

Workshop Content

Workshop Objectives

- *Establish credibility (Rakesh as 1Huddle CTO & Codewalla Founder)*
- *Showcase One Huddle as a successful AI-native transformation story*
- *Introduce Codewalla's "5 Shifts" framework as derived from real experience*
- *Subtly market Codewalla as AI-native product studio*
- *Demonstrate actual AI-native features (no gimmicks)*
- *Final CTA → **Invite serious founders to subscribe for a AI Sprint with Codewalla***

Section 1: Welcome & Problem Framing (12–15 mins)

Slide 1: Title Slide

- Time: 1 min
- Title: AI-Native Product Development Workshop
- Subtitle: From prototypes to production-grade AI products
- Presented by: Rakesh Raju — CTO, 1Huddle | Founder, Codewalla

Speaker Notes:

"Hi, I'm Rakesh. I lead tech at 1Huddle and founded Codewalla.

It's a great time to be building AI native products

But how ?

We'll share lessons, failures, tools, and how we turned ideas into reliable systems."

Slide 2: Market Reality

- Time: 2 min
- Title: Market Reality
- Content

Title: Why It's the Best Time to Build AI-Native Products

Content (bullets):

- Expanded universe of products now possible with AI.
- Unprecedented enterprise demand — no skepticism.
- Talent leverage: Tiny teams → Huge ARR.
- Faster learning cycles — tighter feedback loops.
- Fullstack AI → New moats, not just features.
- Founders who build AI-native now will be harder to disrupt.

👉 **This sets the ambition** — you want them hungry before you dive into the mechanics.

-
- Everyone's building AI prototypes, but most won't survive in production.
- Prototypes ≠ Production success
- What breaks:
 - Trust
 - UX
 - Infra
 - Process

Speaker Notes:

"Everyone's trying AI right now—hackathons, GPT wrappers, flashy demos. But very few are turning that into something users can rely on every day."

What we've seen—and what this session will unpack—is that the real failures come after the demo:

Trust breaks when AI output is inconsistent

UX breaks when you try to bolt on AI without rethinking interaction

Infra breaks under load

*And your team's process just isn't built for probabilistic systems
This workshop is about learning from those cracks and building with AI from the core."*

Slide 3: The Bar Has Moved

- Time: 2 mins
- Title: The Bar Has Moved
- Content
 - What used to be called an MVP is now seen as just another prototype.”
 - AI frameworks and tooling have improved
 - The cost of building has dropped dramatically
 - But that raises expectations
 - Founders are now expected to show:
 - Polish
 - Traction
 - AI-native thinking in UX & design

Speaker Notes:

"Let's be honest—AI has made it cheap and fast to build.

That should be a win... but it's actually raised the bar.

What used to be called an MVP? Now it's seen as just another prototype.

Investors and customers now expect polish, traction, and AI-native design much earlier in the lifecycle.

That shift is what we'll keep coming back to throughout this session."

Slide 4: Is Your AI Real or Cosmetic?

- Time: 3 mins
- Title: Is Your AI Real or Cosmetic?
- Content
 - Think about your product...
 - Do users trust AI?
 - Are the AI features actually used?
 - Can you measure their impact?
 - Whether or not you filled the workbook — take 15 seconds.
 - What's your honest answer to these questions?

Speaker Notes:

"Take a moment and think about your own product.

Whether or not you filled out the workbook, this is your chance to check in:

Are your AI features used?

