

# The CIO's Guide to Using ChatGPT to Migrate an Application

If you are a CIO — and haven't been asked this already — this question is coming your way:

“Have we tried GenAI to migrate a legacy application?”

by [Darryl Worsham](#), GAP's General Manager — Modernization Services  
Posted on [Mobilize.Net](#) on April 28, 2024



"Have we tried generative AI to modernize our legacy code?" It's the question every CIO will face sooner or later. Just last week, a CIO from one of our clients reached out to our team at Growth Acceleration Partners after getting that exact question from their CEO. They had experimented with ChatGPT for code migration, and let's just say, it was a bit more challenging than anticipated.

My team recently migrated a legacy VB.NET Windows Forms application to a modernized C#/Blazor web application. They experimented with the idea of using ChatGPT 4.0 to simplify and streamline the process. It wasn't a seamless journey, but they learned a lot along the way! Here's a breakdown of their experience, potential benefits of using ChatGPT, and essential tips for successful legacy application modernization.

## The Experiment: ChatGPT as Migration Copilot

The .NET 4.8/Entity Framework 6.4.4 VB.NET desktop app was the perfect candidate for a modern makeover. The team aimed to transform it into a .NET 8/Entity Framework Core 8.0.2 C#/Blazor Server web app. ChatGPT appeared to offer a way to streamline the process through its code generation and natural language processing capabilities.

## What Worked: ChatGPT's Strengths

ChatGPT demonstrated its potential in the following areas:

- **Efficient code migration:** It effectively translated VB.NET code snippets into corresponding C# code. This streamlined the process, especially for the business logic layer.
- **Explainer tool:** ChatGPT often explained its rationale and code suggestions, which helped the team learn new approaches and .NET best practices.
- **Adaptability:** It handled various target platforms including Node.js/Angular—it wasn't just C# and Blazor.
- **Database model support:** ChatGPT helped them build database models effectively from DDL.
- **Debugging assistance:** While not its primary strength, it still offered some troubleshooting tips.

## The Caveats: Where ChatGPT Needs Attention

Generative AI is impressive but needs refinement for a smooth code migration. Here's why:

- **Context matters:** To get the best results, the team had to repeatedly provide precise context about the project, .NET version, and target framework details. ChatGPT has limited ability to “remember” project-wide details.

- **No big picture:** You can't hand the entire application to ChatGPT and expect a migrated solution. It can handle snippets, but you'll still need to architect the solution yourself.
- **Manual intervention is key:** The generated code often needed adjustments or a steer in the right direction, especially regarding compatibility with the latest .NET versions and Blazor practices.
- **Inconsistent results:** The team got different results with code output, and technical recommendations - even when they used the exact same prompt.

## Recommendations for Successful Migration with ChatGPT

1. **Feasibility first:** Not every application is a good candidate for this approach. Assess the complexity and your team's readiness before committing.
2. **Plan, plan, plan:** Start with a detailed migration plan. Break the process into manageable phases.
3. **Know your target:** Invest time in mastering the intricacies of your target technology (in our case, Blazor and .NET 8).
4. **Refine, don't copy blindly:** ChatGPT gets you started. You'll need to refine, debug, and adapt the generated code to ensure quality.
5. **Teamwork is key:** Collaborate closely with human developers who can offer the experience and judgment that AI currently lacks.
6. **Document everything:** Thoroughly document every step! Include the learnings, prompts you used, and challenges faced. It'll be invaluable for future projects.



## Final Thoughts

Generative AI, much like the advent of the mobile phone, stands as a transformative force in technology. In our journey, ChatGPT shows potential to simplify and accelerate the legacy application migration process. While it certainly isn't a magic wand, it serves as a powerful assistant that can augment human expertise and strategic planning. Embracing these tools is akin to embracing any major technological revolution: it offers an opportunity to redefine our approaches and achieve results that were previously unattainable.

As we look to the future, the possibilities are vast. Imagine a world where AI doesn't just assist but collaborates, enabling us to push the boundaries of what we can achieve in software development and beyond. If this glimpse into AI-assisted migration has sparked your interest, let me know — I'm eager to dive deeper into specific code examples and prompts we used to turn old into gold. Join us in exploring how we can leverage these groundbreaking tools to not just keep up with the times, but to lead the charge in innovation.

Want to learn more details about this experiment?

We have a much more detailed blog post here:

[We Migrated a Legacy App with ChatGPT: Here's What Happened](#)

**Darryl Worsham, general manager of GAP's Modernization Business Unit, is a seasoned C-suite executive with 30+ years of experience in product lifecycle management, sales, marketing, engineering and software delivery.**

### ABOUT GROWTH ACCELERATION PARTNERS:

At GAP, we consult, design, build and modernize revenue-generating software and data engineering solutions for clients. With modernization services and AI tools, we help businesses achieve a competitive advantage through technology.

To find out more, please visit [WeAreGAP.com](https://www.WeAreGAP.com) 



05202024

