

PRODUCT MANAGEMENT

FOR ENGINEERS:

BUILD SMARTER, NOT JUST FASTER



WHY ENGINEERING SKILL ALONE ISN'T ENOUGH



The technical craft of building software is genuinely hard.

But building the **right** software is a different challenge entirely. Engineers might lean toward building “cool” features or technically optimizing the product, even when those improvements don’t align with business priorities or customer needs. That’s not a character flaw. It’s a natural consequence of training that focuses entirely on the **how** without touching the **why**.



The problem compounds at scale.

Research on software project failures frequently identifies unclear, incomplete, or ambiguous requirements as a leading contributor to budget overruns — often outweighing execution-phase issues. Many overruns are not created during execution; they are built into the project foundation from the start. Engineers who understand product management principles can catch these problems **before** a single line of code is written.



The numbers on late-stage rework are sobering.

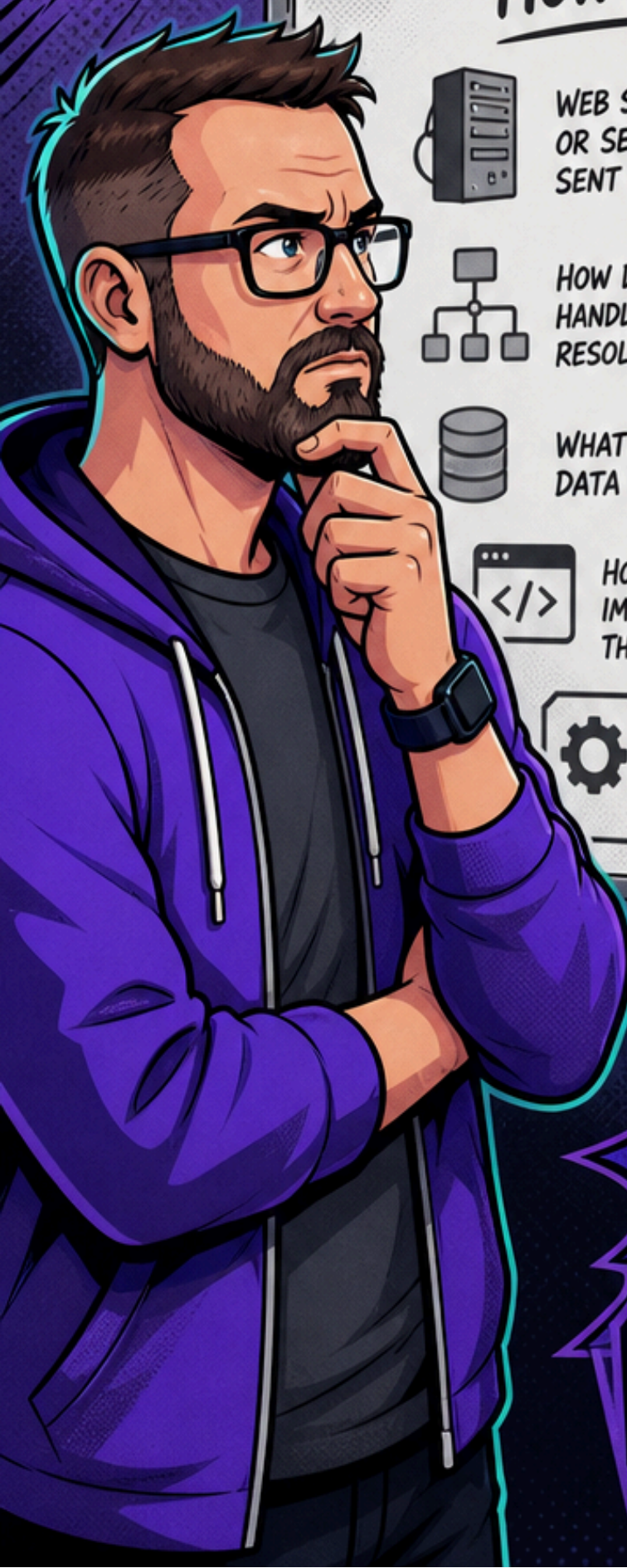
Industry analyses suggest that a significant share of rework stems from misunderstood or incomplete requirements — not just a technical problem, but a business loss. And research going back to Barry Boehm’s foundational 1981 work has long held that defects discovered late in the lifecycle cost substantially more to resolve than those caught during discovery — though the exact multiplier varies by context and has been debated in the literature. The practical implication holds: **catching problems earlier is almost always cheaper** than fixing them later.



