

# AI in Vertical Software

## Vertical Software

What changes for the vertical software company as AI enters its industry

*“Your codebase is not the moat. Your domain knowledge is. The question is whether it is institutional or personal.”*

01

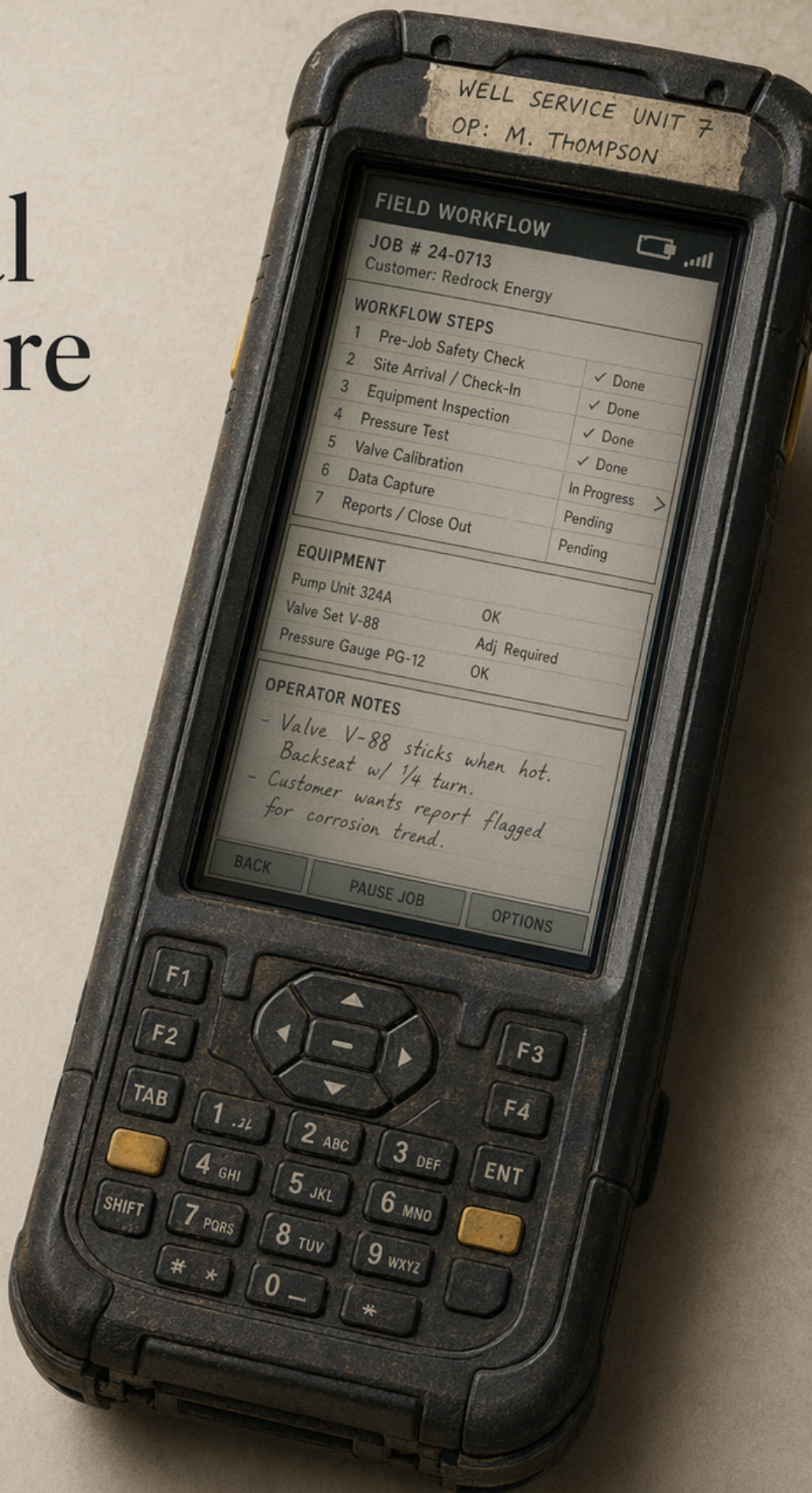
What is your competitive win rate on new logos?

