface display, and software APIs that connect each terminal to the network. Testing of these hardware terminals often requires physical button pushing which can be replicated using robotic testing, and a KVM can be used to validate display interfaces across any operating system.
2. **POS software feature testing** — Every POS software system has a set of functions it is designed to deliver. This may involve sales and checkout, payment processing, inventory management, customer relationship management, employee and team management, marketing tools, and reporting/analytics. Each function of the POS software needs to be tested for functionality, vulnerability, speed, accuracy, and clarity. Testing each involves testing features under a wide range of input possibilities to ensure proper response even when being operated out of normal bounds.
3. **Back-office integration testing** — Delivering the POS software features requires interfaces to your company's unique technology stack, or techstack. Each company's techstack and API parameters are unique, so testing to your specific ERP system, CRM system, HRM system, payment processing architecture, security architecture as well as any business intelligence reporting systems need to be created. Testing the techstack involves testing expected as well as unexpected API data values looking for cases where the POS system experiences failures.

## Weeks Hours

Testing semi-annual **software updates** such as cloud vendor new feature releases  
software releases

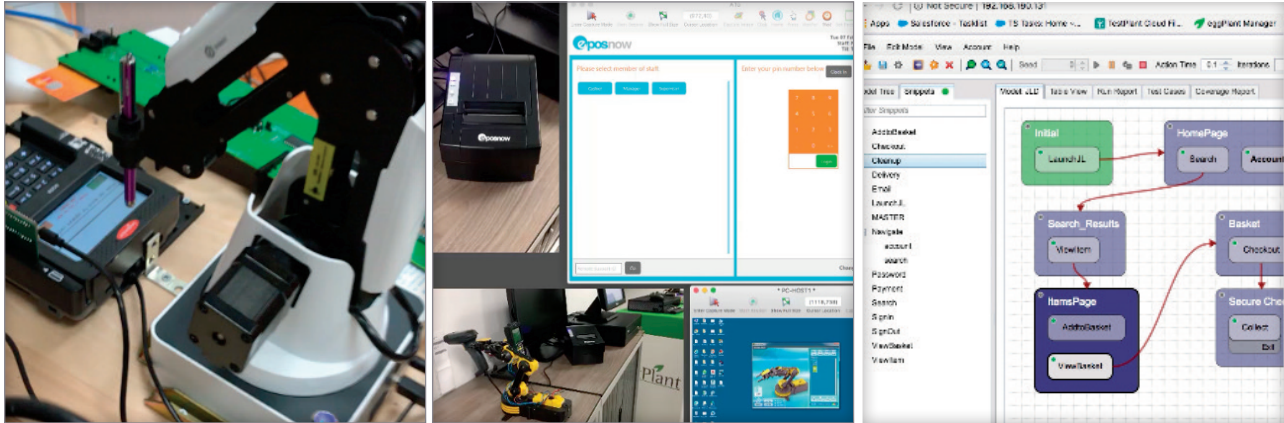
## Months > Days

Testing **workflow changes** such as adding curbside or touchless delivery



# Automating Software System Testing Changes Everything

**Automating tests can deliver months of complex manual testing in a few days**



Automating card readers

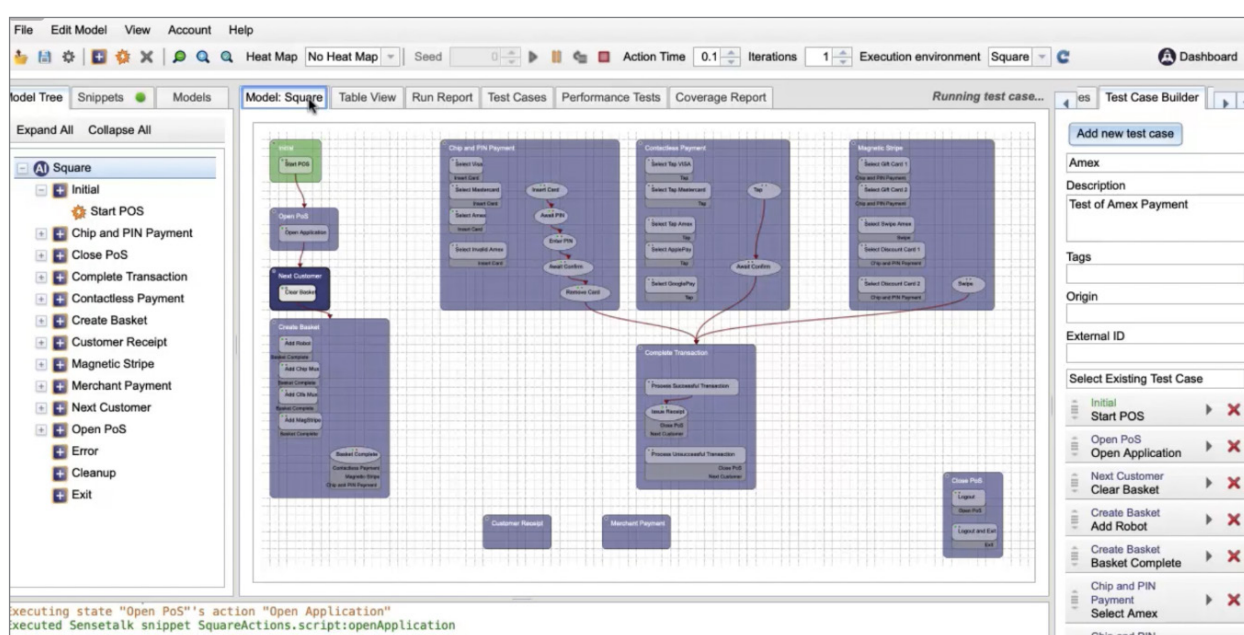
Automating POS software

Automating mobile apps

When choosing how to test a complex POS system, most companies create a scale model of the larger system and test using one of the following approaches:

- **Manual testing** — Using human testers to click buttons and see what happens or run individual function tests is the most intuitive method, but it is also the most time consuming and expensive method. The manual test method also suffers from documentation issues as human testers document differently, have trouble replicating tests, and suffer from test fatigue — a state where they are running tests via a checklist and not fully paying attention to the results.
- **Automated regression testing** — Most software developers are highly skilled in how to create embedded software test routines capable of loading and validating functionality of specific features and interfaces within their code. These tests are faster to execute but are limited to a subset of the total system functions, and always run the same tests. It also requires each test to be manually built and evolving the test breadth often requires the original developer. Testing this way can validate individual capabilities, but does not test the entire system, cannot evolve the tests being run based on results, and do not provide easily accessible robust analytics.
- **Operations monitoring** — Development teams who build and update the systems and operations teams who make sure the systems are operating properly typically use different evaluation approaches. They also often don't talk to each other and end up duplicating test/monitoring assets. Insights from production monitoring are rarely fed back into development to identify problems and optimize testing. For instance, one simple question that is rarely fed back involves how a POS system is actually used by operators and customers versus how it was intended. What common workflows are used and which are rarely touched.