LEARNING TO THINK LIKE A PRODUCT MANAGER DOESN'T MEAN ABANDONING YOUR ENGINEERING INSTINCTS. IT MEANS GIVING THOSE INSTINCTS MORE CONTEXT TO WORK WITH.



THE MINDSET SHIFT: FROM "HOW" TO "SHOULD WE"



How?



WEB SOCKETS
OR SERVER-
SENT EVENTS?



HOW DO WE
HANDLE CONFLICT
RESOLUTION?



WHAT'S THE
DATA MODEL?



HOW DO WE
IMPLEMENT
THIS?



THE ENGINEERING
BRAIN OPTIMIZES
SYSTEMS.

Should We?



HOW MANY USERS
ACTUALLY COLLABORATE
IN REAL TIME RIGHT
NOW?



WHAT PROBLEM
DOES THIS
SOLVE?



IS THIS A
MUST-HAVE FOR
THE NEXT RELEASE...

...OR A NICE-TO-HAVE
THAT COULD WAIT
TWO QUARTERS?



THE PRODUCT MINDSET
FOCUSES ON OUTCOMES.
NOT OUTPUTS.

ASKING THOSE QUESTIONS
DOESN'T SLOW DEVELOPMENT DOWN.

IT FOCUSES IT.



THE PRINCIPLE IS SIMPLE:
TRACK PROGRESS AGAINST **OUTCOMES**,
NOT OUTPUTS. DON'T MEASURE FEATURES
DELIVERED. MEASURE **PROBLEMS SOLVED**.

OPPORTUNITY SIZING: KNOWING WHAT'S WORTH BUILDING

SUPPORT TICKETS

- Login Issues 320
- Billing Errors 210
- Checkout Problems
- Feature Requests
- Other