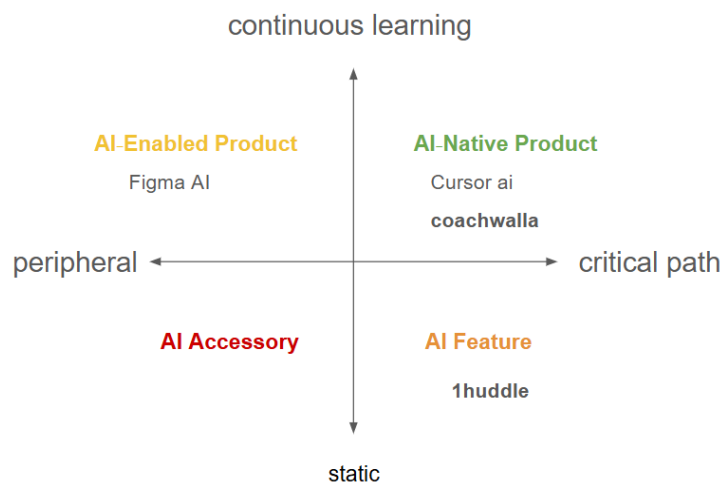
Can you trust the outputs?

Are you even measuring the right things?

This self-check will help frame what comes next."

Slide 5: What Do We Mean by AI-Native?

- Time: 3 mins
- Title: What Do We Mean by AI-Native?
- Content
 - Bring in the *expanded universe of businesses* idea.
 - AI-Native = **Products impossible without AI.**
 - **Embed a short definition:**
“AI-native products aren’t adding AI. They are built *because* AI now makes them viable.”



- Top-right = AI-Native... in theory
- But our definition adds a real filter: Business Impact
 - Engagement
 - Retention
 - Sales
- If it's not moving metrics, it's not AI-native.

Speaker Notes:

"Most people stop at the quadrant.

Yes — AI-native features live in the top-right: they're in the critical path and they evolve.

But that's not enough.

Unless they drive real outcomes — engagement, retention, sales — they're just expensive distractions.

So we use the 2x2 to spot potential, and business metrics to confirm value."

Slide 6: Why It Matters

- Time: 2 mins
- Title: Why It Matters
- Content
 - AI-native products change your numbers.
 - Higher Engagement
 - Better Retention
 - Increased Revenue
 - But only if AI is:
 - Useful
 - Trusted
 - Measurable

Speaker Notes:

"This is what it comes down to.

AI-native is not a buzzword — it's a multiplier.

But only if it checks three boxes:

It's useful — people actually use it

It's trusted — people rely on it

It's measurable — you know it's working

If those aren't true, it doesn't matter how fancy your tech is. If those aren't true, the business impact won't follow.

And here's the truth: most teams don't start there.

They start with a thin AI feature that demos well — just like we did at 1Huddle."

Section 2: From Feature to Foundation – 1Huddle (15 mins)

Slide 7: Everyone Starts Here

- Time: 1 min
- Title: Everyone Starts Here
- Content
 - Most teams start with AI as a feature.
 - It demos well
 - It impresses investors
 - But it breaks at scale
 - Our story at 1Huddle followed this exact arc.

Speaker Notes:

"You add a thin AI feature, maybe wrap a prompt around a flow, and it feels great. Early feedback is good. It demos well. Investors love it.

But once you ship it to real users... things start to fall apart. That's exactly what happened to us at 1Huddle."

Slide 8: The Problem with Gamified Training

- Time: 3 mins
- Title: The Problem with Gamified Training
- Content
 - Gamified training worked — players loved it.
 - Content creation was a bottleneck:
 - Managers struggled to create quality games
 - L&D teams took hours (sometimes full days)
 - Content velocity limited platform value
 - So in 2024, we launched “AI Assist” → Aimed to help managers generate games using prompts

Speaker Notes:

"Let me give you context before we talk about AI."

At 1Huddle, the core mechanic — gamified training — worked. Players loved it. Engagement was never the problem.

But the bottleneck was content creation.

Managers found it hard to write good games.

L&D teams said it could take up to a full day just to produce a single quiz that was both compliant and engaging.

So in 2024, we launched AI Assist to fix that.

It let managers generate game content from a few prompts.

It was fast. It was exciting. But as you'll see next — it also broke fast."

