R6AI

Playbook No. 01

A field guide for business owners and leaders

The AI Culture *Playbook*

Why 95% of AI projects fail — and it's *not* the technology.

By **Michael Dogger**
R6 AI · r6ai.com.au

Tested on our own P&L
before we ever shared it with a client.

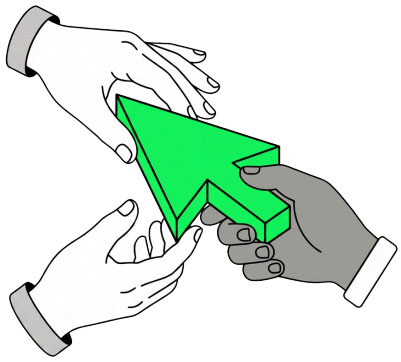
Contents

Inside this *playbook*

01	The problem nobody wants to talk about	03
02	AI amplifies what's already there	04
03	The Judgment Matrix	05
04	War stories	06
05	The Champion Model	09
06	The 90/10 rule	11
07	Bringing people on the journey	12
08	Your AI culture checklist	13

THE ONE IDEA TO TAKE AWAY

AI doesn't replace judgment.
It amplifies it.



01 The Problem

The problem nobody wants to *talk about*

Here's the uncomfortable truth about AI in business: the technology works. It's been working for a while now.

95%

of AI pilots deliver no measurable return on investment.

MIT Media Lab, 2025

70%

of AI transformation is a **people** problem.

BCG, 2025

Not a data problem. Not a model problem. A *people* problem.

Every week, another business buys an AI tool, gives it to their team, and waits for magic. Six months later, they've got a subscription nobody uses, a handful of people generating mediocre content at volume, and leadership wondering what went wrong.

What went wrong is they treated AI like a software rollout when it's actually a *culture change.*

02 AI Amplifies What's Already There

AI amplifies what's *already there*

This is the single most important idea in this entire playbook, and it's one most people get wrong:

AI doesn't replace judgment. *It amplifies it.*

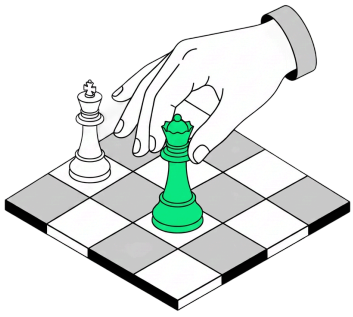
Good judgment + AI

Give AI to someone with good judgment, and you get a force multiplier. They produce better work, faster, with fewer resources. They know when the output is good and when it's rubbish. They iterate. They improve. They build things that actually create value.

Poor judgment + AI

Give AI to someone with poor judgment, and you get the opposite. They produce *more* work, faster — but it's worse. They can't tell good output from bad. They ship things that shouldn't be shipped. They burn money on API costs building things that don't work.

They become what Dan Hockenmaier brilliantly calls **Slop Cannons** — people firing off high volumes of low-quality AI-generated output with no filter.