FEATURE USAGE



POTENTIAL IMPACT

80%

OF ALL USERS

USER SEGMENT

- Power Users 5%
- All Users 80%

ESTIMATED VALUE



NEXT STEPS

- Validate with users
- Size technical effort
- Prioritize



RICE AND MOSCOW: FRAMEWORKS ENGINEERS CAN ACTUALLY USE

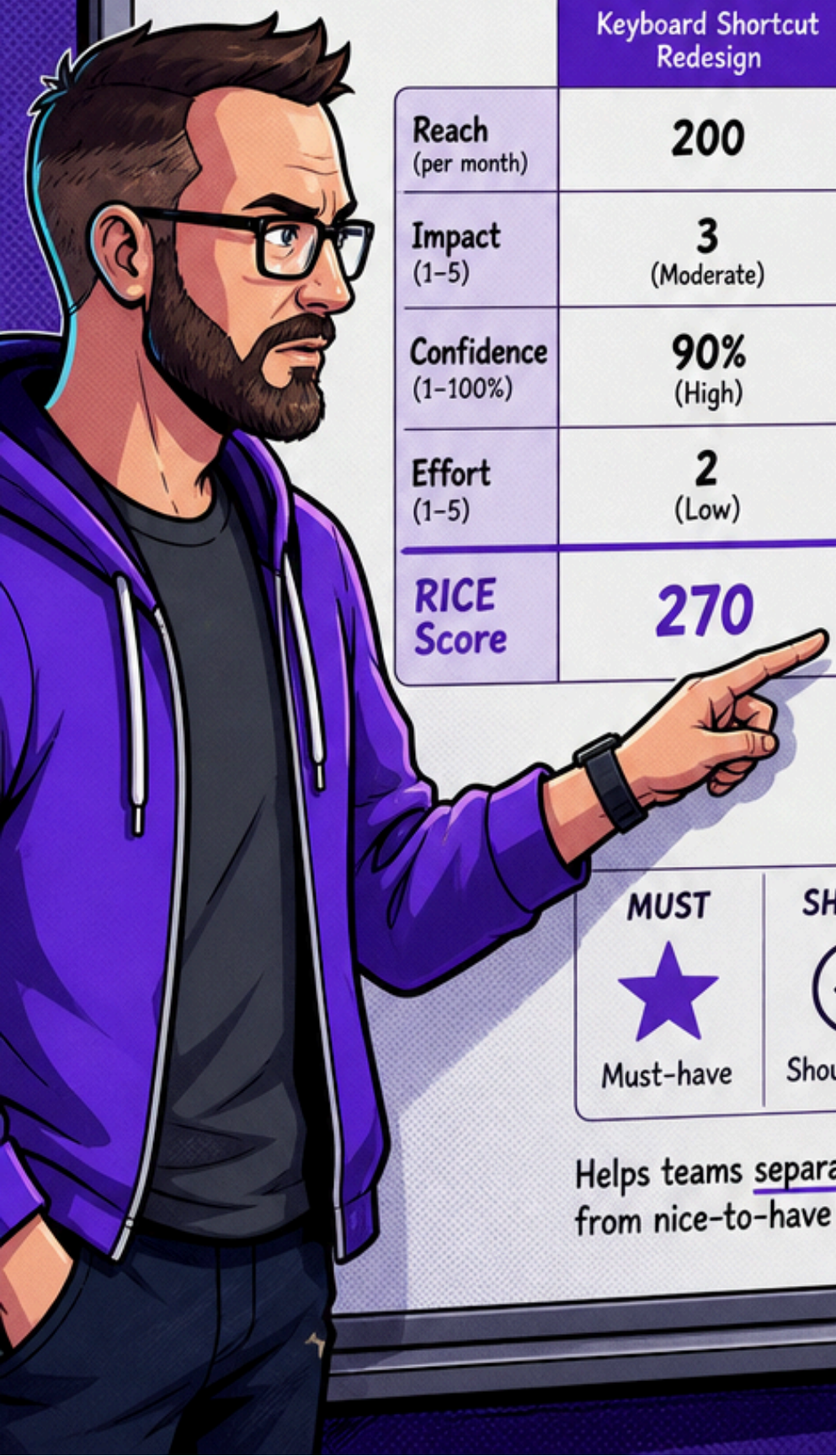
RICE

Reach, Impact, Confidence, Effort
Score = (Reach × Impact × Confidence) / Effort

	FEATURE A Keyboard Shortcut Redesign	FEATURE B Fix Mobile Onboarding Flow
Reach (per month)	200	5,000
Impact (1-5)	3 (Moderate)	5 (High)
Confidence (1-100%)	90% (High)	60% (Medium)
Effort (1-5)	2 (Low)	3 (Medium)
RICE Score	270	5,000

MUST ★ Must-have	SHOULD ✓ Should-have	COULD ○ Could-have	WON'T ✗ Won't-have this time
-------------------------------	-----------------------------------	---------------------------------	--

Helps teams separate critical requirements
from nice-to-have ideas.



USER STORY MAPPING: SEEING THE WHOLE JOURNEY

USER
ACTIVITIES →

DISCOVER

SIGN UP

ONBOARD

FIND /
SEARCH

VIEW
DETAILS

TAKE
ACTION

SHARE

USER
STORIES
↓

PRIORITY
(HIGH)