Slide 9: What Broke When We Tried to Scale

- Time: 4 mins
- Title: What Broke When We Tried to Scale
- Content
 - Process Friction
 - Traditional QA didn't hold up
 - Prompt regressions and inconsistent outputs
 - Infra Limitations
 - Latency and context loss under load
 - Model couldn't handle large inputs reliably
 - UX & Trust Gaps
 - Output quality varied too much
 - Managers started losing confidence in using it live
 - Enterprise Constraints
 - Didn't meet network isolation expectations
 - Enterprise Security audit feedback flagged maturity gaps

Speaker Notes:

"As AI Assist started hitting real usage, we ran into a few truths:

Our QA process wasn't built for AI. Outputs kept changing, and we didn't have a way to evaluate them systematically.

On the infra side, context loss, latency, and prompt size limits became real issues. We couldn't support larger game inputs — especially for big training programs.

Managers saw inconsistent results — sometimes great, sometimes way off. Trust dipped.

And from an enterprise lens, we didn't meet some security expectations, like network isolation. The feedback during audits made it clear: the system needed maturing."

Slide 10: What Changed in AI Assist v2

- Time: 4 mins
- Title: What Changed in AI Assist v2
- Content
 - Rethought from prompt to multi agentic workflow
 - Prompt orchestration
 - Dynamic system prompts
 - Context-aware suggestions
 - Guardrails and evaluation
 - Prompt regression testing
 - Output review tools
 - Trust mechanisms
 - Tracked which AI outputs were ignored vs accepted
 - Players could flag invalid questions
 - Enterprise readiness
 - Improved handling of large inputs through chunking and context management
 - Enabled secure private deployments

Speaker Notes:

"In v2, we made AI part of the product's architecture — not just bolted on top.

We moved to a multi agentic workflow — with system agents, prompts and logic that adjusted to the use case.

We added guardrails — things like regression testing and internal review tools to keep output quality consistent.

And on the trust front, we moved beyond just offering AI.

We started measuring:

Which suggestions were accepted, edited, or ignored

That gave us a real signal on what was working and what wasn't.

We also improved how the system handled longer or more complex inputs — breaking them down, managing context, and avoiding truncation — which was critical for larger enterprise use cases."

Slide 11: One Feature Led to a Complete Shift

- Time: 2 mins
- Title: One Feature Led to a Complete Shift
- Content
 - It changed how we design and architect products
 - It changed how our teams work and collaborate
 - It changed how we test, release, and learn
 - That's why we had to rethink the fundamentals — across product, process, and people.
 - We call these the Five Shifts.

Speaker Notes:

"What started as a feature completely rewired how we build.

It forced changes across:

Product — how we design and architect

People — how teams collaborate

Process — how we test, ship, and learn

That led us to define five fundamental shifts — the ones we now use to guide any AI-native build.

Let's walk through them."

Section 3: The LIFFT Framework (15–20 mins)

Slide 12: The LIFFT Framework

- Time: 3 mins
- Title: The LIFFT Framework
- Content
 - We turned our hard-earned lessons into a repeatable system.
 - LIFFT = Learning · Infrastructure · Foundation · Flow · Trust
 - It's not a methodology. It's a diagnostic lens for building AI-native products
 - This is what we wish we had when we started.
 - Tie it to **how top startups are winning**:

"The fastest-growing AI-native startups today — tiny teams, huge leverage, deep trust — are winning *because* they have these 5 shifts baked in."

Speaker Notes:

"After building AI features at 1Huddle — and later helping other teams do the same — we realized the real challenge wasn't model selection or UX polish.

It was the foundation:

How teams learn. How systems evolve. How people collaborate around AI.

We kept running into the same five patterns.

That's where LIFFT came from.

It's not a playbook or process. It's a lens — a way to assess whether your product, team, and workflows are actually AI-native, or just pretending to be.

Each letter stands for a shift we had to make."

Slide 13: Learning

- Time: 3 mins
- Title: Learning
- Content
 - Launch is not the end — it's the beginning
 - What this shift means:
 - Continuous evaluation of AI behavior in production
 - Feedback loops with both human and automated checks
 - Prompt tuning, model adjustments, and trust monitoring post-release
 - Metrics that go beyond usage: What's ignored? What's trusted?