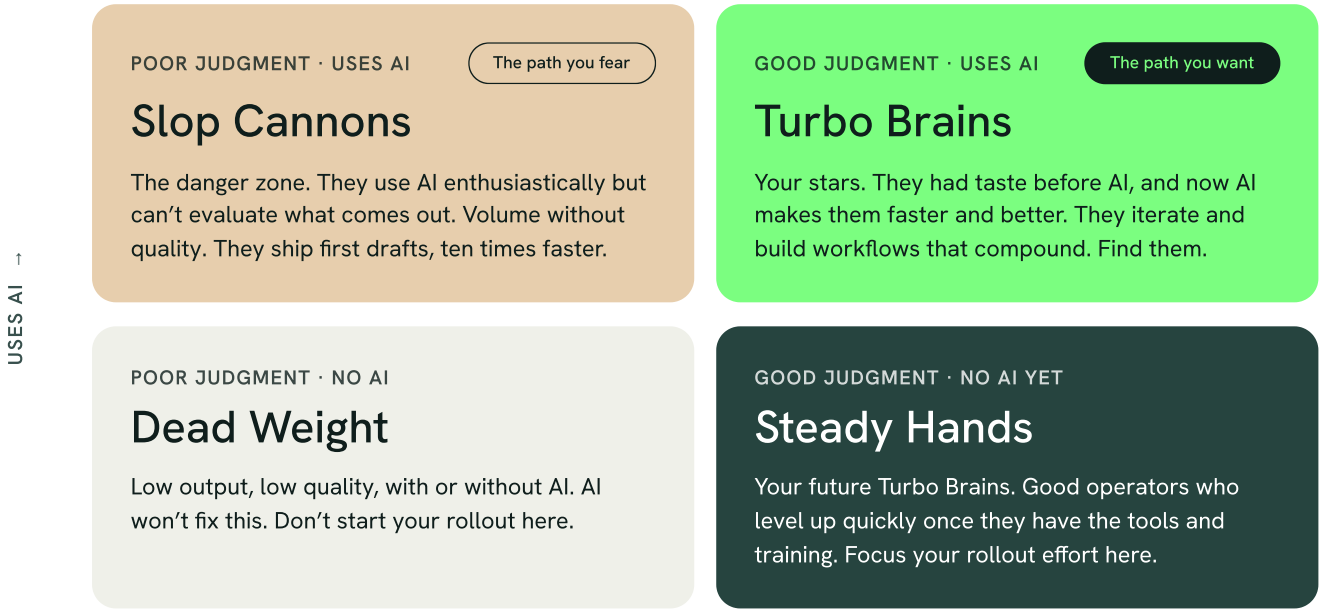


This isn't theoretical.
We've lived it.

03 The Judgment Matrix

The Judgment *Matrix*

Dan Hockenmaier created a simple 2×2 that captures this perfectly. It maps people on two axes: whether they use AI, and whether they have good judgment.



THE PATH YOU WANT

Steady Hands → Turbo Brains
Good judgment, add AI.

THE PATH YOU FEAR

Dead Weight → Slop Cannons
Poor judgment, add AI.

This matrix should change how you think about every AI initiative in your business. The question isn't *"who should get AI access?"* The question is **"who has the judgment to use it well?"**

04 War Stories

What happens when you *get this wrong*

These aren't hypothetical scenarios. These are things that happened to us, and things that happened publicly to businesses that should have known better.

From inside R6

The $2,000 coding disaster

One of our team members — smart, motivated, not a coder — decided to build a software project using AI coding tools. They had zero programming experience and no technical judgment to evaluate what was being generated.

The AI happily wrote code. Lots of it. The person couldn't tell good code from bad. They didn't know when something was broken, and they didn't know the right questions to ask to fix it. **Two days and $2,000 in API costs later, we had nothing usable.**

The problem wasn't the AI. The AI did exactly what it was asked. The problem was giving a power tool to someone who had no framework for evaluating the output. They were a Slop Cannon — high volume, zero quality filter.

> THE LESSON
> AI coding tools don't make non-coders into coders. They make good coders faster. Without the underlying judgment, you're just burning money at scale.

Hours lost clicking a button

We were automating a process in a Windows desktop application — getting AI to interact with the interface, click buttons, fill forms. Someone on the team spent *hours* trying to get the AI to click a single button. It kept missing. They burned through time and tokens, trying the same thing slightly differently each time.

When I looked at it, the fix took minutes. The AI's click coordinates were off by a handful of pixels because of a header bar that shifted the layout. With some technical context, I could spot the offset, fix it, rewrite the script to be more robust, build it into a reusable skill, and set up iterative loops that got faster every run.

Same tool. Same problem. Completely different outcomes — because one person had the technical context to evaluate and correct the AI's output, and the other didn't.

> THE LESSON
> AI doesn't eliminate the need for domain expertise. It rewards it. The person with context gets 10x leverage. The person without it gets 10x frustration.

