

# How a Regional Health Plan Reduced Enrollment Errors by 92% and Cut Testing Time by 73%

## Success Metrics

**92%** Reduction in Enrollment Errors

**73%** Faster Test Execution

**50x** Larger Test Coverage

 Industry

Healthcare Insurance (Payer)

 Location

Southeast United States



# Table of Contents

3 Company Profile

4 When Over 3,000 Enrollment Forms Had The Wrong Benefits

4 Closing The Gap Between Testing And Reality

5 Coming Across TestMu AI To Solve The Test  
Coverage Problem

6 Starting Small, Testing Smart

6 Parallel Testing Changes Everything

7 Key Impact Metrics

7 Looking Forward

9 About TestMu AI



# Company Profile

## About:

A regional health insurance provider serves 380,000 members across four southeastern states, offering Medicare Advantage, individual marketplace plans, and employer group coverage.

The company built a reputation on personalized service and local expertise. Members could visit physical offices, call dedicated support teams, or manage everything through their online portal and mobile app.

But their digital platforms needed to work flawlessly, especially during the six-week annual open enrollment period when members made critical healthcare decisions for the coming year.

"We weren't failing. Our systems worked fine for most members. But when 8% of your enrollments have errors during open enrollment, that's thousands of people affected. We needed to understand what was happening on devices we didn't own."

## At a glance

### Industry :

Healthcare Insurance (Payer)

### Challenge :

During annual open enrollment, members reported benefits displaying incorrectly on specific Android devices and browsers. The company's internal testing setup couldn't replicate these issues before launch, leading to enrollment errors and frustrated members.

### Location :

Southeast United States

### Key Highlight :

TestMu AI's real device cloud and parallel testing helped the health plan test across **400+** actual member device combinations, reducing enrollment errors from **8%** to less than **1%**, cutting testing time from **11** days to **3** days, and improving member satisfaction scores by **34** points during their busiest period.

### Solution Used :

[Real Device Cloud](#)

[Automation Cloud](#)

[HyperExecute](#)

# When Over 3,000 Enrollment Forms Had the Wrong Benefits

The 2024 open enrollment period started smoothly. The QA team had tested the member portal across their standard test matrix: Windows desktops with Chrome and Edge, iPhones with Safari, and a few Android devices they kept in the office.

Two weeks into enrollment, member services started flagging a pattern. Callers using Samsung Galaxy phones with Chrome saw different copay amounts than what appeared in their confirmation emails. And members on certain LG devices couldn't see their prescription drug coverage tiers at all.

By the end of enrollment, the team had manually corrected over 3,000 enrollment forms where members had selected plans based on incorrect benefit information displayed on their devices.

The compliance team raised concerns, the legal team wanted documentation, and the member services team was overwhelmed.

"We tested everything we could with what we had. Five desktop configurations, three iPhone models, two Android phones from the office. We thought that covered our bases. Clearly, it didn't."

— QA Lead

## Closing The Gap Between Testing and Reality

The QA team analyzed what went wrong and discovered the following issues:

- **Device-Specific Rendering:** Samsung's Internet browser and certain versions of Chrome Mobile rendered their benefit comparison tables differently than expected, causing copay amounts to shift columns.
- **Screen Size Edge Cases:** Benefits displayed correctly on 5.5" and 6.1" screens but broke on 6.7" screens due to CSS media query breakpoints they hadn't tested.
- **Android Version Variations:** Android 12 handled its JavaScript benefit calculator differently than Android 13, causing prescription tier displays to fail.

The problem wasn't that their platform was broken. It worked perfectly on the devices they tested. The problem was that they couldn't test on the hundreds of device-browser-OS combinations their members actually used.

Their internal test lab had 8 devices. Member analytics showed their users accessed the portal from 200+ distinct device configurations.