Browse
landing
page

Create
account

Welcome
tour

Search
items

View item
details

Add to
collection

Share
item

See
features

Verify
email

Set up
profile

Filter
related

Add to
cart

Filter
results

Read
blog
posts

Skip
tour

Save
search

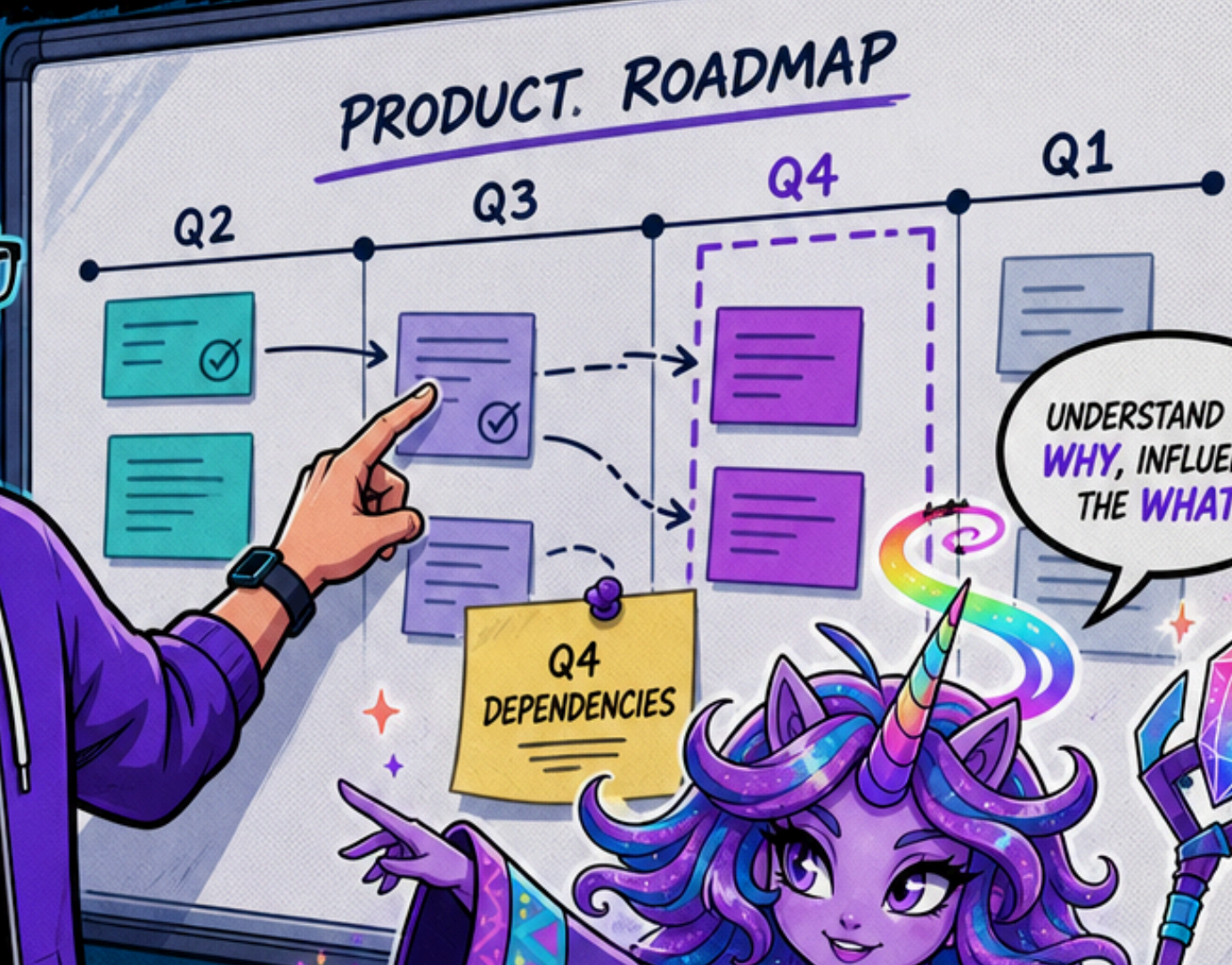
View
reminder

Set
remialar



ROADMAP LITERACY AND STAKEHOLDER REASONING

ROADMAPS ARE SETS OF BETS. EVERY PRIORITY HAS A REASON.



UNDERSTAND THE WHY, INFLUENCE THE WHAT.



ROADMAPS ARE BETS.
Each priority is shaped by assumptions, trade-offs, and business constraints.



ENGINEERS INFLUENCE OUTCOMES.
Share context only you have: effort estimates, technical debt, dependencies, and architectural impact.



SPEAK STAKEHOLDER.
Translate trade-offs into what sales, marketing, and customers care about.



MOVE FROM ORDER-TAKERS TO STRATEGIC PARTNERS.
Ask better questions, surface evidence earlier, shape better outcomes.

UNDERSTAND. INFLUENCE. BUILD BETTER.

CHALLENGING REQUIREMENTS CONSTRUCTIVELY







REQUIREMENT

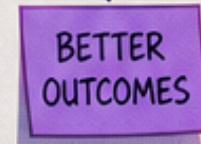
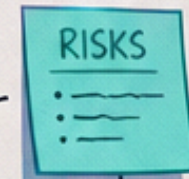


ASSUMPTIONS



PM-LITERATE ENGINEER QUESTIONS:

-  WHAT OUTCOME ARE WE TRYING TO DRIVE WITH THIS?
-  WHO ARE THE PRIMARY USERS AND HOW FREQUENTLY DO THEY HIT THIS PROBLEM?
-  HAVE WE VALIDATED THAT USERS WANT THIS, OR ARE WE ASSUMING?
-  WHAT'S THE SIMPLEST VERSION OF THIS THAT WOULD TEST THE HYPOTHESIS?

