

Balancing Experience & AI-Driven Productivity

BALANCE CLARITY WITH COMPASSION

As AI becomes more deeply woven into engineering workflows, leaders are asking how to make it work for their people, not just their products.

This roundtable brought together senior engineering leaders to explore how AI is reshaping productivity, processes, and team structures, and what's actually driving measurable impact.

SESSION LED BY:



Meng Muk
VP of Engineering
 Automata

PURPOSE OVER POLISH: REDEFINING "GOOD ENGINEERING"

AI is changing what 'good engineering' means. It's less about the craft and more about the outcome, the ability to deliver faster, experiment confidently, and apply the right level of human oversight.

- Engineers who believe in the problem they're solving are often more willing to experiment with AI tools.
- The craft of coding is shifting toward systems thinking and prompt engineering.
- Senior engineers should focus on the "why" behind decisions, not just the "how" of execution.

Actions for leaders

- ✓ Create space for experimentation within clear guardrails.
- ✓ Update performance frameworks to reward outcomes, not output volume.
- ✓ Encourage problem-first, not tool-first, conversations in sprint planning.

REMEMBER! → "We're here to solve real problems, not write good code."

BALANCING EXPERIENCE & AI FLUENCY: DESIGNING MODERN TEAMS

- Seniors are shifting from "doing the code" to designing the system and mentoring both people and models.
- Juniors often promote good prompt-engineering techniques, reverse mentoring is becoming essential.
- Teams without juniors risk losing innovation, curiosity, and future leaders.

TOP TAKEAWAYS FOR ENGINEERING LEADERS

Define your AI "why" where it adds measurable business value.

Blend senior **experience** with AI fluency for resilient, future-ready teams.

Strengthen **observability** and **governance** as AI speeds up delivery.

Update competency **frameworks** to reflect AI skills, usage, and teaching.

MEASURING AI PRODUCTIVITY: BEYOND "IT FEELS FASTER"

- AI boosts speed but not always effectiveness; the bottleneck often lies in deployment, not code generation.
- Developer-experience tooling (DX) can measure improvements, but adoption is inconsistent.
- Non-engineering teams often see clearer ROI from AI tooling than engineering squads.

AI READY CHECKLIST:

Strategy & Purpose

- ☐ Is your problem statement clear, specific, and worth solving with AI?
- ☐ Have you set clear boundaries for cost, security, and ethical use from day one?

Team Design

- ☐ Do you have a balanced mix of engineering depth and AI fluency across the team?
- ☐ Is knowledge shared both ways (e.g. juniors teaching AI tools, seniors teaching system thinking)?

Tools & Implementation

- ☐ Do you pilot tools with a small group before rolling out to the whole team?
- ☐ Is there a clear separation between experimentation and production workflows?
- ☐ Have you documented simple, non-negotiable internal guidelines for daily AI use?

Culture & Learning

- ☐ Do teams openly share what worked, what failed, and what they're learning?
- ☐ Have you updated role expectations and competency frameworks so AI skills are recognised?
- ☐ Do you value impact and outcomes rather than overly polished code?

Measurement

- ☐ Do you balance quantitative data with team sentiment to understand the real impact?
- ☐ Do you run regular AI-focused retros to refine processes?
- ☐ Are decisions still grounded in human judgement, customer empathy, and business context?