02

What are you selling that AI-native competitors cannot build?

03

What fraction of your moat survives a founding team transition?



# What changes for the vertical software company as AI enters its industry

---

Vertical software built its defensibility on three pillars: deep domain knowledge encoded in product, long integration timelines that punished switching, and workflow embeddedness in customer operations. AI has collapsed the first two. The third is weakening as AI-native entrants invest in migration tooling that used to be uneconomical. The moat that felt permanent three years ago is now a ticking clock.

What remains defensible is the accumulated understanding of the domain itself: why the data model is structured this way, what the regulator actually enforces versus what they publish, what customers need that they have never articulated. That knowledge is the real moat. It is also in the heads of the founding team and a handful of senior employees. When they leave for AI-native opportunities, the product development velocity drops and the competitive response capability degrades. The asset that matters most is the one most likely to walk out the door.

For the CEO of a vertical software company, this is not a product question. It is a revenue question, a profit question, and a valuation question. Revenue, because AI-native entrants reach feature parity in twelve to eighteen months and the competitive win rate on new logos is already shifting. Profit, because a feature race against competitors with no legacy architecture compresses margin from both sides. Valuation, because the ARR multiple investors will pay depends entirely on how they price the moat, and personal knowledge in three heads earns a discount while institutional knowledge encoded into infrastructure earns a premium.

The strategic move is not to outship the entrants. That race is unwinnable. It is to encode the domain knowledge before the founders leave, turn every customer interaction into training data that compounds the moat, and reprice the equity story around institutional advantage, not feature count. The companies that move first do not just protect what they have. They rebuild the moat on foundations AI cannot erode.

*This briefing is about what that move looks like, and how a management team gets it done before the next AI-native entrant raises a funded round.*

# The moat just inverted. What was defensible is now a ticking clock.

---

Vertical software was built on an assumption about how long it takes to reach feature parity. Five years. Maybe more if the domain was complex enough. That five-year runway was the defensible asset. It let incumbents accumulate customer relationships, refine workflows, and build switching costs faster than any entrant could catch up. The moat was time.

Foundation models have ingested enough domain knowledge that an AI-native entrant can now reach feature competitiveness in twelve to eighteen months. Modern AI infrastructure slashes the integration complexity that used to anchor customers. Migration tooling that was previously uneconomical is now routine. The technical barrier dropped by roughly eighty percent, and the drop is not finished. Every six months, a new generation of models makes the gap narrower.

*Your codebase is not the moat. Your domain knowledge is. The question is whether it is institutional or personal.*

What remains after the technical moat collapses is the accumulated understanding of the domain itself. Why the data model works this way rather than the obvious alternative. What the regulator actually cares about versus the text of the regulation. The workarounds customers use that reveal what they actually need. The edge cases that look like bugs but are load-bearing logic. This knowledge is the real moat, and the people who hold it are the people AI-native competitors are actively recruiting. The incumbent's task is to convert it from personal to institutional before the window closes.

## THREE QUESTIONS

# Revenue. Profit. Valuation.

*Under the old model these three questions had three different answers. AI ties them together. The same repositioning moves all three at the same time.*

### 01 · REVENUE

#### **What is your competitive win rate on new logos?**

AI-native entrants reach feature parity in twelve to eighteen months instead of five years. Every new logo you compete for, they are in the room. Every renewal cycle, your enterprise customers are reviewing alternatives with better architecture and lower price. The competitive win rate is already shifting. The renewal cohorts will show it before the ARR data does.

### 02 · PROFIT

#### **What are you selling that AI-native competitors cannot build?**

A feature race against AI-native competitors with no legacy architecture is unwinnable. They ship faster because they have no legacy. Pricing power erodes. Gross margin compresses from both sides. The Rule of 40 math gets worse every quarter a feature-race strategy continues. The profit question is whether you are defending an unwinnable hill.

### 03 · VALUATION

#### **What fraction of your moat survives a founding team transition?**

The ARR multiple investors will pay depends entirely on how they price your moat. Personal domain knowledge in three heads is a discount. Institutional domain knowledge encoded into workflow, data model, and customer intelligence infrastructure is a premium. Same ARR, very different multiples. Same company, two different equity values.

