Obsessing Over Clean Code

As a Computer Science student and aspiring software developer, I constantly strive to write “beautiful” code. This pursuit led me to “Clean Code” by Robert C. Martin and its follow-up, “Clean Architecture.” These books have profoundly influenced my approach to coding, teaching me to write clear, maintainable, and well-organized code. However, they also pushed me into obsessing over perfecting my code.

“Clean Code” emphasizes the idea that code should be easy to read and understand, promoting practices like clear naming, small functions, and clear responsibilities. “Clean Architecture” builds on these ideas at a larger scale, focusing on creating software that can grow and adapt over time. These books should be read together; only after reading both did I fully grasp their purpose.

Despite gaining incredible insights, I overlooked one crucial idea: perfect code/software is a fairytale. I became fixated on making my code “beautiful,” rather than writing functional code. This led to over-engineering, where I spent too much time seeking the ideal structure instead of solving problems. Although this affected my productivity, it taught me an important lesson: software, like us, has different needs and goals. What works for me might not work for you.

In conclusion, while “Clean Code” and “Clean Architecture” offer valuable advice for writing flexible software, it's essential to find a balance between theoretical purity and practical needs. Experience will help you determine this balance, ensuring you write effective and adaptable code without losing sight of the project's goals.