Visteon®

NEWS RELEASE

How Visteon Engineers Saved 24 Hours Per Week with AI

2025-07-29

Sometimes the best solutions come from engineers who are just tired of doing the same thing over and over again.

Let's be honest: writing test cases isn't the most exciting part of automotive software development. But it's absolutely critical. When you're working on cockpit systems that drivers rely on every day, comprehensive testing isn't optional—it's everything.

Our RoboFIT team was facing a problem that probably sounds painfully familiar to anyone in automotive software. Every new program meant manually creating 100-200 diagnostic test cases. With a limited number of engineers per project handling both test design and script development, we were constantly playing catch-up.

The "There Has to be a Better Way" Moment

The breaking point came during a particularly demanding project cycle. We were spending entire days just parsing through JDA requirements documents, trying to extract the key information needed for test case creation. Then more time writing the actual test cases. Then even more time generating the RoboFIT scripts.

One of our engineers put it bluntly: "We're basically doing the same pattern recognition and generation work that computers should be able to handle. Why are we doing this manually?"

That question sparked what became vGenTest.

Building Something That Actually Solves the Problem

Instead of jumping straight into the latest AI trends, we stepped back and analyzed what we were actually doing when creating test cases. We realized our process had two distinct phases:

Reading through specifications, extracting key diagnostic needs, understanding the context to understand requirements.

Creating test cases that map to those requirements, then writing RoboFIT scripts that actually work with our libraries

The AI solution needed to handle both phases, automate the writing and understand the automotive context. That’s why we built vGenTest around a model that we trained specifically on automotive diagnostic requirements, pulling information from our tested library functions.

Results That Actually Matter

Here's where it gets interesting. We're not just talking about marginal improvements—the results were significant enough that people started paying attention:

Time savings: We're looking at about 24 hours saved per week when fully deployed. That's not just efficiency—that's giving engineers back nearly a full day to work on actual innovation instead of repetitive documentation.

Cost impact: 75% reduction in the resources needed for test case generation. In an industry where margins matter and time-to-market is critical, that's substantial.

Quality improvement: This one surprised us. The AI-generated test cases consistently showed better coverage and caught edge cases that manual processes sometimes missed. We're measuring about 90% improvement in completeness and accuracy.

Building Trust Through Transparency

Let's address the elephant in the room: engineers are rightfully skeptical of AI-generated code, especially for something as critical as automotive diagnostics. That’s why we built a structured review and validation process into the system from day one. Every AI-generated test case is peer-reviewed before deployment, ensuring both accountability and quality.

This approach is all about unlocking the potential of our people. Interns and new joiners are now actively involved in meaningful projects, using AI-generated outputs as a foundation to build skills, contribute fresh ideas, and gain confidence faster.

What started as a lightweight, cost-effective tool is also laying the groundwork for a larger, scalable solution. The team remains committed to improving accuracy and functionality through ongoing development. So far, feedback has been overwhelmingly positive: engineers appreciate focusing on review and refinement — instead of reinventing the wheel every time.

What's Next

This is still a proof of concept, but the results are promising enough that we're planning full-scale deployment. We're looking at integrating with Azure AI for more robust model capabilities and expanding the end-to-end automation pipeline.

We're also exploring how this approach could work for other aspects of automotive software development – anywhere we have repetitive pattern recognition and generation tasks that could benefit from AI assistance.

Why This Matters for Automotive

The automotive industry is rapidly shifting toward software-defined vehicles. The standout cockpit experiences we see today rely on robust, thoroughly validated software. Tools like vGenTest not only improve process efficiency but also expand testing coverage, ultimately resulting in more reliable and cutting-edge vehicle features. Our goal is to free engineers to focus on the creative, complex problem-solving that truly fuels innovation in mobility.