# Two columns. Every function in the company sorts itself between them.

The pattern is consistent across every function in a vertical software company. Standard feature implementation, testing, documentation, and boilerplate code are all compressing toward commodity. The work that appreciates is the work that depends on knowing the domain, the customer, and the workflow at a depth a generic platform cannot match. The further a function sits from the domain, the more pressure it is under.

DEPRECIATING

Standard feature implementation and code generation

Documentation, testing, basic QA

Onboarding flows and usage analytics

Standard support and health monitoring

Demo preparation and competitive analysis

APPRECIATING

Why the data model works this way, not another way

Regulatory interpretation the text does not cover

Customer workflow edge cases and unexpressed needs

Pricing creativity and relationship trust

Domain judgment that survives a founding team transition

Whichever column the management team trains the organization into over the next three years is the company it becomes. The decision is not whether AI affects product development. It is which side of the line the leadership deliberately moves the organization toward.

# Two paths. Same technology. Opposite outcomes.

The financial logic of the choice in front of the management team is unforgiving in either direction. The same AI technology, deployed into the same company, produces opposite outcomes depending on what the company chooses to defend.

## THE DEFAULT PATH

### Compete on features

Add AI badges. Ship faster. Try to out-execute entrants whose only legacy is the first version of the product.

### Treat domain knowledge as implicit

Tribal knowledge stays in founders' heads. Three people carry ten years of context. Every retirement is a capability hit.

### Let customer interactions disappear

CS, support, and services generate deep knowledge daily. None of it gets captured. It walks out with the team.

### Defend the current ARR multiple

Stay in the same feature-race basket every vertical software company is in. Multiple stays anchored to feature count. Risk priced in quietly.

## THE REPOSITIONING PATH

### Compete on domain depth

Make the domain knowledge the product. Encode it into workflows, data model, and customer intelligence. Let the AI features be commodity; let the domain be the moat.

### Institutionalize the domain knowledge

Encode the reasoning: why the data model works, what the regulator enforces, what customers need but never say. Make it infrastructure before the people who know it leave.

### Turn customers into training data

Every customer interaction is domain signal. Structure it, encode it, use it for your own domain-specific AI. The intelligence compounds and becomes unreplicable.

### Earn a new valuation multiple

The same ARR earned through institutional domain intelligence is worth more on the multiple. Repositioning changes what investors price.

*Same technology. Opposite outcomes. The difference is the repositioning decision, not the AI decision.*

# The growth levers that compound when used together

---

Repositioning is not a single decision. It is four levers, each available to the company right now, that compound when pulled together. Used in isolation, any one produces a modest gain. Used together, they rebuild the moat on foundations AI cannot erode.

1

## **Encode domain knowledge before founders leave**

Not code documentation. The reasoning. Why the data model works this way. What the regulator actually enforces versus what they publish. What customers need that they have never articulated. The failure modes that look like bugs but are load-bearing. Make it institutional. This is the lever that converts personal moat into permanent moat.

2

## **Turn every customer interaction into training data**

Customer success, support, professional services, sales calls, even the Slack channel where the CS team discusses edge cases. Every interaction is domain signal. Structure it, encode it, use it as training data for your own domain-specific AI. The moat compounds with every customer. The entrant cannot catch up because they do not have the interaction history.

3

## **Use AI to accelerate your roadmap faster than entrants can build depth**

The incumbent's advantage is ten years of customer interactions, regulatory edge cases, and workflow understanding. Use AI internally to ship from that base faster than entrants can build equivalent depth from scratch. The entrant starts with a better architecture but zero domain understanding. The incumbent with the data wins if they can actually leverage it.

4

## **Reprice the equity story around institutional moat, not feature count**

Investors reprice on the moat story, not the feature list. A vertical software company whose equity story is "ten years of domain knowledge, institutionalized, encoded into customer intelligence infrastructure" earns a meaningfully different multiple than one whose story is "we ship faster than entrants". The reprice is available the moment the story changes. The board deck is the artifact.

