Who Gets to Build?

Designing AI Systems Without a Technical Background

Keena Williams

Founder, Struxa AI

Executive Summary

As AI tools become increasingly conversational, the barrier to building digital systems is shifting away from technical execution and toward human judgment. Across creative, operational, and business contexts, domain experts often identify clear opportunities for improvement, yet lack the technical access to translate insight into functional tools. As a result, many ideas stall between documentation and implementation.

This case study examines an AI-assisted development approach that separates judgment from execution, enabling non-technical practitioners to design and deploy working systems without traditional engineering workflows. Rather than positioning AI as a decision-maker, the model treats AI as an execution layer operating within human-defined intent, constraints, and oversight. Responsibility remains with the builder, while AI accelerates implementation.

Two system artifacts are presented to illustrate this approach: a Podcast Operations Command Center designed to support human-led creative workflows, and truckCheck, a prototype application designed to analyze terms and conditions and privacy policies to surface potential areas of risk and concern. Together, these artifacts demonstrate how conversational development enables builders to move from insight to execution while preserving accountability.

The implications extend beyond individual tools. As AI collapses the distance between idea and execution, the quality of systems increasingly depends on the clarity of human intent embedded in their design. This shift expands who gets to build, while elevating the importance of literacy, specification, and human-in-the-loop architecture as foundational elements of responsible AI system design.

Section I: The Problem — The Technical Bottleneck Between Insight and Execution

Across creative, operational, and business contexts, domain experts regularly identify opportunities for new tools, workflows, and systems. These insights often emerge from lived experience and practical problem-solving. However, translating them into functional software has historically required formal engineering skills, access to developers, or reliance on rigid third-party platforms.

This dependency creates a persistent bottleneck. Ideas are documented, discussed, or deferred, but rarely built. The distance between recognizing a problem and implementing a solution results in lost momentum, diluted intent, or complete abandonment of otherwise viable systems.

As AI tools become more capable, this bottleneck is beginning to shift. The central question is no longer whether non-technical practitioners can build software, but how systems should be designed when execution is increasingly delegated to AI. This shift introduces new opportunities for access and speed, while also raising questions about responsibility, oversight, and decision-making in AI-assisted development.

Section II: The AI-Assisted Build Model — Separating Judgment from Execution

This case study explores an AI-assisted development model that intentionally separates human judgment from technical execution. Rather than treating AI as an autonomous decision-maker, the model positions AI systems as execution layers operating within human-defined intent, constraints, and oversight.

The approach is structured across three complementary layers:

Specification Layer (Human)
Definition of system intent, goals, success criteria, constraints, assumptions, and ethical boundaries. This layer determines what the system is allowed to do and, critically, what it is not.

Execution Layer (AI)
AI-assisted code generation, automation logic, and application scaffolding based on human specifications. In this layer, AI accelerates implementation without owning judgment or accountability.

Orchestration Layer (Human-in-the-Loop)
Ongoing review, intervention, iteration, and decision-making at key checkpoints throughout

the system lifecycle. This layer ensures alignment, correctness, and contextual awareness as systems evolve.

This structure enables non-technical builders to focus on logic, outcomes, and responsibility while preserving human authority over system behavior. The tools used to support this model are secondary to the architectural principle itself: execution may be automated, but judgment remains human-led.

Section III: System Artifacts Produced

Using the AI-assisted build model described above, this project produced a small set of functional system artifacts designed to test how non-technical practitioners can translate domain expertise into working software. Each artifact prioritizes adaptability, human oversight, and practical use over polish or commercialization.

Podcast Operations Command Center

The Podcast Operations Command Center is a modular, AI-assisted workflow designed to support the end-to-end podcast production lifecycle. The system ingests raw audio and video assets, generates transcripts, assists with editorial organization, and tracks production stages across multiple episodes.

Rather than automating creative decisions, the system is intentionally structured to keep humans in the loop at key checkpoints, including content selection, narrative shaping, and final distribution. AI is used to accelerate execution and surface options, while authorship and judgment remain human-led.

This artifact demonstrates how AI can function as an operational partner in creative workflows without displacing human decision-making or eroding creative control.

