



Success story

From instability to confidence with VirtusLab's expert IDE knowledge

The team behind Continue had already delivered a solid experience with its stable and performant VS Code plugin. But as demand grew across developer communities, Continue sought to expand its reach to JetBrains IDEs, home to a large segment of professional engineers.

However, this move introduced significant instability: the new JetBrains plugin froze and crashed IDE instances, frustrating early users and halting adoption.

Faced with unfamiliar architecture and compounding technical debt, Continue partnered with VirtusLab to bring expert-level IntelliJ development knowledge to the table. In just three months, VirtusLab stabilized the plugin, modernized its architecture, and enabled the Continue team to confidently resume independent development.





The Challenge

Continue's mission was to offer developers the same seamless AI assistance in JetBrains IDEs as in VS Code. Their JetBrains plugin, however, fell far behind its VS Code counterpart. Internally developed, the plugin suffered from:

- Frequent crashes of the entire IDE (not just the plugin).
- Freezing behavior due to heavy operations being run on UI threads.
- An unstable in-IDE sidebar interface using an embedded browser JCEF.
- Legacy testing framework (Robot) was cumbersome due to repetitive custom coding.
- Compatibility with LLM plugins was weak due to non-standard API usage.

Without internal IntelliJ expertise or proper documentation, Continue struggled to make progress. Despite the product's strong potential, the team was at risk of losing developer trust on a major IDE platform.



The solution

VirtusLab stepped in as a mentor and technical partner, quickly diagnosing architectural weaknesses and stabilizing the plugin from the inside out. The engagement focused on five key areas:

- **Stability improvements**
 - Resolved crashes and freezes by moving heavy operations off the Event Dispatch Thread (EDT).
 - Ensured correct threading practices aligned with JetBrains architecture standards.
 - Introduced error submitters that allowed users to opt in and attach thread dumps during crashes, significantly improving diagnostics.
- **API modernization**
 - Replaced legacy implementations with the new JetBrains autocompletion API designed for LLM integration.
 - Improved compatibility with other plugins by aligning with updated interfaces.
- **Testing overhaul**
 - Removed the brittle Robot testing framework.
 - Introduced JetBrains' new end-to-end test API, eliminating numerous lines of legacy code.
 - Identified and removed test-only dependencies (e.g., JUnit, Hamcrest, Apache Commons) that were mistakenly bundled in public release artifacts, some containing known CVEs.
 - Reduced dependency bloat from dozens to just two production dependencies, all while avoiding bundling the Kotlin SDK.
- **Observability enhancements**
 - Integrated Sentry for real-time exception tracking.



- **Codebase consistency**
 - Refactored fragmented components, migrating the code from EDT to BGT and establishing a messaging funnel.
 - Introduced clearer separation between UI and background tasks.
- **Knowledge transfer**
 - Documented architecture, debugging techniques, and development workflows.
 - Conducted collaborative sync meetings to upskill Continue's team.



The results

With VirtusLab's guidance, Continue transformed their IntelliJ plugin from an unstable prototype into a maintainable, modern foundation to improve their product internally.

Key outcomes included:

- Crashes and freezes resolved; the plugin became significantly more stable
- ≈ 1000 lines of legacy test code removed
- Modern JetBrains APIs adopted — enabled LLM integration and better plugin compatibility
- Observability integrated — Sentry added for exception tracking; PostHog usage upgraded
- User-driven debugging enabled — users could opt-in to submit thread dumps during crashes
- Build and dependency hygiene improved — test-only libraries removed from production builds
- Security risks mitigated — CVE-prone dependencies eliminated
- Dependency footprint reduced — from dozens to just two runtime dependencies (Sentry + PostHog)
- Improved maintainability — cleaner architecture, easier debugging, and reduced technical debt
- Knowledge transfer completed — internal team ready to maintain and evolve the plugin independently

Although the JetBrains Marketplace rating remained low during the short 3-month engagement, Continue now has the technical foundation and capability to regain developer trust and steadily improve the product experience.



The tech stack

/ Languages



/ JetBrains APIs



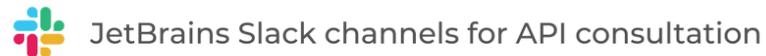
/ Testing



/ Tooling



/ Collaboration



About VirtusLab

At VirtusLab, we aim to lead in software technology, working consistently to enhance efficiency. Our profound commitment to research and development and a dedicated focus on emerging trends and inspirations fuels an innovative culture. This ethos precisely guides advancing our cutting-edge solutions, inviting collaboration to expand the boundaries of software technology collectively. We welcome you to be a part of this transformative journey.

[Let's connect](#)

Contact Details

info@virtuslab.com

POLAND

Kraków Headquarters

Virtus Lab Sp. z o.o.
ul. Szlak 49
31-153 Kraków

GERMANY

Berlin Office

+49 30 52014256
VirtusLab GmbH
Potsdamer Platz 10
10785 Berlin

UNITED KINGDOM

London Office

+44 (0)20 4577 1051
Virtuslab Ltd.
40 Bank Street HQ3
London E14 5NR