

Table of Content

Sections	Executive Summary	02
	When AI Learned to Code	03
	AI Solves the Easy Part: Humans Still Define the Hard Part	05
	Vaadin – A Perfect Fit For The AI-First World	07
	The UI For AI	08

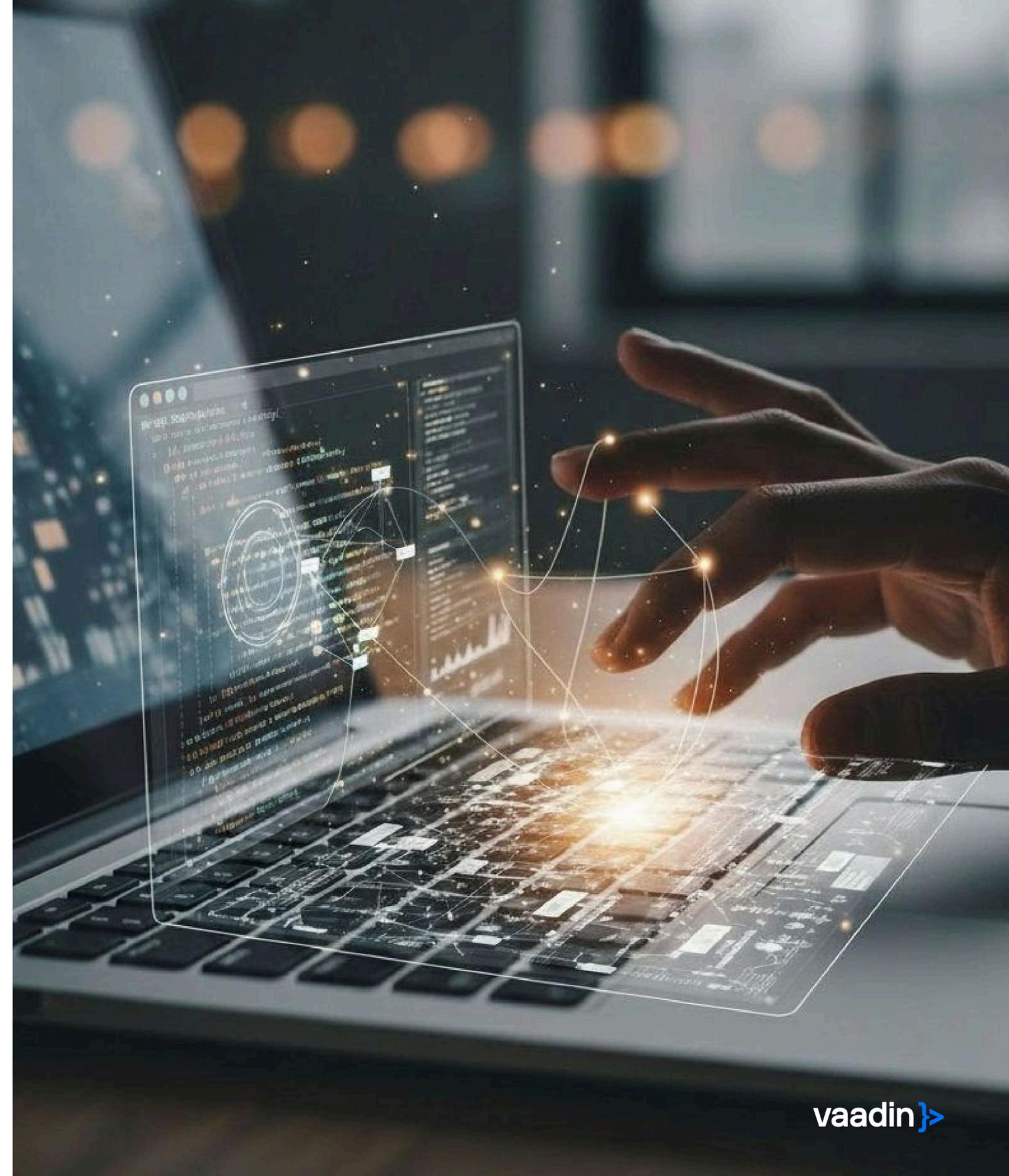
Executive Summary

Generative AI has transformed how software is written, but not why. Developers still defined intent and AI just accelerates execution.

