

Your Next Teammate Has Arrived:

Our Story & Reflections on Building AI Coding Assistant for Clinical Insights Generation

Vincent Shen, Roche, Mississauga, Canada

Introduction: A New Paradigm in Clinical Data Science

In the world of clinical data science, we are no longer just writing code; we are racing against time to deliver life-saving medicines to patients. Every efficiency gain in how we process clinical data directly impacts our ability to generate insights that change lives. With that in mind, Roche embarked on a journey in 2023 to redefine our relationship with technology.

With AI emerging as a revolutionary technology for the past few years, we have also moved beyond viewing AI as just a simple tool. We believe we are entering the era of the "Next Teammate" for statistical programmers and analysts in traditional biometrics teams — a collaborative, intelligent partner that scales with our ambitions. In this paper, we are going to share our story of how we evolved from basic adoption of AI tools to a sophisticated, multi-agent ecosystem that acts as a true peer in the clinical insights generation process.

The Catalyst: Roche's Bold Move into Open-Source and Cloud-Based Computing Environment

As we migrated clinical studies from the legacy *entimICE* system to the modern cloud-based *OCEAN* platform, we faced a monumental shift in our tech stack. While *OCEAN* brought powerful capabilities—including Git-based workflows, GitLab integration, and a suite of specialized R packages for SDTM, ADaM, and TLG creation—it also introduced significant friction:

- **Significant Resource Requirements:** The "human toll" of change management was high. Supporting thousands of users through a platform shift required immense resources and a dedicated help desk infrastructure.
- **Steep Learning Curves:** Even with robust documentation and training, the jump to modern software engineering practices (like Git) and specific clinical R packages created a bottleneck for many seasoned data scientists.

To bridge this gap, we realized that documentation wasn't enough. We needed an assistant that lived where the work happened. In the next few sections of this paper, we will focus on discussing one of our many efforts to address these frictions: building our own GenAI coding assistant.

Five Phases of Evolution for Our Own AI Coding Assistant

The development of our AI coding assistant was not a linear path, but a series of aggressive technological pivots designed to keep pace with the fastest-moving sector in tech history.

We began with an initial Feasibility Test in August 2023. This phase utilized the GPT-4 Code Interpreter to determine if a large language model (LLM) could successfully ingest selected function manuals, established TLG (Tables, Listings, and Graphs) code templates, and variable dictionaries to accurately edit code for common clinical programming tasks.

Building upon this proof of concept, the project transitioned into developing NExpert 1.0 from September 2023 to March 2024. Leveraging the GPT-4 API in conjunction with LangChain, this iteration established a more structured workflow. The core focus was to classify user topics and generate accurate, context-aware responses by performing vector-searched queries across extensive R package documentation. This foundational structure then led to the more significant NExpert 2.0 phase (April 2024 – July 2024), which represented a major "pivot" in user experience. This phase involved fine-tuning the Llama 3 model with curated materials, and provided a first glimpse on how we may integrate an AI coding assistant into the user's primary workspace like RStudio.

The project's architecture advanced further with the launch of AIR (AI for R programming), developed between December 2024 and July 2025. This phase marked a move towards a "collaborative", multi-agent-based system, built on LangGraph and the Gemini 2.5 model, and integrated directly into the Positron IDE. It provided a chatbot interface with integrated webR extension, where users could test the AI-suggested code within the chatbot. AIR was also able to be connected with established coding assistant tools like Github Copilot and Claude Code via MCP (Model Context Protocol).

The current phase, starting in December 2025 and ongoing, focuses on Claude Code "Skills." This strategic move aims to leverage the best offerings from established coding assistant tools like Claude Code, and embed Roche-specific coding templates and standard as "skills" to boost the performance of AI code suggestions.

Strategic Reflections

As we look at the relationship between development effort and user satisfaction, we uncovered a critical strategic insight:

1. Public GPT-4: Low effort, but impossible to provide access to all users.
2. NExpert 1.0 (LangChain): A massive spike in development effort led to positive results - this is the first time our internal users see and feel how AI may help their work.
3. NExpert 2.0 (Fine-tuning): Significantly more development effort, with improved user satisfaction.
4. AIR (LangGraph + MCP): Development work is manageable with all the advancement in the AI field, and the product launch is most successful, specifically with integration with IDEs.

5. Claude Code "Skills": The breakthrough in Sustainability. By moving away from building bespoke chatbots and instead injecting our logic ("Skills") into established coding tools, we achieved "mind-blown" satisfaction levels while reducing the engineering burden.

The shift to GenAI can also be compared to our earlier transition to R and Open-Source. However, the velocity and organizational impact are vastly different:

The Open-Source Transformation, primarily through the adoption of R for clinical insights generation, represents an industry-internal shift with a relatively long gestation period. Roche initiated the journey 9 years ago, with the first {teal} (interactive analysis dashboard built on RShiny) developed as a proof-of-concept. Over the years, the R packages evolved a lot, but foundationally remained relatively stable (with regular updates). The open-source transformation overall required considerable but manageable adoption efforts, but also resulted in a significant gain in efficiency.

In contrast, the GenAI Coding Assistant Transformation is characterized by its outside-of-industry originality, and evolves significantly faster due to the advancement of LLMs at a record pace. It has been just 2.5 years since our first feasibility test with the GPT4 Code Interpreter feature, and 2.5 years later, the GPT4 model itself has already been seen as outdated. This transformation presents manageable adoption efforts for users due to its user-friendliness and intuitive design. However, it brings considerable challenges for IT, legal and quality teams, where "the answer" is generated probabilistically rather than deterministically. The efficiency gain of these AI coding tools is unprecedented and there is a huge potential to be realized in the coming years.

The Future: From Helpdesk to Data Science Partner

We view the role of AI through a three-stage evolutionary lens, moving from Passive to Innovative and finally Revolutionary states:

1. The Helpdesk (Passive): Focused on the basics, where human programmers still do most of the work, and only turn to AI if there is a need for debugging, code explanation, or basic training of new tools. Most organizations stop here.
2. The Service Provider (Innovative): AI coding assistants are asked to do some of the routine work by human programmers, but with human programmers as final approvers. AI handles code suggestion, optimization, and execution (with human approval), which accelerates the turnaround delivery time significantly. AIR, Github Copilot, and Claude Code are all examples of this, and some companies are integrating these tools into their routine workflows.
3. The Data Science Partner (Revolutionary): This is our North Star. In this stage, the AI understands the business problem, understands clinical data, and generates actionable insights. It is no longer just helping you code in a specific language/package; it is also helping human programmers in areas they may not have expertise in.

Conclusion

The journey of building an AI coding assistant at Roche has taught us that while leveraging AI as a "Helpdesk" is a great first step, it is not the destination. The true revolutionary potential of GenAI lies in moving beyond simple syntax help and toward a partnership that broadens the very scope of what a data scientist can explore. If you are still treating AI as a glorified helpdesk, you are probably missing the opportunity of the decade. It is time to start leveraging AI coding assistants to tap into the core of clinical insights generation, and only then can we truly accelerate the delivery of the next generation of clinical breakthroughs.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Author Name: Vincent Shen

Company: Roche

Email: vincent.shen.vs1@roche.com