Decades of Code

A Journey in Software Engineering: 1982 — Present

Experience · Expertise · Evolution

The Origins (1982-1983)

- **Started in 1982:** Began the journey with Basic and Assembler on a Sinclair ZX 81, laying a low-level foundation for understanding how machines work.
- **First Commercial Success:** Sold my first proprietary software for **5,000 DM** in 1983.
- **Early Discipline:** These constrained environments taught me the value of efficient, optimized code long before modern frameworks existed.

Academic Foundation

Bachelor of Software Engineering

Deutsche Forschungszentrum für Künstliche Intelligenz (DFKI)

Focused on the theoretical underpinnings of computer science and AI. This rigorous academic environment instilled a research-driven approach to solving complex software problems.

Master of Software Engineering

Fraunhofer Institute for Software Engineering

Thesis: "Designing Software for Maintenance"

Deep dive into the lifecycle of software. My research prioritized long-term system viability, architecture, and maintainability—principles that guide my work to this day.

Professional Evolution (1997-2025)

Full Time Developer

Consultant

Trainer

Project Lead

Lead Architect

Head of Engineering and Architecture

Head of Software Development AI

I never stopped developing software

Professional Tech Stack

Systems

Assembler, C, C++

Embedded & Performance critical

Enterprise

Java, C#, Kotlin

Large-scale server backends

Modern/Scripting

Python, Ruby,

TS/JS

Rapid dev & Web ecosystems

Academic & Esoteric Knowledge

True engineering expertise comes from understanding different paradigms, not just syntax. I have studied and used diverse languages that shape how I think about code:

- **Functional:** Haskell, Lisp (Scheme/Common Lisp)
- **Structured/Modular:** Modula, Oberon
- **Logical/Stack:** Prolog, Forth
- **Scientific:** Fortran

Systems & Scale

Embedded Systems

Developed critical software for automotive systems and multi-media units where resource constraints and reliability are paramount.

Large Server Systems

Architected massive, scalable backends handling high throughput, ensuring uptime and data integrity for enterprise clients.

Desktop Applications

Created rich user experiences with complex state management and local processing requirements.

Engineering Philosophy

Architecture First

Code is a liability; Architecture is an asset. I prioritize clean, modular designs that decouple systems and reduce technical debt from day one.

Quality Assurance

Static and dynamic QA are not afterthoughts. They are integral parts of the pipeline, ensuring robustness through automated testing and analysis.

Design Patterns

Leveraging proven patterns to solve recurring problems, ensuring code is readable, maintainable, and familiar to other engineers.

More Than Just Code

"I have seen a lot of things in my career."

My value isn't just in writing syntax—it's in decades of solving hard problems, preventing failures, and designing systems that last.

Image Sources

https://images.stockcake.com/public/f/b/7/fb7a65d3-b777-4acd-8b45-0c45c1e6367c_large/vintage-green-code-stockcake.jpg

Source: stockcake.com

https://www.comciencia.br/wp-content/uploads/2020/09/virtualization.jpg

Source: www.comciencia.br

https://static.vecteezy.com/system/resources/previews/036/285/151/non_2x/abstract-blue-circuit-board-technology-futuristic-connection-system-digital-data-and-high-tech-technology-geometric-design-on-dark-blue-background-illustration-vector.jpg

Source: www.vecteezy.com