PHYSICAL TESTING & VALIDATION · PROTEUS

Things that never happen keep happening: how ASAP Testing replaced dozens of manual tests in Amazon's Proteus perception program.

When Amazon Robotics needed to prove that Proteus — its first fully autonomous mobile robot — could safely perceive the people and obstacles around it, the testing plan was rigorous, approved by TÜV Rheinland in advance, and executed almost entirely by hand. Engineers manually positioned standardized test objects around a moving robot, ran dozens of repetitions per test point, then spent weeks compiling the data into 200-plus-page reports. **Fennec's automated test cells replaced that process end to end:** connected directly to the robot and its environment, physically repeatable, and audit-ready reports generated in hours instead of weeks.

AT A GLANCE

INDUSTRY	Advanced Robotics / Autonomous Mobile Robots
APPLICATION	Physical verification and validation of perception at the full-system level
STANDARDS	Test methodology agreed with TÜV Rheinland; standardized test pieces per IEC 61496
SOLUTIONS	ASAP Testing (automated physical test cells) + ASAP Platform (T2-qualified tools in the validation model)
OUTCOME	Manual perception testing and reporting replaced with automated, repeatable physical verification; re-verification per release cycle became routine instead of a program event

3 days	64	2 wks → hrs	Human out
Full end-to-end test execution by tech operators, data processing running in parallel	Tests per test plan — the manual repetition load per configuration, automated end to end	Report compilation, from hand-built 200+ page document to auto-generated reports	No personnel in the sensor's field of view; no approximate positioning; no drift

The challenge: rigorous by design, manual by necessity

The Proteus perception test plan was exactly what a certification body wants to see. Test scenarios approved by TÜV Rheinland in advance. Standardized test objects built to the relevant ISO/IEC specifications. Static tests at defined positions around the robot, dynamic tests with objects moving into the path of a robot in motion, and environmental challenge conditions layered on top: lighting variations, flashing beacons, pollution in front of the sensors — anything the robot would plausibly encounter in a fulfillment center.

The problem was the execution. Every one of those tests was performed manually. An engineer physically moved the test object into position, judged "close enough" by test-plan alignment. The multiplicity compounded fast: multiple test objects, multiple positions around the robot, separate dynamic sequences, and environmental conditions — some requiring multiple phases that each meant manually running the test, resetting the robot, running it again, and running it once more without the challenge condition. The team performed these across multiple test plans, some containing over 60 tests.

Then came the reporting. After the physical work was done, an engineer spent approximately two weeks compiling the collected data and formatting it into a Microsoft Word technical report that ran past 200 pages — all by hand.

And none of it was one-time. Every firmware update, every revision to the detection software, every hardware change meant re-running the full suite or carefully scoping the affected subset. For a robot on a modern release cadence, the manual approach didn't just cost time; it put a ceiling on how often verification could realistically happen.

The hidden cost: every repetition is a different test

Beyond the hours, manual testing carried a structural validity problem. A person holding a test object appears in the robot's field of view, contaminating the very perception scene under test. Hand placement is approximate — and approximate placement means the limits of the system are never tested at the actual limits. And human performance drifts: the way an engineer runs repetition 60 differs from repetition 6, which means the test conditions themselves change across a run.

> For a safety case whose entire purpose is demonstrating repeatable, bounded system behavior, the least repeatable element in the loop is *the manual nature of the test method itself.*

There was a talent cost, too. Highly trained safety engineers were spending their weeks positioning foam test pieces and formatting documents — expensive expertise consumed by administration, exactly the pattern that burns teams out in the final push before a release.

The solution: an instrumented test cell

Fennec's ASAP Testing cells automate the entire verification chain. The cell connects directly to the robot and the test environment: it commands the robot's movements, positions and moves the test objects with instrumented precision, tracks ground-truth positions continuously, and extracts the robot's own data — trajectories, sensor output, detection results — in real time.

01 Exact, identical placement

Test object placement is exact and identical on every repetition, so the system's limits are tested at the limits.

02 A clean perception scene

No human appears in the sensor's field of view, so the scene under test is never contaminated by the tester.

03 A repeatable script

The tedious manual sequence becomes a script — environmental challenge sequences (run, reset, re-run, baseline) execute in order without variance.

04 Reports that assemble themselves

Fennec's Ardvarc data-processing module compiles results against ground truth automatically and generates standardized, audit-ready reports in a format the assessor already knows.

In practice, Amazon Robotics tech operators completed full end-to-end testing in three days, with data processing and reporting running in parallel with the test campaign. The gap between the last test and the finished report shrank to nearly nothing.

Verification that survives the next release

Because the cell is automated and repeatable, re-verification stops being a program event and becomes routine practice. When firmware changes or detection software updates, the affected test suite re-runs at the pace of the release cycle — monthly or quarterly — without pulling engineers off development work. This is the functional safety V-Model in practice: safety verification as a continuous loop that travels with the system through its operating life, not a gate it passes through once.

There's a design lesson embedded in the program, too — one Fennec's team emphasizes with every builder: **testability has to be designed in.** When the robot's data (motion, states, detection results) is accessible from the start of development, verification cascades smoothly to the end of the program. When it's not, teams pull teeth late in the game, forcing diagnostics into architectures that don't support them. The builders who internalize "how will we prove this is safe?" on day one collect the dividends at release.

Why it matters beyond Amazon

Amazon tested Proteus this rigorously because its robots share floors with its own workforce. But the case for physically verified safety is universal, and coming into sharper relief as the industry expands.

Trust is the product

A robot marketed as taking on grueling jobs still has to answer the worker's question: will it hurt me when it turns around? On floors with organized labor, "we used safety-rated components" has never been an acceptable answer; documented system-level testing in conditions that approximate the working environment is what earns the deployment. In an industry where a single viral video of a misbehaving robot sets perception back years, demonstrable safety is a collective asset.

Liability is the forcing function

An insurer facing a claim doesn't ask whether the robot was built from good components; it asks whether the operator exercised due care. Documented, standardized, physically executed verification — in a form an NRTL or certification body can review — is what due care looks like on paper. Structured physical verification is on track to become an insurance expectation for collaborative deployments.

Safety and quality are converging

Robotics is following the path automotive and aerospace already walked: past a certain point of public deployment, a product is not considered high-quality unless it is demonstrably safe. The builders who treat verification as a product capability rather than a compliance cost are the ones who will be trusted with the environments where the market is heading.

Why it worked

01 Ground truth over generated truth

Physical testing with instrumented ground-truth tracking eliminates both the systematic-error blind spot of virtual-only verification and the approximation error of manual testing.

02 The whole system in the cell

Verification at the integrated-system level — where safety actually lives — under the environmental conditions the robot will actually face.

03 Built for iteration

Re-verification at release cadence, without consuming the engineering team. The safety case stays current with the machine.

04 Engineers doing engineering

Safety experts analyze results instead of holding test objects and formatting documents. The expertise the program paid for is the expertise the program gets.

Key standards & qualifications: Test methodology agreed with TÜV Rheinland · Standardized test pieces per IEC 61496 · ASAP Platform T2 qualified by TÜV Rheinland