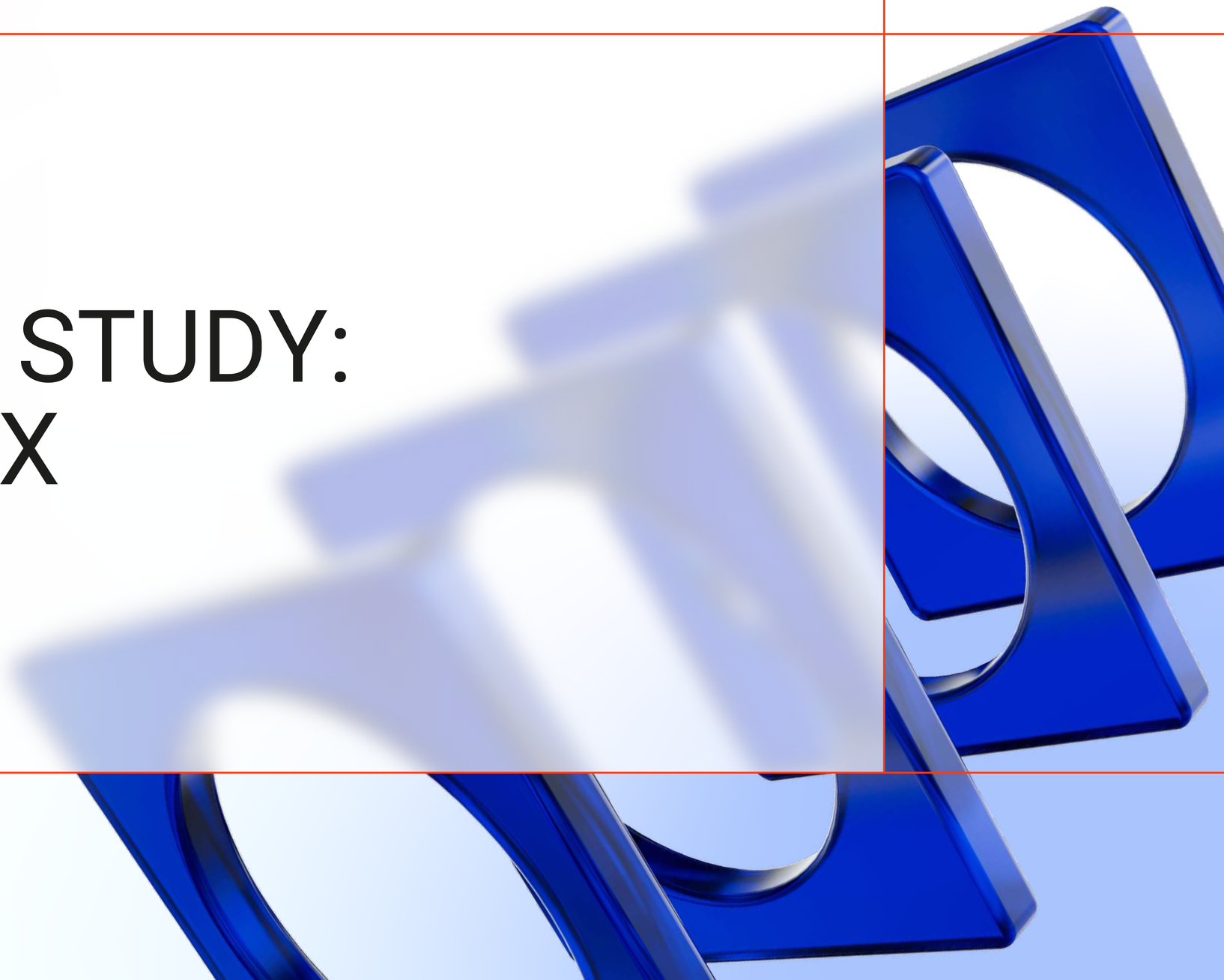
aimprosoft

CASE STUDY: UBIDEX



aimprosoft

Engineering an AI-powered QA framework for AdTech

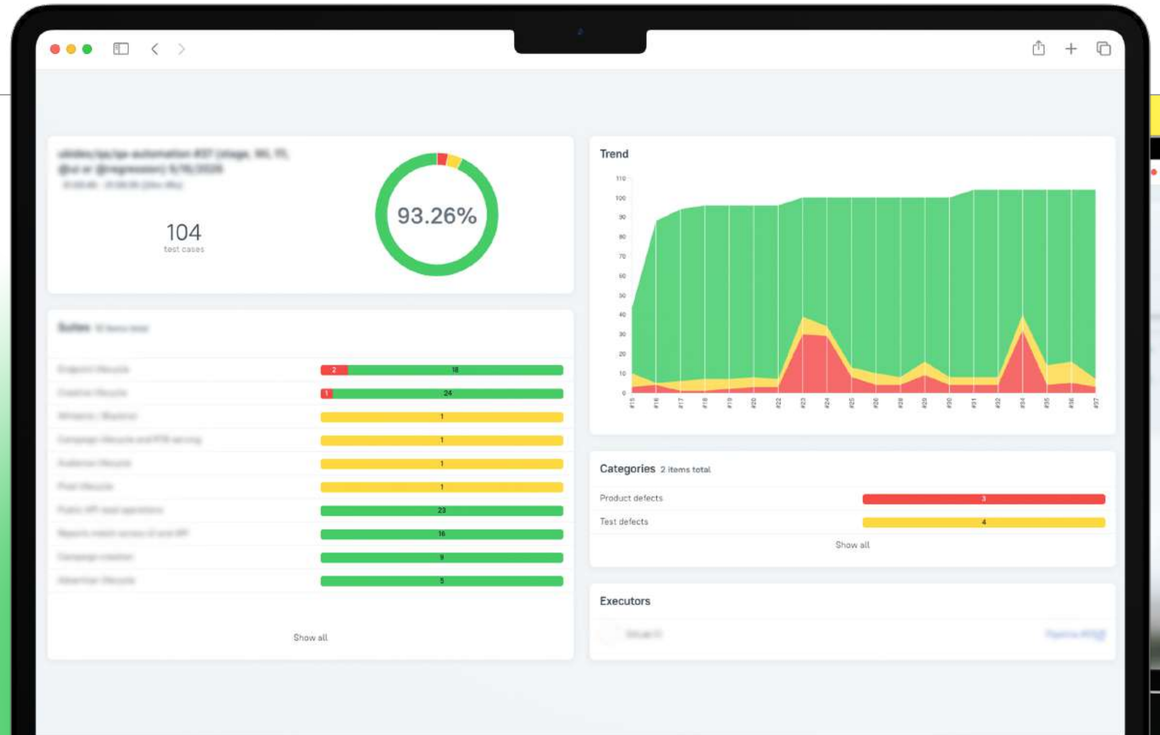
Partner information:

Partner: **UBIDEX**

Headquarters: **Europe (under NDA)**

Team size:

Delivery manager: 1
AI/QA specialists: 2
UI developer: 1
Software developer: 1





PROJECT OVERVIEW

UBIDEX is an AdTech platform that helps brands increase user value by offering automated ad buying and retargeting based on audience analytics. To keep up with the fast-moving market, the company needed to ship new features and releases on a tight, consistent schedule.

The company realized it couldn’t sustain the pace of product evolution with its existing capacity and processes. Largely manual testing and release verification had worked fine for a smaller-sized operation but were incredibly difficult to scale. UBIDEX wanted to accelerate testing and development not by hiring more people, but by improving current processes. This could help the team do more with the same capacity.



Considering AI as a possible solution, UBIDEX partnered with our team to figure out how it could solve the bottleneck. Together, we identified test automation as the most practical opportunity for AI application in their engineering workflow and selected Claude Code as the key tool for the new approach.