# Coming Across TestMu AI to Solve The Test Coverage Problem

The digital team needed a solution before the next open enrollment cycle. They evaluated three cloud testing platforms against specific requirements:

1. Real devices, not emulators and their issues only appeared on actual hardware
2. Browsers members actually used, including Samsung Internet and older Chrome versions
3. Parallel execution to test faster without adding QA headcount
4. HIPAA-compliant infrastructure for testing with member data patterns
5. Integration with their Jenkins CI/CD pipeline
6. Clear audit logs for compliance documentation
7. Support team that understood healthcare regulatory requirements

"We chose TestMu AI because they had the actual devices we needed to test on. Not simulations. Real Samsung Galaxy S21s, S22s, S23s. Real LG Velvet phones. Real Pixel 6 and 7 devices. That's what our members use."

— Director of QA

# Starting Small, Testing Smart

The QA team didn't migrate everything at once. They started with a two-week pilot focused on their benefit comparison page, the page that had caused the most problems.

## Week 1:

- Connected TestMu AI to their staging environment
- Ran existing Selenium tests on 50 device configurations
- Found 12 rendering issues they'd never caught before

## Week 2:

- Expanded testing to the enrollment flow
- Created a test matrix based on actual member device data from analytics
- Set up parallel execution for regression testing

The team found issues immediately: the same CSS problems, JavaScript calculation errors, and screen size bugs that had plagued the previous enrollment period. But now they could see them, reproduce them, and fix them before launch.

## Reduced Test Cycle Time from 11 Days to 3 Days

Manual testing previously took 11 days per release cycle. The team would run through scenarios sequentially: desktop browsers, then iPhones, then Android devices, one configuration at a time.

With [TestMu AI's HyperExecute](#), they ran tests across hundreds of configurations simultaneously. The same test suite that took 11 days, now completed in 3 days.

More importantly, they could test every release candidate instead of just the final build. If developers pushed a fix for a Samsung-specific bug, QA could verify it across all Samsung devices in minutes, not days.

"We went from 8 devices in our office to testing on the exact phones and tablets our members use. That's the difference between guessing and knowing."

— QA Lead

# Key Impact Metrics

- Testing time reduced from 11 days to 3 days per release cycle
- Enrollment error rate dropped from 8% to less than 1%
- Test coverage expanded from 8 devices to 10,000+ device configurations
- Member satisfaction scores increased by 34 points during open enrollment
- Complete compliance documentation with audit logs for all tested configurations

# Looking Forward

Testing is a competitive advantage for the company now. The health plan can release updates weekly instead of monthly. They can test new features on actual member devices before committing to development. They can demonstrate compliance through comprehensive test coverage documentation.

Most importantly, they can trust that when members need to make critical healthcare decisions during open enrollment, the platform will work exactly as expected, regardless of which device they're using.

Want to ensure your healthcare platform works across every device your members use? [Book a demo with TestMu AI](#) and discover how real device testing can transform your member experience.