The $80K-a-month ad account ban

We know of an e-commerce business with tech-savvy owners. Early adopters. Comfortable with new tools. They went deep on AI ahead of most of their peers.

Their mistake was assuming the AI's recommendations across their Google Ads, Meta, and SEO accounts were better than the work coming from the human marketers who actually understood their strategy and their customers. When their marketing team flagged real issues, the owners pushed through anyway. They trusted the polish over the people. The AI's confidence read as authority.

–$80K/mo Google banned the ad account first. Meta followed. Revenue dropped almost overnight. The cleanup has been slow, expensive, and ugly. The business may not survive it.

> THE LESSON
> AI is a force multiplier, not a substitute for expertise. If you can't evaluate whether a recommendation is any good, and you can't guide the AI toward a better outcome, you don't have a tool. You have a confident voice telling you to do things you can't verify. Tech-savviness is not domain expertise.

In public

Samsung engineers upload source code to ChatGPT

Engineers uploaded proprietary source code directly to ChatGPT — confidential IP sent to an external AI service, because nobody had established clear boundaries about what data can and can't go into these tools. They weren't malicious. They were enthusiastic. Nobody had given them guardrails, and they didn't have the judgment to set their own.

The Chevrolet chatbot that sold a car for $1

A dealership put an AI chatbot on its website. A customer convinced it to agree to sell a vehicle for $1, with the chatbot claiming the offer was "legally binding." It became an internet joke overnight. No human oversight, no guardrails, no testing of edge cases.

Air Canada's chatbot gets the company sued

The chatbot made false claims about bereavement discounts. The customer relied on them, bought a ticket, then sued. The court ruled it "negligent misrepresentation" and held Air Canada liable — because a chatbot is part of the company, and the company is responsible for what it says.

The vibe coding epidemic

Non-technical people building entire software products with AI became widespread in 2025–2026. A Cloud Security Alliance study found **62% of AI-generated code contains security flaws**. "Rescue engineering" — fixing AI-built products — is tipped as the hottest discipline of 2026.

One founder built an entire SaaS product using Cursor AI with zero hand-written code. Within days of launch, attackers were bypassing subscriptions, API keys were exposed in frontend code, and the database was completely unprotected. The founder couldn't fix any of it because they didn't understand what had been built.

THE COMMON THREAD IN EVERY ONE OF THESE STORIES

People used AI without the judgment to evaluate, constrain, or correct its output. The technology performed as designed. *The culture wasn't ready.*

05 The Champion Model

How to *actually* do this

If 95% of AI projects fail, what do the 5% that succeed look like? They follow a pattern we've seen work at R6 Digital, at R6 Automate, and now at R6 AI when we help clients.

1 Identify your champions

Look at your team through the Judgment Matrix. Who are your Steady Hands — good judgment, not yet using AI? Who already shows signs of being a Turbo Brain?

Champions don't need to be the most technical people in the room. They need *taste* — the ability to look at an output and know whether it's good. They need curiosity, and the credibility with peers to eventually train everyone else.

Pick two or three. Not ten. Not the whole team.

2 Give them a real problem

Don't send champions to a generic AI workshop or give them a sandbox to "explore." Give them an actual business problem with a measurable outcome.

"Can we cut our campaign turnaround time in half?" "Can we automate this weekly report that takes four hours?" "Can we build a daily intelligence brief from these five data sources?"

Give them access to the tools, give them time, and give them permission to fail.

3 Let them build

Champions experiment. They test tools, write prompts, build workflows. Critically, they also figure out what *doesn't* work — where AI falls short, where it hallucinates, where it needs a human check.

This is the 10% of the effort. It looks like the main event, but it isn't.

4 Roll out through champions, not mandates

Here's the part most companies skip: **90% of the work is the rollout, not the build.** It doesn't go through an email from the CEO saying "start using AI." It goes through the champions — peer-to-peer, hands-on, showing real results.

Training that's specific

Not "here's how ChatGPT works." Here's the exact prompt for client reporting, what good output looks like, and three things to check before you send it.

