

PractiTest Unlocks Project-Aware AI for QA Teams with New MCP Capability

REHOVOT, - NONE -, ISRAEL, February 23, 2026 /EINPresswire.com/ -- [PractiTest](#) today announced a new [MCP](#) (Model Context Protocol) capability that connects AI Models like ChatGPT and Claude directly to PractiTest's project data. Teams can use real context to generate tests from requirements, suggest edge cases, analyze coverage gaps, and then create and link approved outputs back into PractiTest.



AI is only as reliable as the context it can see. MCP unlocks project-aware AI by connecting AI tools directly to PractiTest context, so outputs become consistent, grounded, and usable.”

Joel Montvelisky , PractiTest's CPO.

“AI is only as reliable as the context it can see,” said Joel Montvelisky , PractiTest's CPO. “Most QA teams are still using AI in isolation - re-explaining their project, copy-pasting artifacts data, and getting answers that do not reflect what's actually happening in their testing data. MCP unlocks project-aware AI by connecting AI tools directly to PractiTest context, so outputs become consistent, grounded, and usable.”

Why QA Teams Need Project-Aware AI

Most AI in QA breaks down for the same reason: the work lives in a test management system, while the AI lives outside of it. That gap leads to generic answers, repeated context setup, and manual back-and-forth that erodes trust.

The industry is also experimenting with fully autonomous QA agents, but most efforts stall because AI still lacks dependable context and teams still need humans in the decision loop. MCP focuses on what works now: bringing real PractiTest context into AI so teams can move faster without over-claiming autonomy.

What MCP Unlocks in PractiTest

PractiTest MCP connects your AI models to PractiTest so it can retrieve real project context and perform actions directly in PractiTest using supported MCP tools.

Examples of practical workflows teams can run with MCP include:

Write tests from requirements: Ask your AI to generate a scripted or BDD test based on a requirement, then create it in PractiTest.

Suggest edge cases and missing scenarios: Use coverage details for a requirement, have the AI identify gaps, then propose additional tests for review before creation.

Coverage gap analysis to test creation: Request [requirement coverage](#), analyze existing linked test cases and identify gaps, create missing tests, and link them back to the requirement to improve traceability and completeness.

Cross-tool orchestration: Use PractiTest's MCP together with tools like Jira to create end-to-end workflows, generating tests from requirements, syncing coverage, and keeping quality signals aligned across your entire SDLC.

"MCP turns generic AI into project-aware AI," said Joel Montvelisky, PractiTest's Chief Product Officer. "When AI can see structured testing context, it stops guessing. And when it can push approved work back into PractiTest, teams move from ideas to execution without inefficient manual handoffs."

Noa Segol
PractiTest
86376997398

[email us here](#)

Visit us on social media:

[LinkedIn](#)

[YouTube](#)

[X](#)

This press release can be viewed online at: <https://www.einpresswire.com/article/894484936>

EIN Presswire's priority is source transparency. We do not allow opaque clients, and our editors try to be careful about weeding out false and misleading content. As a user, if you see something we have missed, please do bring it to our attention. Your help is welcome. EIN Presswire, Everyone's Internet News Presswire™, tries to define some of the boundaries that are reasonable in today's world. Please see our Editorial Guidelines for more information.

© 1995-2026 Newsmatics Inc. All Right Reserved.