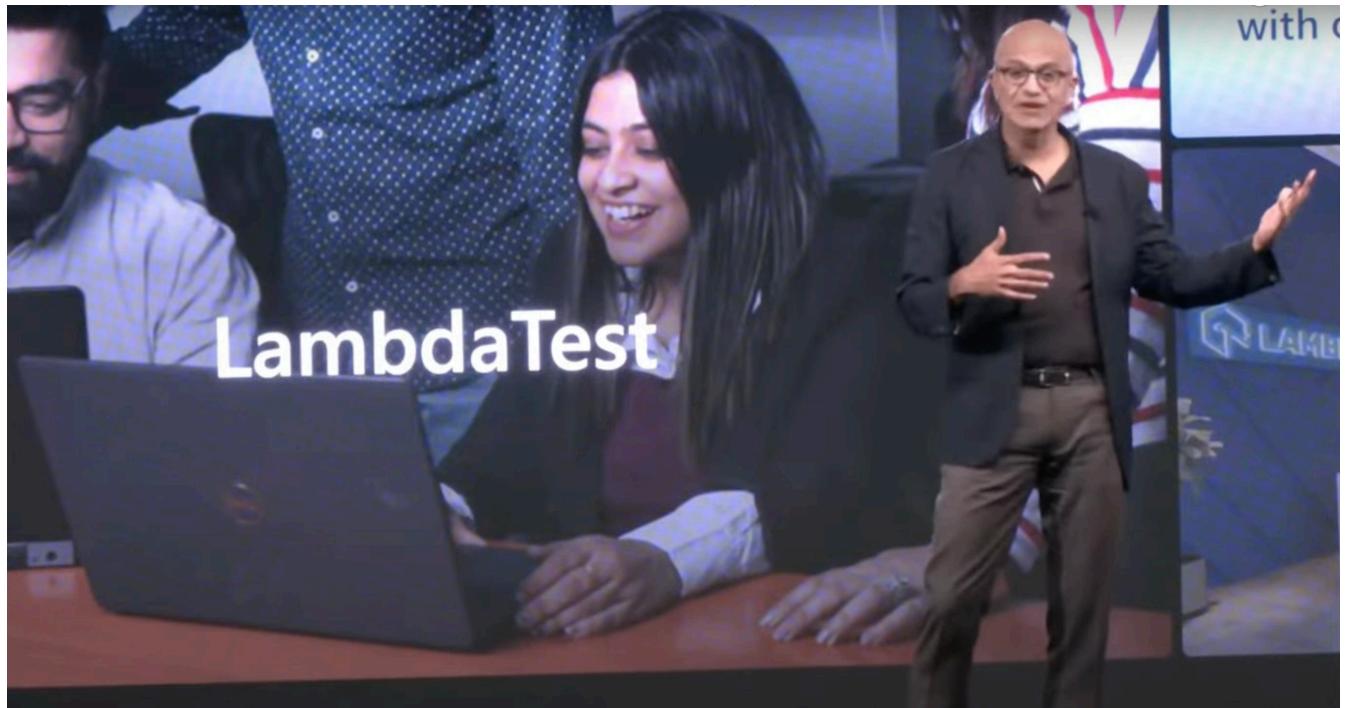


Image Source : Microsoft Future Ready Event

66

**LambdaTest(Now TestMu AI) is creating that next level of efficiency around test automation so that people can actually focus on testing versus test orchestration.”**

Satya Nadella, CEO, Microsoft

# About TestMu AI

TestMu AI (Formerly LambdaTest) is a fully autonomous agentic quality engineering platform that empowers teams to test intelligently, smarter, and ship faster. Over 10,000+ customers and 2 million+ users across 132+ countries rely on TestMu AI for their testing needs.



**1.2 Bn+**

Tests



**2M+**

Users



**10K+**

Enterprises



**132+**

Countries

## **Exploratory Testing**

Enhance web and app quality to ensure seamless user experience with real-time, live, exploratory testing on 10,000+ devices.

## **KaneAI**

Boost testing efficiency with an AI platform that uses natural language to create, debug and evolve tests.

## **Test Manager**

Streamline test creation, management, & reporting for improved efficiency with AI - native unified Test Manager.

## **Automation Cloud**

Accelerate product releases with secure, scalable, end-to-end test automation in the cloud.

## **Real Device Cloud**

Test on 10,000+ real Android and iOS devices, and 3000+ browser combination cutting costs while ensuring compatibility.

## **HyperExecute**

Accelerate testing speed by 70% with AI-Native orchestration for faster digital transformation.

## **Accessibility Testing**

Ensure inclusive, accessible websites with TestMu AI's manual and automated Accessibility Testing tool.

## **Visual UI Regression**

Achieve UI perfection quickly with AI-Native visual regression testing across all platforms.

**TestMu AI**  
Formerly LAMBDATEST

+1 (866)-430-7087

[www.testmu.ai](http://www.testmu.ai)

[sales@testmu.ai](mailto:sales@testmu.ai)

**TEST INTELLIGENTLY. SHIP FASTER.**