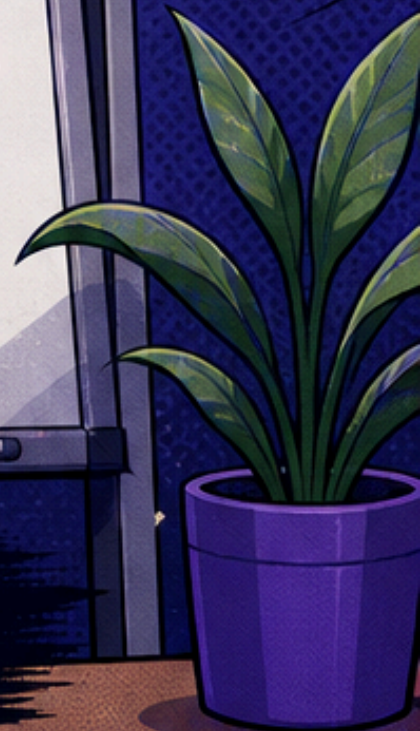


THOSE QUESTIONS AREN'T OBSTRUCTIONISM. THEY'RE DISCOVERY. What starts as a small gap in scope definition or ownership during discovery can cascade into rework, delays, and tech debt later on. Research on large project failures consistently finds that many cost overruns trace back to decisions made early, not late, in the lifecycle.

38-3, 38-4



AN ENGINEER WHO ASKS GOOD DISCOVERY QUESTIONS BEFORE WRITING A LINE OF CODE IS DOING SOME OF THE HIGHEST-LEVERAGE WORK ON THE TEAM. THEY'RE PREVENTING EXPENSIVE MISTAKES, NOT SLOWING THINGS DOWN.



LEADING DISCOVERY CONVERSATIONS

Discovery used to be considered exclusively the product manager's job. That model is increasingly outdated.

2-26

Modern teams are redesigning rituals so overlap is built in: shared discovery, joint spec-writing, and reviews where product, design, and engineering critique the work together instead of in separate lanes.

2-26

For engineers, participating in discovery means showing up to user interviews with genuine curiosity. It means reading support tickets and analytics before sprint planning, not just during it. It means being willing to say "I don't think we understand the problem well enough yet" when the evidence for a feature is thin.

26-13, 26-14

User story mapping is not about creating a set of written requirements, but a way of thinking. Telling stories through words and pictures builds understanding and helps solve problems for organisations, customers, and users.

26-13, 26-14

An engineer who internalizes this mindset brings that thinking to every planning conversation.

The payoff is concrete.

Evidence consistently shows that teams investing more heavily in discovery reduce downstream rework – the earlier problems are surfaced, the less costly they are to address.

Engineers who contribute meaningfully to discovery aren't doing less engineering.

They're doing better engineering, because the problems they're solving are the right ones.

USER INTERVIEWS



"I waste time re-entering the same info in multiple places."



"It's not clear what happens after I submit."

SUPPORT TICKET #8421



"I just want a simple way to track progress."



"The dashboard feels cluttered and confusing."



USER STORY MAP

	ACTIVATE	DISCOVER	USE	MANAGE	SUPPORT
USER ACTIONS					
THINKING					
FEELING					
OPPORTUNITIES					

WHAT THIS LOOKS LIKE IN PRACTICE



SALES TEAM

Pushing hard for SSO (single sign-on), citing five blocked enterprise deals.



SUPPORT TEAM



Logged 200 tickets about slow page loads.



CEO

Wants to ship an AI feature because competitors have one.

RICE SCORING

INITIATIVE	REACH (Users)	IMPACT (Score)	CONFIDENCE (%)	EFFORT (Person-Weeks)	RICE SCORE
 IMPROVE PAGE LOAD SPEED	50,000	2.5	80%	5	400
 SSO (SINGLE SIGN-ON)	5,000	3	70%	6	175
AI FEATURE	10,000	3	60%	10	180

DATA-INFORMED ROADMAP



A DECISION THE WHOLE TEAM CAN DEFEND.
NOT ARBITRARY. REASONED.

PRODUCT STRATEGY

USERS FIRST

DATA DRIVES

IMPACT OVER EGO



A PRACTICAL PLACE TO START

You don't need to become a product manager to benefit from these ideas. The goal is fluency, not a career change.



Start by sitting in on product planning sessions with genuine curiosity rather than waiting to receive tickets.



When you get a new requirement, spend five minutes asking who it's for and what outcome it's meant to drive.



MoSCoW

Try applying MoSCoW to your own backlog in your next sprint planning.



Sketch a simple user story map for a feature you're about to build, even if it's just on paper.

