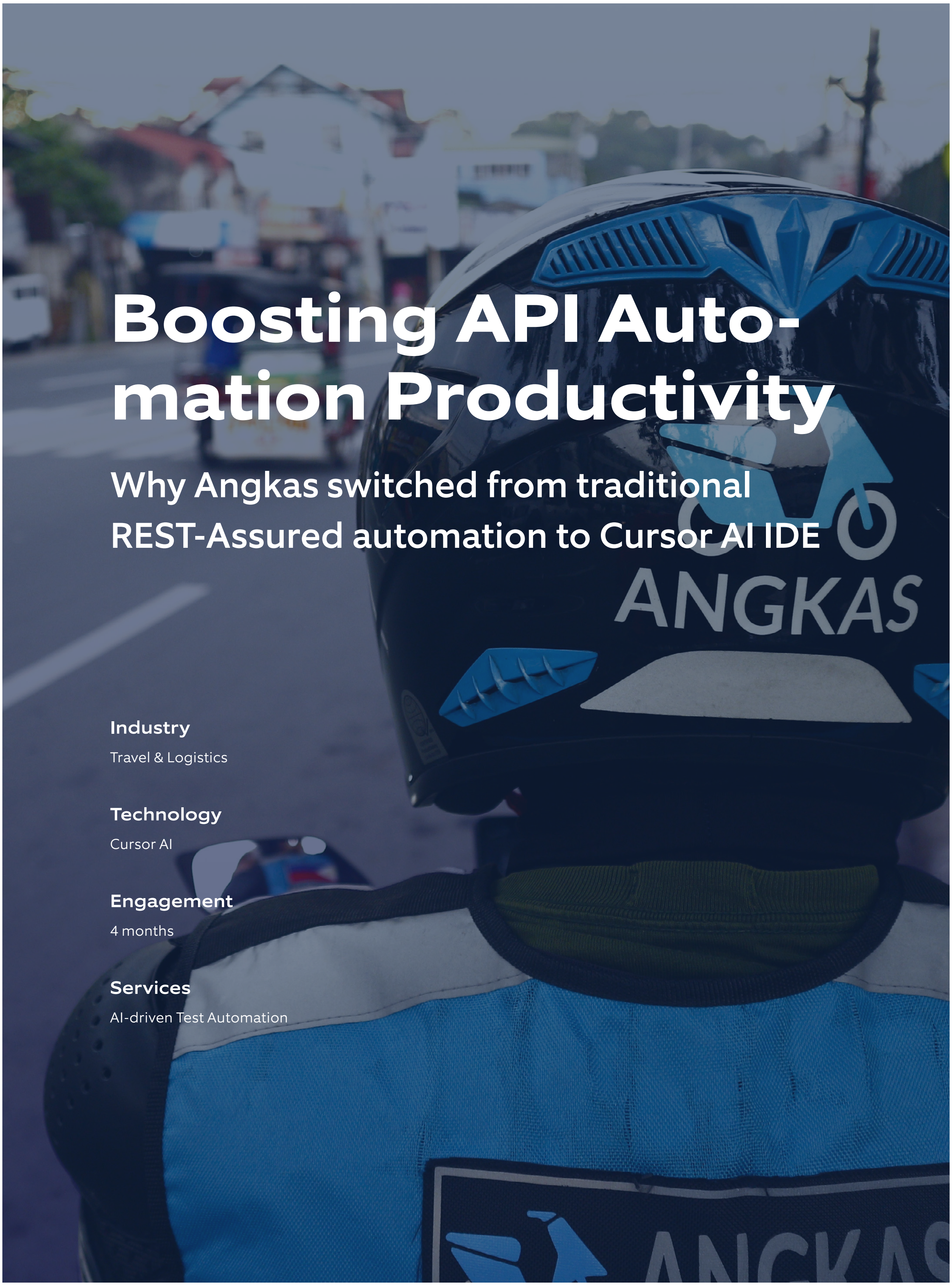

Boosting API Automation Productivity

Why Angkas switched from traditional REST-Assured automation to Cursor AI IDE

Industry

Travel & Logistics

Technology

Cursor AI

Engagement

4 months

Services

AI-driven Test Automation

The story

Angkas is Philippines' first app-based motorcycle ride-hailing platform that operates in a fast-moving digital environment where frequent deployments and consistent customer experiences are critical to success. To ensure reliable delivery of API-driven features, Angkas built a robust Java-based API test automation framework using **REST-Assured** and **Gradle**, spanning over **500 test cases across dozens of REST APIs**.

While traditional REST-Assured implementation initially supported the team's productivity goals with boilerplate suggestions, its limitations began to surface as automation efforts scaled. In response, the project team adopted **Cursor AI IDE**, a project-aware, AI-native coding environment that helped the engineering team **accelerate test automation by 30–50%**, reduce manual overhead, and maintain architectural consistency — all without changing the existing framework or toolchain.

The challenge

As Angkas expanded its API footprint, its engineering team encountered key bottlenecks with the traditional REST-Assured-based automation approach:

- **Repetitive and time-consuming implementation**
  Developers had to manually write boilerplate code for payloads, headers, assertions, and utilities for each new endpoint, which slowed down test creation.
- **Lack of reusability and standardization**
  Without centralized utilities or templates, test scripts varied in structure and quality, leading to inconsistency across hundreds of test cases.
- **High maintenance overhead**
  Updating tests across multiple files and modules required extensive manual effort, increasing the risk of errors and reducing efficiency.
- **Slow onboarding and ramp-up**
  New team members found it challenging to navigate and understand the framework quickly due to fragmented test logic and inconsistent patterns.