The challenge now is translating human understanding into consistent, reliable applications. Vaadin provides the proper tooling for this AI-first world by focusing on what software should do, while automating the how.

With a strongly typed API for the user-interface layer and with abstraction of the communication layer, Vaadin makes code simpler for both humans and AI to generate, interpret, and maintain.

Vaadin acts as the “senior developer in the loop”, ensuring structure, trust, and readability even in AI-generated code. In short, Vaadin delivers clarity for humans and structure for machines, the essential foundation for building meaningful applications in the age of AI-assisted development.



AI has changed the way we build software, but not the purpose behind it. Modern coding tools can now generate entire applications from just a few lines of natural language.

When AI Learned to Code

Large Language Models (LLMs) are heuristic mathematical models. They're inexact resulting in hallucination and non-deterministic implementations. Software however is binary and deterministic. These worlds don't fit together without frameworks such as Vaadin that bridge the gap between LLM descriptions to actual implementations.

For example, a prompt might begin simply: "Build a warehouse management application." Then comes refinements: "Make it suitable for perishable goods." Next, "Ensure it follows European Regulation (EC) 853/2004, which provides specific hygiene rules for products derived from animals."

Each iteration sharpens the AI's understanding, teaching it not just what to code, but what the human means.

This process reveals a key truth: AI doesn't replace coding, but exposes what coding really is. Programming has always been a way to express human intent in a precise, structured form that both people and machines can understand. Vaadin's role in this new landscape is to provide that structure.

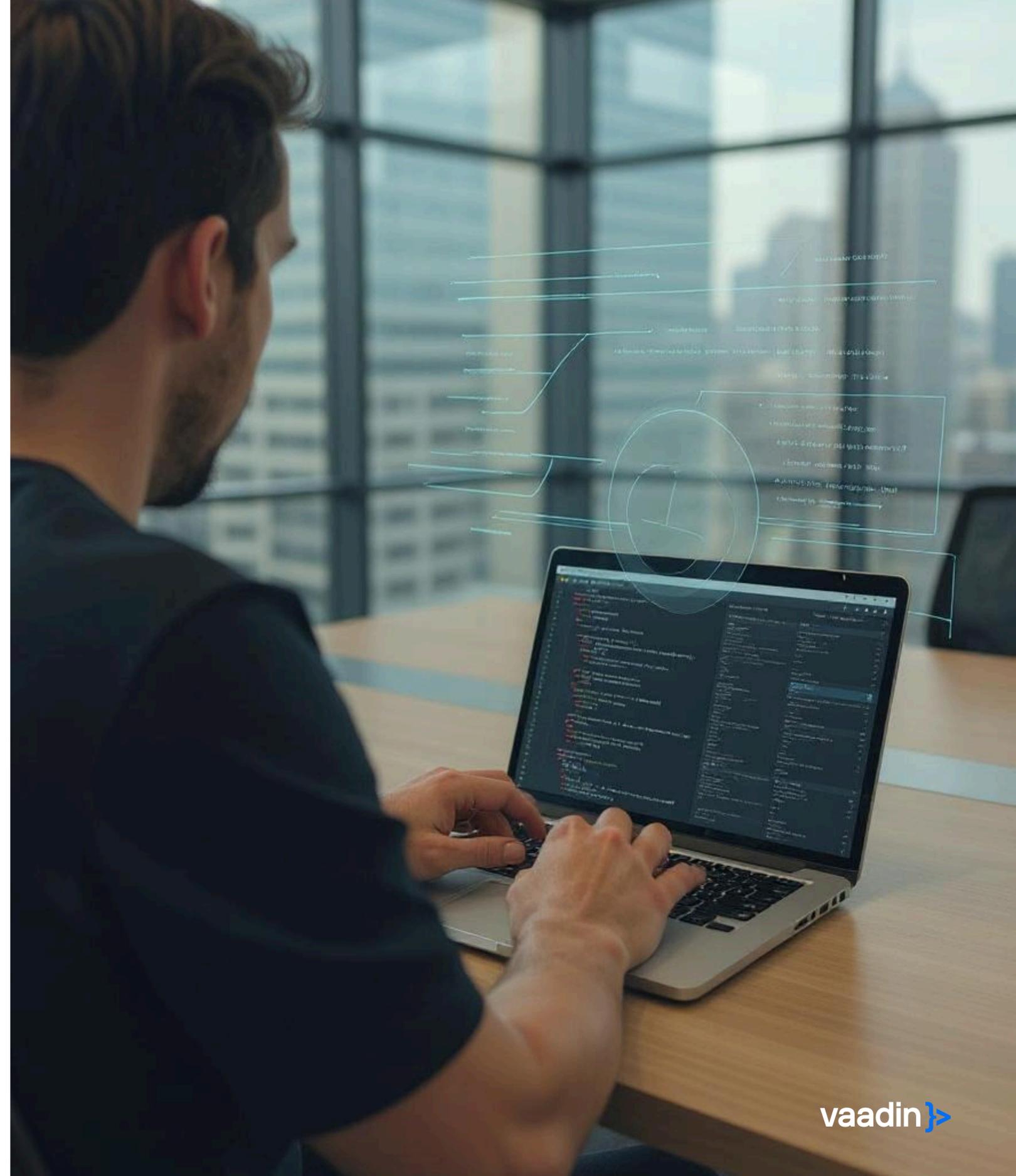
To ensure that when AI generates code, the result is not only functional, but also consistent, maintainable, and aligned with human purpose.

