

Balancing Engineering Speed & Quality in the Age of AI

SESSION LED BY:



**Apurva
Deshpande**

Principal
Engineer

rightmove 



**Jonathan
Ruckwood**

Staff
Engineer

 mapbox

We brought together Staff, Principal and Lead Engineers to explore how teams can embrace the speed of AI-assisted development without compromising quality, accountability or engineering judgement.

ENGINEERING JUDGEMENT MATTERS

- AI can help teams move faster, but engineers still need curiosity, technical depth and the judgement to spot a wrong answer.
- Breadth helps with exploring unfamiliar problems; depth helps understand the consequences of a decision.
- Senior engineers set the standards, put guardrails in place and connect technical choices to customer value.
- Measure success by customer impact, reliability and maintainability, not output alone.



Tool recommendation: Repomix

Repomix packages a codebase into a single file that can be shared with an AI tool.

It gives the AI more of the repository's context when you're exploring unfamiliar code, planning a refactor or reviewing changes.

GUARDRAILS & ACCOUNTABILITY

- Tests, coding standards, secure deployment processes and human review remain essential.
- Encode repeatable patterns into rules files and style guides to improve consistency.
- AI review is useful for stylistic and rule-based checks, but cannot replace human judgement.
- Engineers must understand, review and take responsibility for any AI-generated work they submit.

WHERE IS AI ADDING VALUE?

- AI is used for prototyping, repetitive tasks and large-scale code changes that would previously have been too time-consuming.
- Live prototypes can make product conversations more tangible, helping customers and stakeholders understand an idea before significant engineering time is committed.
- Fast prototypes can also expose weak ideas earlier, allowing teams to reduce scope or abandon work before over-investing.
- AI should remain a tool, not a required way of working. If it adds friction or produces worse results for a particular task, engineers should use a different approach.

ARE YOU CREATING STRUCTURED AI WORKFLOWS?

AI performs better when the problem, constraints and expected outcome are clearly defined. Top tips:

- Break work into small, logical stages with checkpoints and review points, rather than asking an agent to complete an entire feature in one step.
- Smaller models can be useful for planning before implementation begins.
- Use logical commits so engineers can follow what changed, understand the reasoning and intervene when necessary.
- Ask AI to explain why it made key decisions, not only what code it produced.
- Keep experiments lightweight. A prototype should be well-reasoned, but it does not always need to be highly polished.

AVOIDING AI INFRASTRUCTURE OVERLOAD

Teams are building additional prompts, agents, evaluators and knowledge systems around AI. This supporting infrastructure can become another technical burden to maintain.

Human-written documentation that captures business context remains more useful than automatically generated content. Just be careful to check that the infrastructure around AI is creating more value than complexity.