These small but recurring inefficiencies began to slow down the team's ability to add new APIs, expand coverage, and keep pace with agile delivery timelines.

The solution

The project team adopted **Cursor AI IDE**, an AI-enhanced development environment built on top of VS Code. Cursor introduced a transformative experience for the QA automation team, enabling them to scale test coverage faster and more reliably without disrupting their existing tech stack.

Key improvements with Cursor AI IDE:

- **One-click accept/reject for full code suggestions**
  Cursor allows complete class or method suggestions to be previewed and integrated instantly, eliminating piecemeal coding and unnecessary copy-paste.
- **Project-wide awareness**
  Cursor understands the structure and dependencies of the entire test automation framework, generating suggestions that align with existing naming conventions, utilities, and design patterns — something difficult to achieve with manual scripting.
- **Inline chat with live code editing**
  Developers can highlight any section of code and issue commands like "Convert this into a parameterized test" or "Add assertions for status code," with results applied directly within the editor.
- **Seamless, embedded experience**
  Cursor operates as an intelligent assistant inside the coding workflow, rather than a detached tool — supporting faster iterations and fewer context switches.

Real-world application: Faster API test creation

The project team used Cursor AI to significantly streamline the creation of new API test cases while maintaining code quality and structure.

Tech stack

- **Language:** Java
- **Framework:** REST-Assured
- **Build Tool:** Gradle

Tasks enhanced by Cursor AI

- Generating boilerplate test code for new endpoints
- Writing GET/POST logic with headers, query params, and payloads
- Creating helper utilities and request/response models
- Adding assertions and validations (status code, body, headers)
- Generating data-driven tests
- Refactoring repetitive blocks into modular helpers
- Updating constants/configurations in appropriate files with awareness of existing structure

Use case

To add test coverage for /products/search and /products/details, the team prompted Cursor to replicate the structure of an existing test (/users/list), asking it to create all relevant components.

Cursor AI responded by

- Creating two new test classes with correct annotations, imports, and naming conventions
- Generating or updating utility classes for payloads and validations
- Adding new constants and test data values in existing shared files
- Suggesting architectural improvements (e.g., moving common logic to shared helpers)
- Integrating the tests into the Gradle test suite without disrupting existing setup

This end-to-end generation would previously have required multiple steps across different files and teams, which Cursor handled easily and seamlessly.

Feature comparison: Traditional REST-Assured approach vs. Cursor AI IDE

Feature	Traditional REST-Assured approach	Cursor AI IDE
New API Test Creation	Manual effort for setup and structure	~30–50% faster with context
Code Suggestions	None; developers write from scratch	Project- and pattern-aware
Refactoring Support	Manual across files and classes	Interactive and in-place
Copy-paste Requirement	Frequent for repeating structures/utilities	Rare
Developer Productivity Boost	~10–15% (depending on reuse)	~30–50%

The outcome

By switching to Cursor AI IDE, Angkas expects to achieve significant efficiency gains and quality improvements in their test automation processes:

- **30–50% faster onboarding and automation** for new APIs
- **Reduced manual effort** through context-aware code generation
- **Fewer copy-paste cycles**, improving development flow and reducing errors
- **Consistent, scalable architecture** across over 500 test cases
- **Improved refactoring and framework maintainability**

These results helped Angkas shorten release timelines, expand test coverage faster, and reduce bottlenecks across their CI/CD pipeline.

Key learnings

- **Traditional REST-Assured automation is reliable** but becomes inefficient at scale, especially when dealing with hundreds of APIs and repetitive test logic.
- **Cursor AI IDE delivers real impact** in complex, repeatable tasks like test automation, where structure, consistency, and speed are crucial.
- **Embedding AI in the IDE workflow** by not just being a code suggester but as an active assistant. This unlocks next-level developer productivity.

Conclusion

For large-scale automation projects, where hundreds of API test cases need to be written, updated, and maintained within a structured framework, **context-aware AI development tools make a measurable difference**. Cursor AI IDE helped Angkas move beyond generic code generation and toward intelligent, architecture-aligned automation which led to better speed, consistency, and scale.

Testimonial

"At Angkas, our AI Center of Excellence has set a clear vision: integrate AI meaningfully across the software development lifecycle to accelerate delivery while maintaining quality. The Nagarro team fully embraced this direction. By adopting Cursor AI, they significantly improved our test automation velocity and helped us scale test coverage. This has become an increasingly important focus as AI enables faster code generation, which puts additional pressure on QA to keep pace.

In addition to automation, we introduced the use of AI to support onboarding. This included using AI tools to explain complex codebases, generate UML diagrams, and create system visuals to make onboarding faster and more efficient. The Nagarro team adopted this approach seamlessly, incorporating these practices into their workflow in alignment with our AI-first engineering direction.

Nagarro demonstrated a deep understanding of our AI strategy and translated it into real, day-to-day impact. Their work has been instrumental in helping us scale our engineering capabilities with greater speed and consistency."

Mark Basmayor
Head of Software Engineering, Angkas

About Nagarro

Nagarro helps future-proof your business through a forward-thinking, fluidic, and CARING mindset. We excel at digital engineering and help our clients become human-centric, digital-first organizations, augmenting their ability to be responsive, efficient, intimate, creative, and sustainable. Today, we are around 17,500 experts across 39 countries, forming a Nation of Nagarrians, ready to help our customers succeed.

For more information, visit **www.nagarro.com**.