We designed a domain-oriented framework, using specialized Claude Code agents to speed up framework creation without breaking the overall architecture and design principles. Framework integration with the existing GitLab pipeline automated smoke testing in production and enabled faster release verification.

Today, we continue our collaboration on further test automation and framework improvement.

SERVICES & TECHNOLOGY STACK

Provided services

- AI consulting
- AI implementation
- QA & software testing
- Product engineering

Technologies & tools

- **Backend:** Java
- **Testing frameworks:** Cucumber, Gherkin, Selenium, RestAssured, Allure
- **DevOps:** GitLab, Maven
- **AI tools:** Claude Code, Claude, Claude in Chrome



REQUIREMENTS & CHALLENGES

UBIDEX wanted to increase test coverage and release quality, and all that shouldn't affect the speed of development or require hiring new QA specialists. Given that releases should go out every two weeks, even small inefficiencies and delays in testing and task completion quickly compounded into significant time loss. The team also had no automated test coverage, so they needed to establish a foundation for automation without interrupting active sprint work.

Creating a test suite was only part of the challenge. The framework had to be reusable and maintainable as coverage grew, support both UI and API testing, and fit the team's current engineering workflow to simplify the transition. In addition, introducing AI into the development process called for well-defined architectural boundaries so AI-generated code would follow the same design rules as the rest of the framework.

