T & C

The Rendering Gap Is Your Competitive Advantage

95% of your backlog is invalid. 80% of development effort corrupts the outcome. We built the system that fixes both — AI agents architected by a 20-year outlier methodology.

THE REAL PROBLEM

Software doesn't fail because of bad code. It fails because of corrupted rendering.

Organizations perform at one level while actually operating at another. They use Orange vocabulary — "data-driven," "agile," "strategic" — while making decisions from fear, ego, and authority-dependence. The result is predictable.

60-90%
Consulting Projects Fail
Designed for performed capability, not actual capability

80%
PM Activity Is Busywork
Meetings, reporting, planning that never narrows latent space

95%
Backlogs Are Invalid
Catch-up processing masquerading as real work

41-87%
AI Deployments Fail
Bolting AI onto dysfunctional structures

THE CORE INSIGHT

The Inverted 80/20

The universally observed 80/20 distribution isn't a curiosity — it's a rendering resolution phenomenon. And the 80% doesn't just fail to produce. **It actively corrupts the output.**

The 20% — Clean Renderers

Intention collapses to precise output with minimal distortion. Clear observer state. No buffer overload. No ego-protection filtering.

These are the people who see the problem in 2 minutes while everyone else is still scheduling the meeting to discuss scheduling the meeting.

→

The 80% — Catch-Up Processing

Extensive activities that don't narrow latent space. Meetings, architecture debates, status reports, code reviews that protect reputations rather than solve problems.

In nerd-wound organizations, this isn't inefficiency — it's *technical performance theater.*

THE SOLUTION

Outlier-Architected AI Agents Close the Rendering Gap

Historically, outliers who could render cleanly were forced to translate through human layers — each translation degrading the signal. Multi-agent systems eliminate that degradation. But only when architected from the right foundations.

5 + AI = 500

5 right humans + AI agents = the output of 500 traditional employees

HOW WE WORK

The See-It Cycle — Applied to Everything

Where do you want to go? Where are you? How do you get there? What works? Every stage follows this loop. AI handles execution. Humans hold intention and make decisions.

01	**Intention Setting**	Define the real problem — not the projection. Challenge backlog validity before optimizing.	HUMAN
02	**Product Analysis**	AI agents analyze capabilities, estimate effort, plan execution — grounded in data, not guesses.	AI AGENTS
03	**Commitment Gate**	Not a checkbox. A Cycle of Choice — timeboxed commitment with Promise and Stretch targets.	HUMAN
04	**Architecture + Build**	Technical research concurrent with product analysis. Coding agents implement, test, and ship.	AI AGENTS
05	**Reality Check**	Mid-sprint: on track? Post-sprint: what rendered cleanly? What corrupted? Drag Factor calibration.	BOTH
06	**Delivery + Learning**	Under-promise, over-deliver. Every cycle feeds the next. Empirical, not theoretical.	BOTH

WHAT MAKES THIS DIFFERENT

We Operate at Every Level of the Stack

Most consultancies optimize at the UI level. We operate from the source code up — intention, field state, pattern recognition, and then action. Not more action. Right action.

SOURCE CODE LEVEL

Intention + Methodology

20 years of battle-tested project methodology. The foundations aren't heuristics — they're operating principles derived from thousands of engagements. Toyota, Disney, NFL, Riot Games, Splunk, ConsenSys.

RENDERING ENGINE LEVEL

AI Agents That Reason

Not generic AI bolted onto dysfunction. Agents that reason from foundations — breaking projections, verifying empirically, applying Pareto judgment. They don't just execute. They think.

COMPONENT LEVEL

Full SDLC Automation

Planning, architecture, estimation, implementation, testing, deployment — every stage automated with human gates at critical decision points. Real CI/CD. Real tests. Real code review.

UI LEVEL

Visible, Verifiable Output

GitHub PRs, test results, deployment logs, sprint reports. Full audit trail. Complete transparency. No black boxes. You see everything.

RESULTS

Efficacy First, Then Efficiency

150 → 35

People Needed

75% headcount reduction, same or better output

$2M → $700K

Monthly Cost

65% cost reduction with improved quality

24/7

Always Shipping

AI agents don't sleep, don't burn out, don't quit

100%

Audit Trail

Every decision tracked. Full transparency.

*Success metric: **"Boredom-inducing trust"** — stakeholders disengage because delivery is so predictable they don't need to watch.*

WHO THIS IS FOR

Two Paths, One System

The system serves two very different audiences. Both get the same quality of output — but the conversation is different.

For the Architect

You already see the whole system. You need execution infrastructure that renders at your resolution.

- → Multiply your vision across 10 projects simultaneously
- → AI agents that reason from your methodology
- → Full control over architecture and decisions
- → Your judgment amplified, not replaced

For the Operator

You need software built fast, built right, and built affordably. No drama.

- → Describe what you need. Get working software.
- → Clear timelines. Clear costs. No surprises.
- → Approve the plan. Review the output. Ship.
- → Certainty from a system that just works.

Stop Rendering Noise.
Start Shipping Signal.

We don't add more people to your problem. We replace the catch-up processing with AI agents that render cleanly — architected by methodology, governed by humans, measured by results.

Let's Talk →

T&C — Outlier-Architected AI Development