

# AI-Accelerated Unit Test Migration

## OVERVIEW

An enterprise application team needed to upgrade an outdated frontend framework, but legacy test tooling created a significant migration barrier across hundreds of files. Updating the tests manually was estimated to take four to six weeks, leaving the organization exposed to security and maintenance risks associated with outdated dependencies. Source Allies used AI-assisted development to accelerate the migration and unblock the critical framework upgrade.

## APPROACH

Our team established clear reference patterns before using AI to perform repetitive code transformations at scale. Engineers remained responsible for reviewing the output and refining the instructions as new scenarios emerged.

- **Engineer-Defined Patterns:** Manually converted representative tests to establish the target architecture and modern Playwright standards.
- **Reusable AI Instructions:** Created a living Markdown instruction file defining syntax, Playwright patterns, and project standards for the coding assistant.
- **Iterative Refinement:** Updated the instructions as edge cases emerged, improving consistency across more than 350 test conversions.
- **Human Quality Control:** Shifted developer effort from repetitive refactoring to reviewing AI output and making targeted adjustments.

## Results

350+

Unit tests migrated to Playwright

80-88%

Reduction in migration time

140-220  
Hours

Estimated engineering time saved

## SOLUTION

Source Allies combined engineer-defined reference patterns with GitHub Copilot to migrate more than 350 unit tests to Playwright. A living Markdown instruction file provided consistent guidance across the codebase, while developers reviewed the output and refined the instructions as new scenarios emerged. The approach completed a large-scale migration quickly while establishing modern testing patterns and clearing the way for the Angular upgrade.