AI Solves the Easy Part: Humans Still Define the Hard Part

Software development involves two kinds of complexity. Essential complexity is about knowing what the application should do—its business logic, rules, and user experience. This requires human understanding and judgment.

Accidental complexity, on the other hand, is about specifying how to make those things work in code: the data bindings, state management, and UI synchronization that often clutter development.

AI can now handle much of the accidental complexity—it's excellent at producing the “how.” But without clear definition of the “what,” its results can easily become incoherent or wrong. That's where Vaadin's design philosophy comes in: by minimizing accidental complexity and letting developers (and AI) focus on intent, Vaadin ensures that applications are both simpler to build and easier to maintain.



Software development involves two kinds of complexity.

Type of Complexity	Definition	Who Handles It Best	Vaadin's Role
Essential Complexity	Understanding what the application should do — defining rules, business logic, and user experience.	Humans — they provide context, domain knowledge, and intent.	Offers clear abstractions so intent is expressed precisely and consistently.
Accidental Complexity	Implementing how things work — handling data binding, state synchronization, and UI updates.	AI (assisted by frameworks) — can automate routine implementation details.	Removes boilerplate and standardizes development and UI patterns resulting in clean code and UX.

Vaadin – A Perfect Fit For The AI-First World

Vaadin provides a grammar of intent: A clear, stable foundation where humans define goals, AI accelerates execution, and the resulting applications are predictable, maintainable, and ready for enterprise use.

Built for Clarity of Intent

Vaadin’s strongly typed APIs and components make it easy for both humans and AI systems to understand what an application should do without over-specifying how it should work.

Instant Iteration

The buildless workflow removes bundling steps, allowing rapid experimentation and immediate feedback for AI-driven development loops.

AI-Ready Abstractions

Java interfaces for the UI removes trial-and-error ceremony where the AI would need to “see” the UI and instead exposes clean, logical structures that AI coding tools can easily interpret and extend.

Predictable Design

The components ensure consistent behavior and stable application logic, even when AI systems generate or modify code. Components and compositions built at different times by different LLMs end up looking the same, with Vaadin.

**Vaadin abstracts away the how,
so both people and machines
can focus on the what that
drives business value.**