Documentation that's practical

Prompts, processes, guardrails — written down, not in someone's head. "When you use AI for X, always check Y." "Never put client data into Z."

Culture shift that's visible

Celebrate first wins publicly. Celebrate catching a bad AI output before it reaches a client, too. Make it safe to experiment; make it clear quality still matters.

Iteration that never stops

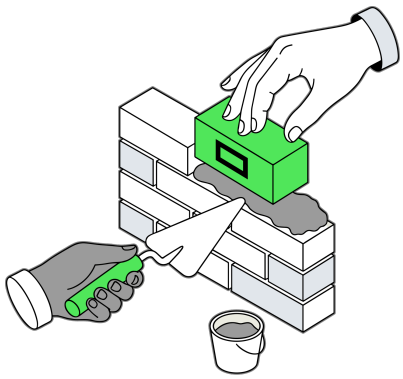
Feedback loops. Update the prompts. Remove workflows that don't deliver. Add new ones. An ongoing process, not a one-time project.

5 Set the guardrails

Every AI rollout needs clear rules — not to stifle innovation, but to prevent the Slop Cannon scenario.

- What data can and can't go into AI tools (learn from Samsung)
- What AI output requires human review before it goes external (learn from Air Canada)
- Who has access to which tools, and at what level
- How to report when AI produces something wrong or unexpected
- What the escalation path looks like when something goes sideways

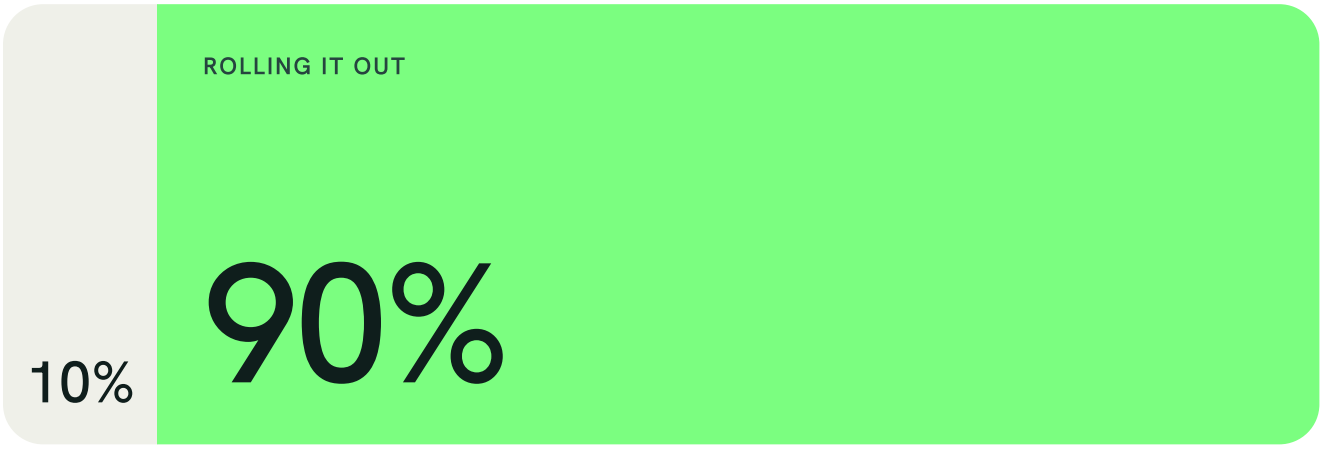
These aren't bureaucratic overhead. They're the difference between a productive AI culture and an expensive mistake.



06 The 90/10 Rule

The *90/10* rule

If you take one operational principle from this playbook, make it this:



Most businesses obsess over the build. They spend months evaluating tools, running pilots, debating architectures. Then they launch something to the team with a Slack message and wonder why nobody uses it.

The rollout is the work. The training is the work. The cultural permission is the work. The documentation is the work. *The iteration is the work.*

Tested on us first · R6 Automate

We've transitioned self-storage facilities from fully staffed to staffless operations using AI and automation. The technology build was a fraction of the effort. The real work was training operators, documenting processes, building trust in the systems, and iterating until it was bulletproof.

