NOVACOMP

SUCCESS STORY

The right use of AI in the SDLC: development & reengineering

Modification of 10 SQLRPGLE programs completed in 7 hours with AI-First development

An IBM i project estimated at 10 days was completed in 7 hours by adopting the AI-First Development paradigm, meeting a critical commitment with time to spare.

Industry: Technology / Software Development

*Image for illustrative purposes only

Summary

An AS400 pass had been returned by the Software Engineering Lead (JIS) on 2 occasions due to bad development practices, inadequate standards, and architecture issues. The accumulated delay was 2 months. The development lead committed to delivering within a maximum of 5 days. The traditional estimate to modify the 10 SQLRPGLE programs was at least 10 days. By applying AI-First Development, the team completed all changes in 7 hours.

Main challenges

- Pass returned twice by the JIS due to bad practices and inadequate standards.
- Accumulated development delay of 2 months.
- Delivery commitment of maximum 5 days with a real estimate of 10 days.
- 10 SQLRPGLE programs requiring modification and correction.
- Observations from the first return had not been properly addressed in the second.

Approach

The AI-First Development paradigm was adopted, where artificial intelligence actively participated at every stage — not as a reactive assistant, but as the central axis of the process:

AI-First analysis of the 10 programs to understand context and issues.	**Collaborative code generation with AI.**
Automated syntax validation with AI.	**Intelligent refactoring to correct architecture and standards.**

Results

- 10 programs modified, corrected, and delivered in record time.
- Time reduced from an estimated 10 days to 7 real hours (93% reduction).
- Client commitment met within the 5-day deadline.

Technologies used

IBM i / AS400

SQLRPGLE

GitHub Copilot (AI-First)

Conclusion

The AI-First paradigm transforms productivity when AI actively participates throughout the entire development cycle. The difference is not in having access to the tools — the team already had GitHub Copilot — but in using them as the central axis of each stage, not as occasional support.

Built on expertise. Driven by people.