WildRaven
STRATEGIC ADVISORY

Gerard J.R. Keenan
Senior Advisor
keenan@iinc.tv

Technical Due Diligence Checklist

For AI & technology investments — a working framework for investors, PE firms, and acquirers who need to know what is real before the check clears.

> **Before you write the check, know what's real, what's smoke, and what it will actually cost to make it work.** This is the checklist I run in the field. Use it two ways: as a gate before an LOI, and as a scorecard during confirmatory diligence. A “no” is not automatically a kill — but every “no” has a price, and your job is to know that price before, not after.

1. Architecture Assessment

You are testing one thing: can this system be understood, and does it rest on solid ground? A company that cannot cleanly diagram its own architecture usually cannot control it either — and what a team cannot explain, it cannot reliably scale, secure, or hand over post-close.

Checkpoint	Why it matters / what a “no” means
A current architecture diagram exists and matches reality	No diagram, or a stale one, signals tribal knowledge and key-person risk.
The architecture can be explained to a non-technical stakeholder	If it can't be explained simply, it usually isn't understood deeply.
No single points of failure in critical paths	One fragile component can take the whole business down — and it is rarely priced in.
Disaster recovery and business continuity are documented and tested	“We have backups” is not a plan until someone has restored from them.
Security architecture reviewed within the last 12 months	Unreviewed systems hide the liabilities that surface right after you own them.
Data architecture supports current and planned scale	Data models are the hardest thing to change later and the most expensive to get wrong.

2. AI / ML Capability Maturity

This is where valuations inflate. The question is not “do they use AI” — everyone claims to. The question is whether the AI is a real, defensible capability in production or a thin wrapper around a public API that any competitor can replicate in a weekend.

Checkpoint	Why it matters / what a “no” means
Models are in production, not just prototypes and POCs	A demo proves nothing about reliability under real load and real edge cases.
Model performance metrics are defined and tracked over time	No metrics means no one actually knows if the model works — or is degrading.
Training-data pipeline is automated and reproducible	Manual, one-off pipelines break the moment the person who built them leaves.
A retraining / evaluation harness exists and is documented	Without it, model quality silently drifts and edge cases go undetected.
Clear line between AI-solvable and conventional-engineering problems	Teams that “AI everything” burn capital solving with ML what a database could do.
The AI creates a genuine moat — not a commodity API wrapper	If the edge is someone else's API, the moat belongs to that someone else.

3. Vendor & Dependency Risk

Every external dependency is both a cost line and a leverage point held by someone other than you. The diligence question is: how much of this company's fate is controlled by third parties it does not own and cannot negotiate with?

Checkpoint	Why it matters / what a “no” means
All third-party dependencies are catalogued with their cost	Uncounted dependencies are unpriced risk sitting inside your model.
No critical single-vendor dependency without a migration path	A provider's price hike or shutdown becomes your existential event.
Vendor contracts reviewed for lock-in and price-escalation clauses	Lock-in clauses quietly transfer future pricing power to the vendor.
Open-source licenses audited for compliance	A single copyleft violation can force code disclosure or block a sale.
Key API dependencies have documented fallback strategies	No fallback means one upstream outage equals your outage.

4. Team & Talent Assessment

You are not just buying code; you are buying the people who can evolve it. Technology transfers cleanly only when the team that understands it stays and is capable. Key-person risk is the most under-priced risk in technical diligence.

Checkpoint	Why it matters / what a “no” means
Key-person dependencies are identified and mitigated	If one departure can stall the roadmap, you are buying a flight risk.
The team can independently maintain and evolve the system	A team that only knows the vendor's playbook cannot own the product.
Engineering culture supports production-quality delivery	Demo culture ships slides; production culture ships reliability.
Technical leadership can make strategic, not just tactical, calls	Absent senior judgment, architecture drifts toward whoever shouts loudest.
A hiring pipeline exists for the critical roles	Scarce skills with no pipeline cap the company's growth rate.

5. Scalability & Technical Debt

Growth is the stress test that turns invisible shortcuts into visible outages. Technical debt is not inherently bad — unquantified technical debt is. You need it named, sized, and priced so it can go into your model instead of ambushing you in year two.

Checkpoint	Why it matters / what a “no” means
The system handles 10x current load without a re-architecture	If growth requires a rebuild, your thesis has a hidden capital call in it.
Technical debt is catalogued with an estimated remediation cost	Unquantified debt is a blank check you are signing on the target's behalf.
Code-quality metrics exist and are trending in the right direction	Deteriorating quality predicts rising cost-to-change and slowing delivery.
Test coverage is adequate on the critical paths	Low coverage means every change risks breaking something you cannot see.
The deployment pipeline is automated and reliable	Manual, heroic deploys do not survive scale or team turnover.

Red Flags That Kill Deals

Any single item below warrants a hard pause. Two or more clustered together usually means the story you were sold and the system you are buying are not the same thing.

- ⚠ **The vendor controls all access to production systems.** You cannot verify what you cannot see — and neither can the founders.

- ⚠ **No production deployment after 12+ months.** The demo-to-production gap has not been crossed, and it is the hardest gap there is.
- ⚠ **“AI” is actually rule-based logic or basic API calls.** The valuation premium is priced on a capability that does not exist.
- ⚠ **Key technical talent has no equity and is a flight risk.** The knowledge can walk out the door the week after close.
- ⚠ **The architecture cannot be independently verified.** Every other claim then rests on trust rather than evidence.
- ⚠ **Total cost of ownership is unknown or clearly understated.** Your returns model is built on a number the company itself cannot produce.

How to Read Your Results

Most “no” answers are not deal-killers — they are negotiating leverage and post-close work items with a dollar figure attached. The failure mode is not finding a problem; it is finding it after the wire has gone out. Run this checklist early, price every gap, and you convert technical uncertainty into deal terms. When the stakes justify it, this is precisely what a WildRaven Technical Reality Assessment™ delivers — the full forensic version of this checklist, with evidence and a Red / Yellow / Green verdict, in seven days.

Ready to Get the Truth About Your Technology?

Every engagement starts with a complimentary 15-minute discovery call — no pitch, no obligation.

keenan@iinc.tv · 310.570.0774

Schedule your discovery call →