Speaker Notes:

"When we say Learning, we don't mean dashboards and usage stats.

We mean your system is actively observing itself post-launch:

Is it behaving reliably?

Are users actually trusting what AI is suggesting?

What gets ignored?

For us at 1Huddle, this shift began when we stopped asking 'Is it working?' and started asking 'Is it learning?'

We tracked which outputs were accepted, which were flagged, and which were silently ignored. That's where real feedback starts — and real iteration begins."

Slide 14: Infrastructure

- Time: 3 mins
- Title: Infrastructure
- Content
 - Your stack needs to evolve to support AI's unpredictability.
 - What this shift means:
 - Prompt versioning and test gates
 - Trust scoring, fallback logic, and observability tools
 - Evaluation pipelines as part of CI/CD
 - You can't debug AI like you debug code. Your infra has to reflect that.

Speaker Notes:

"This shift hit us hard at 1Huddle.

We had infra built for deterministic software.

Suddenly:

Logs didn't tell the full story.

Regression testing became expensive — every model update or prompt change could break prior behavior.

QA couldn't reliably test what would happen in production

So we evolved.

We built prompt regression tests, fallback flows, and integrated monitoring tools not just for if the model runs — but how well it performs in real use.

This kind of infrastructure is what makes AI-native products reliable, explainable, and debuggable."

Slide 15: Foundation

- Time: 3 mins
- Title: Foundation
- Content
 - AI-native products don't bolt on AI — they're built around it.
 - What this shift means:
 - Product value breaks if AI is removed
 - AI behaviors are understandable, measurable, and trusted
 - Prompts, context, and feedback loops are treated as first-class product assets
 - From idea to implementation — AI must be part of the foundation.

Speaker Notes:

*"Let's be clear — 1Huddle isn't fully AI-native **YET**. But in one core area — game creation — we began treating AI as foundational.*

We shifted from:

Trying AI as a feature

→ to embedding it into a user flow where value depended on the AI working well

We designed prompts that were versioned and structured

We captured and responded to how users interacted with AI suggestions

We rebuilt parts of the product to support that interaction — not just generate content

That's what it means to treat AI as part of the foundation.

You don't just add it — you architect around it."

Slide 16: Flow

- Time: 5 mins
- Title: Flow
- Content
 - AI-native teams have to work differently.
 - What this shift means:
 - Product managers design for AI performance, not just user flows
 - Designers shape tone, confidence, and trust — not just wireframes
 - Developers orchestrate models, fallbacks, and evaluation — not just logic
 - QA owns probabilistic validation — not just pass/fail testing
 - AI tools (Cursor, Copilot) amplify strong engineers — vibe coding alone won't ship production-ready code
 - It's about rethinking what your role means.

Speaker Notes:

"This shift is about more than collaboration — it's about how each person's work changes."

Product managers used to define features and flows. Now, they shape how AI performs in the user journey.

In Coachwalla, PMs tested whether AI-generated feedback should be encouraging, neutral, or direct — and designed prompt strategies around what built trust and drove performance.

Designers are responsible for how the AI feels — not just how it looks.

One of our designers simplified how the system communicated uncertainty. For example, when a vague input led to weak quiz output, we added subtle UX nudges that guided users to refine their prompt — without making the system feel broken.

Developers moved from deterministic logic to AI orchestration — handling model selection, fallback flows, and adaptive behaviors.

In AI Assist, they built logic to chunk long inputs and stitch coherent outputs, making the AI reliable inside a real workflow.

QA transitioned from pass/fail testing to evaluating trust.

We used prompt regression tests and user interaction telemetry to identify when AI outputs degraded or caused frustration — long before it became a support issue.

Vibe coding ≠ engineering.

Cursor or Copilot can draft code fast, but shipping to production still needs engineers who

reason about edge cases, security, and maintainability. AI tools accelerate the skilled; they don't replace the skill.