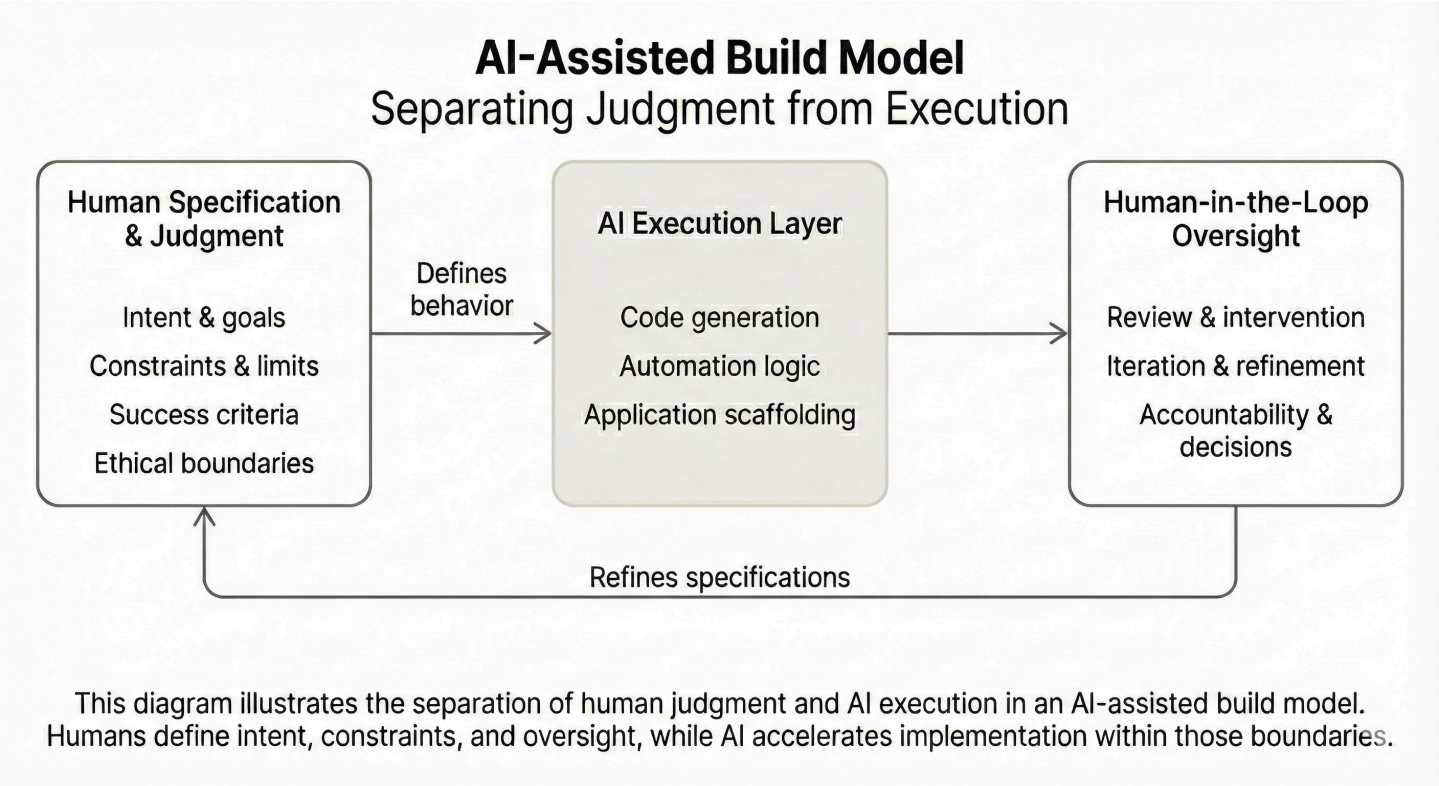
StruckCheck (Prototype Application)

StruckCheck is a lightweight prototype application developed to examine how non-technical builders can encode domain expertise into functional software systems using AI-assisted execution. The application analyzes terms and conditions and privacy policies to surface potential areas of risk, ambiguity, or concern, supporting review rather than automating judgment or decision-making.

The system was developed without traditional backend or mobile engineering workflows. Instead, system intent, behavioral constraints, and evaluation criteria were defined at the specification level, with AI handling structural implementation and application scaffolding. Human oversight remained central in determining what the system evaluates, how findings are framed, and where interpretive limits are enforced.

StruckCheck illustrates how conversational development can enable subject-matter experts to move from insight to execution while preserving human judgment, accountability, and control over system behavior.

Section IV: Implications for AI System Design



This diagram illustrates the separation of human judgment and AI execution in an AI-assisted build model. Humans define intent, constraints, and oversight, while AI accelerates implementation within those boundaries.

Taken together, these artifacts illustrate a broader shift in how AI-enabled systems are being designed and built. As development becomes increasingly conversational, technical capability is no longer defined primarily by the ability to write code, but by the ability to specify intent, define constraints, and exercise judgment across the system lifecycle.

One implication of this shift is a change in who gets to participate in building systems. AI-assisted development lowers traditional barriers, enabling domain experts and non-traditional practitioners to contribute directly to system creation. This expanded access increases the diversity of perspectives shaping AI systems, while also increasing the responsibility placed on builders to articulate goals, assumptions, and limits clearly.

A second implication is the centrality of human-in-the-loop architecture. When AI systems handle execution, the placement of human review, intervention, and decision-making becomes a foundational design choice rather than a secondary safeguard. Systems that lack clearly defined oversight risk over-automation, misalignment, or erosion of trust.

This shift also reframes governance as a design practice. Responsible system behavior increasingly emerges from how workflows are structured, how authority is distributed

between humans and machines, and how decisions are staged over time. In this sense, architecture itself becomes a primary mechanism for accountability.

Finally, increased accessibility to building does not eliminate risk; it redistributes it. As more individuals gain the ability to create AI-assisted systems, the potential for poorly specified logic, unintended consequences, and unchecked automation grows. This makes literacy in systems thinking, evaluation, and responsibility as essential as creativity or operational insight.

As AI continues to collapse the distance between idea and execution, the quality of human judgment embedded in systems will determine whether these technologies empower users or simply accelerate existing failures at scale.