# Business first. Technology second.

GRAIL is an AI-native management consultancy. We help European mid-market leadership teams figure out the one or two things that fundamentally change about their business as AI enters their industry, and then build the capability to act on it before competitors do. For vertical software specifically, we are not a development shop (that is Netlight's territory) and we are not a strategy firm (that is McKinsey's territory). We sit in the gap between the two. GRAIL's founder spent four years as CPO of a vertical SaaS company, which means we have the conversation you actually need from the inside of the product organization, not from a deck.

*The market offers two things: vendor lock-in dressed as transformation, and consulting reports dressed as strategy. GRAIL delivers a third option: working AI capability that you own.*

## WHAT YOU GET

- Senior advisory direct to the CEO and product leadership
- Working AI capability the company owns from day one
- Custom agents built around your product organization
- Outcomes in weeks, not quarters
- No lock-in at any phase of the engagement

## WHY IT MATTERS

- Founded by operators, not technologists
- SaaS CPO experience, not slideware
- Context architecture, change management, product workflow design
- Results before you commit to scale
- Each next step earned, never assumed

# Three sessions with your management team. No IT project. Working tools from week one.

*GRAIL starts with the management team because organizational transformation cannot outrun the leadership team's personal understanding of what is possible. Three workshops. Each delivers standalone value. Each earns the right to the next.*

## WORKSHOP 01

### How to think

The strategic foundation. Why AI-native entrants are fundamentally different from the old horizontal-platform threat. The moat inversion thesis. Competitive intelligence on who is building in your vertical. Your leadership team leaves with a shared frame for the rest of the work, and the first working agent built live in the room.

## WORKSHOP 02

### Your AI team

A half-day activation. Each member of the management team receives a personal suite of five to ten custom agents. Competitive tracker. Customer intelligence synthesizer. Regulatory change monitor. Product roadmap prioritizer. Churn predictor. Working tools from week one, not slides about tools.

## WORKSHOP 03

### Competitive edge

Refinement from real usage. Agents upgraded based on how the team actually used them. The knowledge encoding roadmap. The pricing power assessment. The recommendation for what comes next, earned by what just shipped.

*Three weeks. No IT project. No board approval needed to start. The sharpest strategic conversation your management team will have this year, backed by working tools they are already using.*

## What to ask in your next board call

---

The questions where the management team cannot agree on an answer are the ones worth a longer conversation. Ask these honestly. The point is not the polished response. The point is whether the answer would survive a new AI-native entrant raising a funded round next quarter.

*If your two most senior product people left tomorrow for an AI-native competitor, how long before a competitor could replicate what they carry in their heads?*

*What fraction of your ARR is protected by institutional domain knowledge encoded into product versus tribal knowledge held by a handful of employees?*

*If an AI-native entrant raises a funded round targeting your vertical next quarter, what is the first thing you do, and does the leadership team agree on the answer?*

*Which customer interactions generate the deepest domain signal, and how much of that signal is captured as training data versus disappearing with the CS team's notebooks?*

*When the next board meeting asks about the moat, is the answer a feature list or an institutional advantage that would survive a founding team transition?*

*If any of these questions land somewhere uncomfortable, that is the conversation worth having.*

IN THIS SERIES

# There's a version of this briefing for every seat at the table.

GRAIL has produced industry briefings calibrated to the specific reality of each C-suite role.

**CEO**

▶ YOU ARE READING

**Revenue. Margin. Valuation.**

Three numbers your board reads. AI is rewriting all three in your industry.

**CFO**

**Capital. Forecast. Story.**

What you allocate this year is the story your board reads next year.

**COO**

**Cheaper. Better. Faster.**

The triple threat that used to be a tradeoff is now table stakes.

**CHRO**

**Capacity. Capability. Promise.**

The biggest talent transition of a career. Compete on capacity, not headcount.

**CPTO**

**Moat. Speed. Allocation.**

The platform moat that survives 2028 is being chosen this year.

**CCO**

**Margin. Pace. Position.**

The whole GTM playbook is being rewritten in real time, across every commercial team at once.

Visit [grail.works](https://grail.works) for the full series.

**GRAIL**

An AI-native management consultancy helping European mid-market leadership teams figure out what fundamentally changes about their business as AI enters their industry.

[www.grail.works](https://www.grail.works)