That shift in how each role works — and works together — is what we call Flow"

Slide 17: Trust

- Time: 3 mins
- Title: Trust
- Content
 - Users don't trust AI by default — trust has to be designed.
 - What this shift means:
 - Help users understand what AI can and can't do
 - Design for uncertainty, recovery, and graceful failure
 - Make AI output editable and interruptible
 - Track trust signals (ignored outputs, rewrites, flags) — not just usage
 - Proactively address bias, protect user data, and explain how AI decisions are made

Speaker Notes:

*"Users don't trust AI by default — at least not yet.
And until they do, you have to design for that gap."*

That means:

Giving users control — editable outputs, interruptible flows

Helping them recover when the AI's wrong

And most importantly: watching what users do, not just what they say

In 1Huddle, we saw this when managers started using AI Assist less.

Not because it broke — but because they couldn't predict when it would get things wrong, or why. That hesitation was enough to hurt adoption.

So we started tracking which outputs were ignored, edited, or flagged.

That became our real trust metric — not usage, but confidence in using it again.

We've seen that users also disengage if outputs feel biased, opaque, or privacy-invasive.

So part of this shift is baking in explainability, bias checks, and strong privacy defaults"

Section 4: Coachwalla: Applying the Shifts from Day 1 (7 mins)

Slide 18: Why Coachwalla Is Different?

- Time: 4 mins
- Title: Coachwalla: AI-Native from Day One
- Content
 - Coachwalla was built with LIFFT from day one.
 - Learning: log every AI suggestion and track trust scores
 - Infrastructure: use CI gates and prompt regression to manage rollout
 - Foundation: remove the agent, the coaching flow breaks
 - Flow: Team reviews data together weekly
 - Trust: editable suggestions, fallback logic, no surprises

Speaker Notes:

"Coachwalla was built with LIFFT from day one."

Learning

Every AI suggestion logs whether it's accepted, edited, or ignored.

We track this in Looker using agent trust scores — and regularly act on it.

Infrastructure

Agents and Prompts are versioned in Git. We run regression tests in CI using promptfoo.

We use LaunchDarkly to roll out prompt changes gradually — 5%, then 25%, then full rollout — based on stability and user behavior.

Foundation

The AI agent is essential — it creates and schedules coaching tasks.

If you take it out, the core coaching workflow fails.

Flow

The pod — PM, designer, developer — meets every Friday to review AI metrics.

Trust

Every AI output is editable and interruptible.

Outcome

This approach pays off in a few ways:

We ship usable product increments weekly, not monthly

We've avoided major rebuilds — nothing's been thrown away and rewritten

Our pod is small, but output is high — because feedback loops are tight, and AI behavior is predictable

Most importantly, tech debt hasn't piled up — because we built with AI-native complexity in mind from the start"

Slide 19: Impact of Early AI-Native Thinking

- Time: 3 mins
- Title: Coachwalla: Impact of Early AI-Native Thinking
- Content
 - Impact of designing with LIFFT from the start:
 - Faster iterations, fewer rebuilds
 - Lean team, higher output
 - Tech debt stays flat
 - Users trust and adopt AI without hand-holding

Speaker Notes:

"This is what we've seen by applying LIFFT early :

Faster iterations: *Because learning and trust feedback loops are baked in, we don't stall after launch. We improve mid-flight.*

Fewer rebuilds: *We're not ripping out AI features every few months. The architecture expects change.*

Lean team, high output: *One pod can do what used to take two or three teams — because prompt, UX, and infra decisions are made together.*

Flat tech debt: *Prompt versioning, fallback logic, and regression testing aren't retrofits — they're default.*

Users adopt without hand-holding: Because outputs are editable, interruptible, and designed for clarity, people trust the system without needing a manual."

Section 5: Why Founders Should Care (15 mins)

Slide 20: Building Leverage Early

- Time: 3 mins
- Title: Coachwalla: Building Leverage Early
- Content
 - AI-native isn't about adding AI — it's about multiplying product impact.
 - Founders who adopt early:
 - Learn faster, iterate smarter
 - Ship more with smaller teams
 - Build trust into the product from day one
 - Avoid costly rebuilds and silent adoption failure
 - Add **explicit tie to new economic model**:
"In a world of Machine Money (AI-driven abundance) and Human Money (authenticity, creativity), AI-native products are the bridge — delivering real leverage."