Our team had to address the following challenges:

Laying the foundation for test automation

The client team relied entirely on manual testing and had no existing automated test coverage. So the framework needed to provide a ready-to-expand foundation for UI and API testing that wouldn't take long to set up.

Making tests reusable

Test scenarios had to be built like building blocks—reusable, composable behavior-driven development (BDD) steps—so the team could add new coverage using the same logic.

Integrating AI into the engineering workflow

Claude Code had to fit the team's existing development environment and advance framework development through specialized AI agents.

Automating release verification

Replacing manual smoke checks with an automated suite running through the existing CI/CD workflow, with room to expand into regression coverage.

Preventing inconsistency in AI workflows

AI-generated code had to follow the same architectural rules, reuse existing business logic and framework components wherever possible, and separate UI/API interactions from domain logic.

COLLABORATION APPROACH & PROCESS

UBIDEX partnered with Aimprosoft to get AI consultation, but the engagement soon grew into a broader engineering collaboration. After assessing their workflows and potential AI use cases, we **identified test automation as the highest-value opportunity** and recommended focusing the next stage of the engagement on it. The project **paired human engineering with AI execution.** We worked on the framework architecture and technical strategy, using Claude Code for much of the implementation **with strict boundaries.**

At the start, we aligned with the UBIDEX team on the platform architecture, testing approach, and the framework's main requirement to be **reusable and composable.** On our side, the delivery manager led the framework architecture, and two QA specialists populated it with test coverage and **validated Claude's output.** The UBIDEX team provided the product and platform context needed to assess the output and confirm it matched platform behavior.

Once the initial framework and smoke tests demonstrated value, the UBIDEX team decided to expand the approach. That early success **turned a single automation project into an ongoing partnership.**

STEP-BY-STEP PROJECT FLOW

Stage 1. Architecture and framework design

We studied the UBIDEX platform and aligned on the testing approach and framework requirements. Because test reusability was a key requirement, we structured the framework around the product's business domain, dividing it into four layers with concrete responsibilities, and outlined rules for implementing and reusing UI and API interactions. These rules then serve as architectural boundaries for Claude Code.





Stage 2. Claude Code implementation

We used Claude Code to build the core framework elements. This included a working Maven project with Cucumber, Selenium, RestAssured, Allure, and JUnit 5. We also created reusable BDD steps that could work across UI and API interactions.

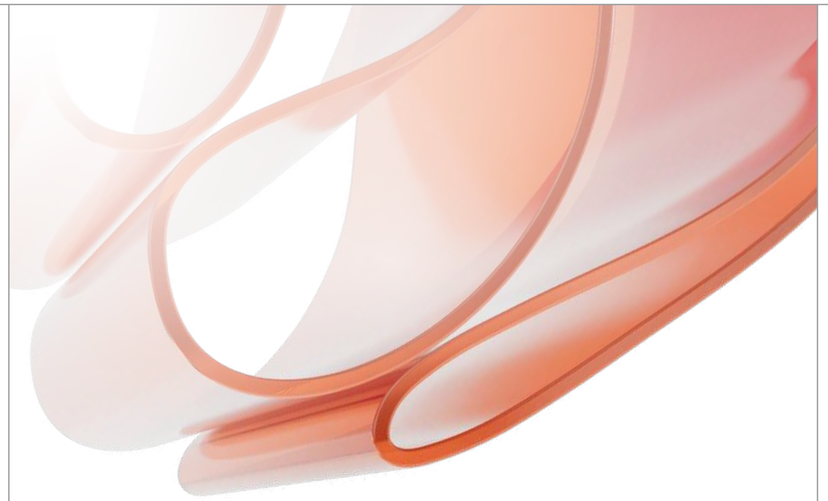
To keep code generation organized and prevent overlap, we set up four Claude Code agents: a Gherkin specialist creates scenarios, an implementation agent builds the test logic, and a log reader analyzes failures. The last one, an architect agent, acts as a design authority reviews new code for consistency with the framework.

STEP-BY-STEP PROJECT FLOW

Stage 3. Production smoke test automation

Our developers connected the framework to the GitLab pipeline and built the initial smoke suite. The automated tests now evaluate the live UBIDEX platform, reducing the need for manual smoke verification. By adding Allure reporting to the pipeline, the UBIDEX team can compare test runs over time and discover trends for recurring issues.