The tool is the easy part. Getting people to use it well is the hard part.

07 Bringing People on the Journey

Bringing people *on the journey*

AI adoption triggers fear. You can't skip or dismiss it — you have to address it head-on. People worry about being replaced, looking incompetent, being left behind. Unaddressed, those fears show up as resistance, sabotage, or people using AI badly because they feel pressured to adopt with no support.

What works

- **Start with a real problem, not AI for AI's sake.**
  People engage when AI solves *their* problem — a specific pain point they deal with every week.
- **Show the first win publicly.**
  In a team meeting, a Slack channel, a conversation at the coffee machine. Not in a management report.
- **Let champions be peers, not enforcers.**
  The best adoption happens sideways, not top-down. Champions should be showing, not telling.
- **Share failures openly.**
  "We tried AI for X, it didn't work, here's why" builds more trust than pretending it's perfect.
- **Give people permission to experiment safely.**
  The worst outcome isn't a failed experiment. It's a team too afraid to try.

What doesn't

- **Mandating AI usage from the top down.**
  This creates compliance without competence. People tick the box without engaging.
- **Giving everyone access with no training.**
  This is how you create Slop Cannons. Access without education is dangerous.
- **Letting anyone push AI output to clients without review.**
  Quality gates matter, especially early in the rollout.
- **Punishing people for not adopting fast enough.**
  Fear-based adoption produces worse outcomes than no adoption at all.
- **Treating it as an IT project instead of a culture project.**
  AI adoption lives in the team, not in the tech stack.

08 Your AI Culture Checklist

Your AI culture *checklist*

Here's the operational summary. Five steps, in order.

☐ *1*

Map your people on the Judgment Matrix.

Know who your Turbo Brains and Steady Hands are before you buy a single tool. If you can't identify the people with good judgment in your team, you're not ready for AI — you're ready for better hiring.

☐ *2*

Pick 2–3 champions.

Give them access, time, and a real business problem. Not a lab experiment. Not a sandbox. A problem with a dollar figure attached to it.

☐ *3*

Let champions build specific workflows.

Prompts, processes, guardrails. Test internally. Measure with actual numbers. "We saved 12 hours a week" is useful. "The team likes it" is not.

☐ *4*

Roll out through champions, not mandates.

Peer-to-peer. Hands-on. Show, don't tell. Celebrate the first wins visibly. Make it safe to experiment.

☐ *5*

Document everything. Iterate constantly.

Feedback loops. Update prompts. Remove what doesn't work. Add guardrails as you learn. An ongoing practice, not a one-time project.

About R6 AI

Tested on *us first.*

Every recommendation in this playbook comes from things we tested on our own P&L — across a marketing agency (R6 Digital) and a technology platform (R6 Automate) — before we ever shared them with a client.

We built AI into our own businesses first. We made the expensive mistakes. We found the patterns that work. And now we help other businesses skip the trial-and-error and go straight to the parts that actually make money.

If AI isn't the answer for your business, we'll tell you. *That's the deal.*

GET IN TOUCH
michael@r6digital.com.au
r6ai.com.au

Sources & further reading

MIT Media Lab (2025)	95% of generative AI pilots fail to deliver measurable ROI
BCG (2025)	"AI transformation is 70% people" (Ernesto Pagano, Managing Director)
Dan Hockenmaier	The Judgment Matrix (Slop Cannons and Turbo Brains framework), danhock.com
Alberto Romero, *The Algorithmic Bridge*	"Slop Cannons and Turbo Brains" analysis
Cloud Security Alliance	62% of AI-generated code contains security flaws
Stack Overflow (2026)	"A new worst coder has entered the chat: vibe coding without code knowledge"
Fortune (2025)	MIT report on generative AI pilot failures
McKinsey (2025)	"Reconfiguring work: Change management in the age of gen AI"
HBR (2026)	"Why AI adoption stalls, according to industry data"

Your Partner *in AI*