*(This injects the deeper podcast idea of **dual economy** — Machine Money vs Human Money.)*

Speaker Notes:

"AI-native isn't about adding AI — it's about multiplying product impact.

That's why it matters now, especially for founders.

Learn faster

You're not waiting for lagging metrics like churn to tell you something's wrong.

When you track trust scores, prompt edits, and AI usage patterns from day one, you get fast feedback and meaningful insight.

Ship more with smaller teams

With prompt orchestration, fallback logic, and automated evaluation in place, a lean team ships faster — with fewer regressions.

That's how we built Coachwalla: one cross-functional pod delivering every week.

Build trust from day one

Trust isn't a setting. It's built through editability, recoverability, and clear fallback behavior.

Users adopt faster when they understand and control what the AI is doing.

Avoid costly rebuilds

We've seen it again and again: AI features that demo well, but fail in production — leading to rewrites months later.

Founders who think AI-native early avoid that trap. They don't just ship faster — they ship smarter"

Slide 21: Common Misconceptions About Building with AI

- Time: 5 mins
- Title: Common Misconceptions About Building with AI
- Content
 - What we've seen leaders get wrong:
 - "AI lets me hire less-skilled developers"
 - "Prompting replaces engineering"
 - "If it works in the demo, it'll work at scale"
 - "Trust is a UI problem"
 - "We'll fix data and evaluation later"
 - "Vibe coding = shippable product"

Speaker Notes:

"Let's pause here — because a lot of smart teams get stuck not on what AI can do, but on how they approach it.

Here are a few traps we've seen often:

"AI lets me hire less-skilled developers"

In reality, AI tools like Copilot amplify good engineers — they don't replace them. You still need people who can reason about systems, edge cases, and feedback loops.

"Prompting replaces engineering"

Prompting is part of the interface — not the architecture. If you're not building structured fallbacks, context windows, or evaluation gates, your product won't scale — no matter how clever the prompt.

"If it works in the demo, it'll work at scale"

A prototype is easy. Scaling means thinking about drift, observability, latency, privacy — and user trust. That's a product and infra problem, not a prompt problem.

“Trust is a UI problem”

We’ve seen teams try to fix trust with a tooltip. It doesn’t work.

Trust comes from how the system behaves — fallback logic, editability, recoverability.

“We’ll fix data and evaluation later”

You won’t. If your learning loop isn’t wired in early, you’ll be flying blind after launch. And once you lose trust, users rarely come back.

“Vibe coding = shippable product”

This is the most seductive trap. Tools like Cursor or Copilot make it feel like you’re done — but vibe coding skips versioning, testing, fallback logic, and long-term maintainability.

It’s exploration, not production. You still need engineers who can turn fast ideas into durable systems.”

Slide 22: The AI-Native Product Formula

- Time: 3 mins
- Title: The AI-Native Product Formula
- Content
 - $\text{Product Impact} = (\text{Validated Learning Rate} \times \text{Useful Idea Volume}) \div \text{Total Cost of Execution}$

Speaker Notes:

"This is the formula we've come to rely on when evaluating product direction and team structure:

Let's break it down:

Validated Learning Rate: How quickly can your team test and confirm what actually works? Not assumptions — validated insights from usage, trust metrics, edits, and real user behavior.

Useful Idea Volume: How many promising ideas can your team generate and put into play? This includes prompting, variations, user journeys — the surface area of innovation.

Total Cost of Execution: Not just engineering cost, but time, team size, rework, coordination overhead — everything that slows you down.

This formula gives founders a mental model for where the leverage really is. It's not "Can we ship AI?" — it's "Can we increase learning and output, while keeping costs low?"