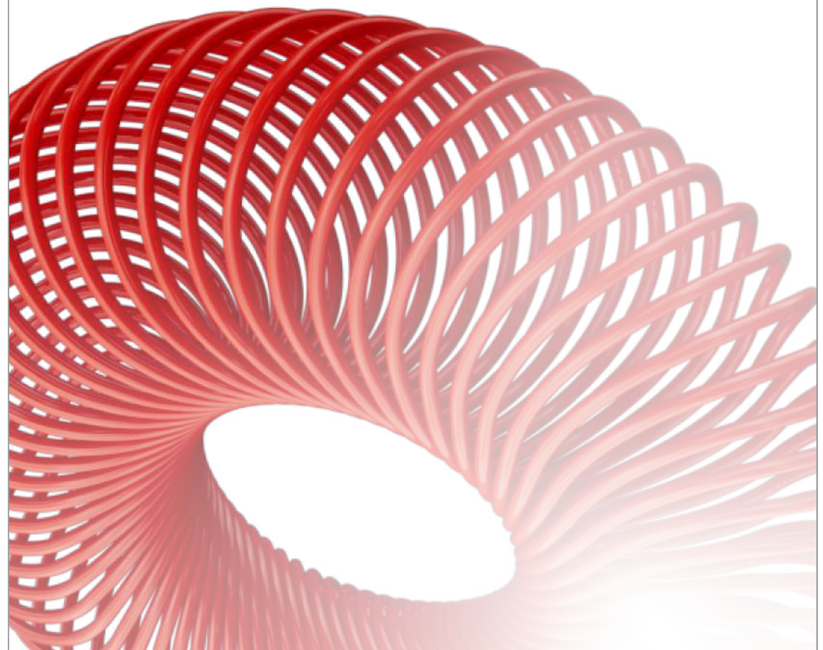


Stage 4. Claude Team rollout across departments

As the automation workflow took shape and other teams got involved, we recommended a Claude Team setup that would give engineering, QA, and product teams shared organizational context through Projects. The shared alignment let each team member access the same working context and use Claude more effectively for CI/CD design, support triage, and requirement refinement.

Stage 5. Regression coverage expansion

As the approach proved useful, our team is working to transfer the successful automation suite from smoke testing to broader regression coverage. In parallel, our team is gathering cycle-time and defect-escape data to measure the workflow's impact over a full measurement cycle.



DELIVERED SOLUTION & FUNCTIONALITY



- **Domain-oriented test-automation framework:** Built a Java framework that mirrors the UBIDEX business domain, divided into functional areas for test scenarios, business logic, UI/API tests, and core infrastructure. This structure separates domain logic from the technical details of how tests interact with the platform.
- **Reusable BDD steps:** Organized Cucumber/Gherkin steps based on the platform's domain logic, allowing the team to combine existing steps into different scenarios for both API and UI interactions. The team can build new scenarios far faster since they don't need to recreate the underlying test logic.
- **Production smoke test suite:** Built an initial smoke suite and connected it to the existing GitLab pipeline to automate release-related routine tasks. Smoke tests are now running automatically against the live UBIDEX platform.
- **Allure test reporting:** Integrated Allure into the GitLab environment to enable easier tracking of test results and historical trends showing test suite behavior over time and revealing recurring problems.
- **Architectural checks:** Added ArchUnit checks to enforce layer boundaries and ensure new tests are created in line with the framework structure.
- **Automatic cache sync:** Implemented runtime cache synchronization into the framework for automated sync with the platform's cache whenever data changes during a test run.

PROJECT IN NUMBERS

Early results from AI-assisted QA and engineering workflows.

Despite the recent rollout, the AI-based framework we developed is already reducing the time spent on repetitive engineering work. Faster testing and support processes help the team spend less time on routine tasks and redirect that capacity toward product development.



3x
faster framework setup and initial test coverage



40%
faster automated test development

33%
faster technical support investigations

25%
faster product task preparation

PROJECT OUTCOMES

Automated release verification

UBIDEX now runs automated smoke tests against its live platform through GitLab, reducing the time and effort QA specialists spend on smoke verification.

Foundation for automated test coverage

The new domain-oriented framework contains all the core elements needed to adapt to new test areas in a fraction of the time it would take to build from scratch or recreate the same test components for each new area.

AI-assisted workflows beyond testing

QA, engineering, and product teams joined a Claude Team environment to increase productivity in their respective domains. Shared project context helps them bring relevant product context into requirement refinement, coding, support investigation, and CI/CD-related work.



aimprosoft

contact@aimprosoft.com