Keysight's Eggplant software takes a different approach. Tests can be automated to evaluate each of the POS system parts: from the hardware display and API interfaces to the software functionality, to the techstack interfaces. One test platform can not only test each subsystem, but also the full customer experience including all of the edge cases. The Eggplant software monitors and validates the software APIs — regardless of the operating system its built upon — including ensuring the database or system of record is feeding back information within expected values. Eggplant also validates the critical human elements such as ensuring that any visual cues meant for employee or customer consumption are intuitive. That means the Eggplant software can spot problems like blue font on a blue background making a value unreadable, or one display window overlaying another obscuring a value.



Test every function, every interface, every output, every time. Eggplant can automate the full testing life cycle, creating a complete digital twin of your system. Testing more completely enables a simple means to define what should be tested by showcasing all the expected and unexpected user journeys. Powered by AI and machine learning algorithms, an Eggplant test system automatically determines what the next best test should be so you don't have to. Our fusion engine then does the hard work of controlling and understanding any system.

If you want to learn more about how to revolutionize testing your solution so you can get to market faster, more intelligently, and with more targeted outcomes, visit our [Eggplant software for retail](#) website and let us know about your testing challenges. We can change your world this month.

# Questions You Should Consider in Any Testing Approach You Choose

1. When you want to implement a change in your business workflow, how long does it take to test new features across your POS network?

As per “[State of DevOps – 2021](#),” when implementing DevOps for software development, elite performers deploy code faster, take less time to implement changes to features and workflows and restore a service when an incident or defect occurs, when compared to low performers. The key ingredient to DevOps is test automation.

2. How confident are you that your current test approach catches all the issues your employees might find?

In a recent survey from [Synopsys](#), shows that the use of open-source is continuing to grow, which mean open-source libraries are increasingly becoming the basis of numerous apps across all industries.

Regrettably, this means vulnerabilities are also on the rise — an average of 158 vulnerabilities were identified per codebase, which is the source code and libraries that underlie an application, service, or library.

And with 84% of codebases containing vulnerabilities, developers must understand the risks that accompany open source development and use.

Unfortunately for retail brands, their sector has the largest number of vulnerabilities and high-risk vulnerabilities, demonstrating the necessity of having a robust test automation solution in place.

3. How often is your POS software release frequency and how often do you implement changes to your workflows?

POS software can have 2 to as many as 12 updates per year with new features and interfaces which require system testing prior to deployment.

When you are ready to take these questions and more out of your business strategy, [talk to us](#) and let us show you how Keysight's Eggplant software can automate and accelerate your entire retail POS experience.

Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at [www.keysight.com](http://www.keysight.com).



This information is subject to change without notice. © Keysight Technologies, 2021 - 2023, Published in USA, March 22, 2023, 7121-1104.EN