Slide 23: The New Competitor Is AI

- Time: 2 mins
- Title: The New Competitor Is AI
- Content
 - “Instead of selling AI tools to incumbents, we’re funding companies that will replace them.” — Y Combinator
 - Fullstack AI is not a layer — it’s the new moat
 - Founders who build AI-native from day one are harder to disrupt
 - “Startups that are AI-native from day one move faster, learn faster, trust faster — and become the new incumbents.”

Speaker Notes:

"This quote sums up where the market is heading — and how the most aggressive early-stage investors are thinking:

"Instead of selling AI tools to incumbents, we’re funding companies that will replace them." — Y Combinator

*In this model, **AI is the moat** — not a feature. And founders who start this way from day one? They’re not disruptable. They’re the ones doing the disrupting."*

Section 6: Outro & Call to Action (10 mins)

Slide 24: AI-Native Isn't a Feature

- Time: 3 mins
- Title: AI-Native Isn't a Feature
- Content
 - AI-native is a strategy shift.
 - It changes how you design, ship, and learn
 - It reshapes team structure and tooling
 - It impacts product velocity, cost, and trust
 - It's no longer optional — it's foundational

Speaker Notes:

"We'll wrap with what we believe is the core truth behind everything we've shared today:

AI-native isn't a feature. It's a strategy shift.

It changes how you design — from UI flows to prompt logic.

It changes how you ship — with learning loops, fallbacks, and observability built in.

It changes how you learn — because metrics, not assumptions, shape iteration.

It also reshapes teams. We've seen PMs write prompts. Designers build for user confidence. Devs maintain prompt pipelines.

The structure has to evolve with the product.

And all of this directly affects what founders care about:

How fast can you learn?

How much can a small team build?

How often do you rebuild vs. evolve?

This isn't optional anymore — it's the new default for products that want to last"

Slide 25: What We Got Wrong (Before We Got It Right)

- Time: 2 mins
- Title: What We Got Wrong (Before We Got It Right)
- Content
 - What We Got Wrong (Before We Got It Right)
 - AI features are easy to demo — hard to keep useful
 - Retrofitting trust, learning, and evaluation is expensive
 - We've rebuilt more AI than we've shipped

Speaker Notes:

"None of this came from theory.

We've made most of the mistakes before we got it right.

We've shipped AI features that looked great in the demo — and failed quietly post-launch.

We've added trust logic, fallback rules, and observability after users disengaged — when it was harder to recover.

And we've rebuilt systems that were never meant to learn or adapt — because they weren't designed AI-native from day one.

These lessons shaped everything we've shown today.

If you want to avoid those same rebuilds, we can help"

Slide 26: The AI Sprint

- Time: 4 mins
- Title: The AI Sprint
- Content
 - Is your AI product stuck between prototype and production?
 - Users don't trust it
 - UX breaks under pressure
 - Outputs feel brittle
 - You're far from production-grade
 - In 2–5 days, we'll help you:
 - Apply the LIFFT Framework
 - Identify blockers
 - Fix what matters
 - Ship smarter

Speaker Notes:

"We built the AI Sprint for one reason: because prototypes are slick. They're exciting. But they hide the real gaps.

This is where most founders get stuck. Not because their idea is wrong — but because they didn't shift the process.

So in just 2 to 5 days, we work with you and your team to:

Apply the LIFFT framework to your real product

Run a strategic audit

Deliver a focused fix plan and roadmap"

Slide 27: Where You Go From Here

- Time: 2 mins
- Title: The AI Sprint
- Content
 - If you're building something AI-native — or want to — we'd love to hear from you.
 - Sprint page / contact us QR code
 - **What Will You Build That Was Never Possible Before?**

Content:

- AI-native is not just efficiency — it's invention.
- What's a product that couldn't exist before AI?
- What's a problem that's solvable now?
- Your unfair advantage: build what couldn't be built before.

(This leaves them inspired, not just informed.)

Speaker Notes:

"That's where we'll pause.

If you're building something in this space — or thinking about it — we'd love to hear what you're wrestling with.

Thanks for your time — and we're happy to take questions."