“The best advice for engineers interested in product thinking is to start acting like a product manager today, right in your current engineering role. Get obsessed with the customer and the business outcome of your work.”

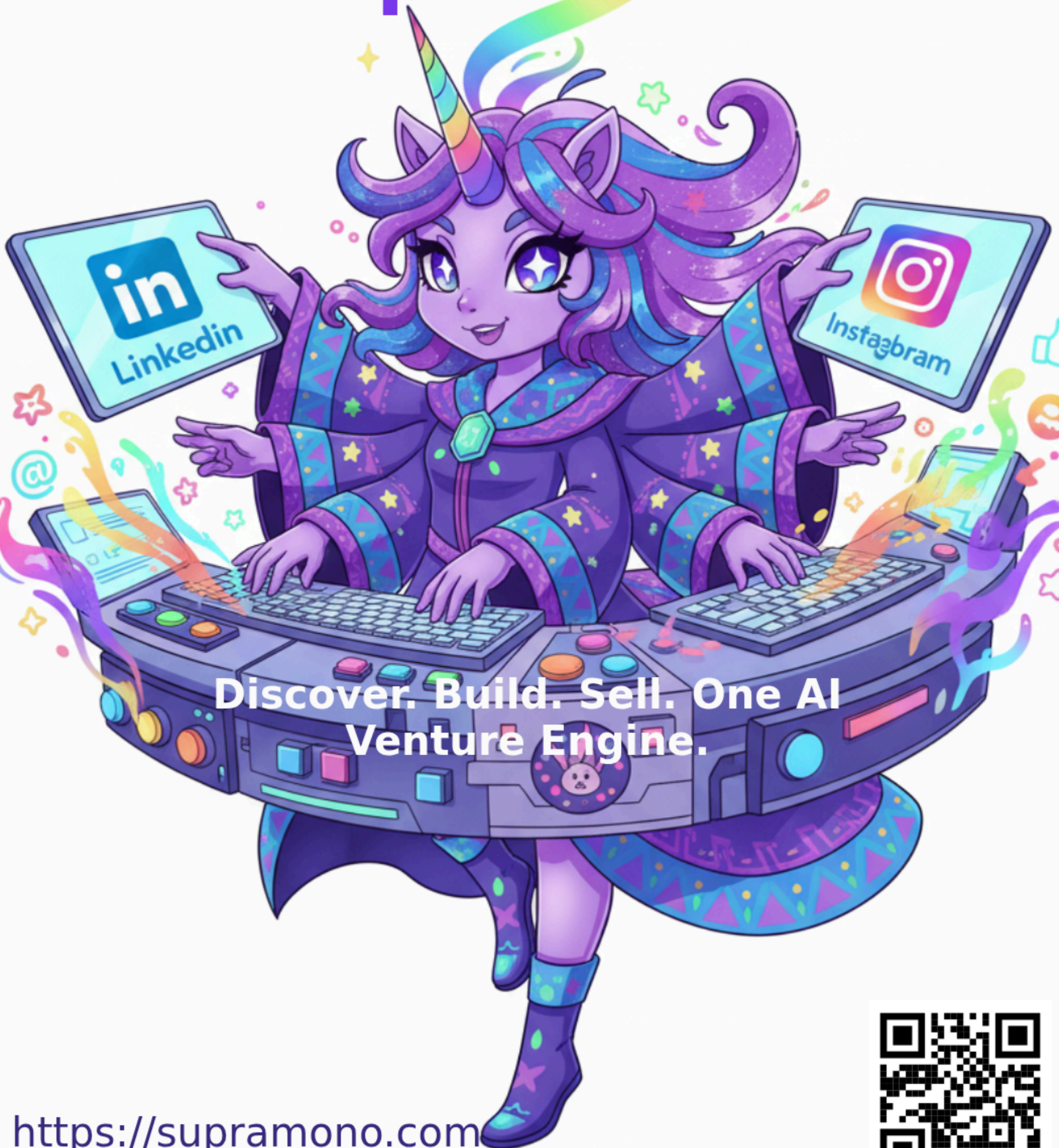
Technical skill remains the foundation. But engineers who have the most impact — whose work consistently matters to users and to the business — tend to be those who pair technical craft with product judgment. They know how to build. They also know when not to, and what to build instead.

That combination is rare. It's also learnable.

supramono
Founder



supramono



Discover. Build. Sell. One AI
Venture Engine.

<https://supramono.com>

