

Preserving Engineering Craft While Building AI-Native Teams

THE TRADITIONAL ENGINEERING PLAYBOOK IS CHANGING

We brought together a group of Engineering Leaders and technical decision-makers to discuss the evolving role of senior engineers, the rise of AI-native development practices and why accountability, judgment and strong engineering fundamentals still matter more than ever.



SESSION LED BY:
Manogna Machiraju

Head of
Engineering



AI has turned code generation into a mass-production assembly line. Traditional management methods can struggle under the sheer speed of AI-assisted development.

Leaders are focusing on:

- Building resilient systems
- Improving architectural guardrails
- Strengthening rollback and recovery
- Maintaining accountability and oversight
- Helping engineers stay in control of the code

MOVE FROM LINE-BY-LINE REVIEWS TO SAMPLE TESTING

As AI-generated code volume increases, manually reviewing every line of code is becoming increasingly difficult to scale. Instead, leaders are moving toward:

- Automated quality controls
- Higher-level architectural reviews
- Sampling strategies
- Stronger testing frameworks
- Better observability and monitoring

AI has changed the speed of engineering. Quality control needs to evolve with it.

WHO IS IN CONTROL? YOU OR AI...

Rather than relying purely on output metrics:

- **Embrace anti-fragility.** You don't need to prevent all failures; instead focus on building systems that can survive and recover from anything going wrong
- **Use agents for low-hanging fruit.** Automate security patches, bug fixes, and routine tasks so engineers can focus on higher-order thinking
- **Protect critical thinking.** Over-reliance on AI and "AI slop" can weaken engineers' ability to think critically, so engineers still need to remain the brain behind their work



HOW ARE LEADERS ADAPTING

- Reducing reliance on manual PR reviews and investing more heavily in automated quality controls, observability and testing frameworks
- Shifting senior engineers away from pure implementation and toward architecture, resilience, systems thinking and AI oversight
- Building stronger rollback, recovery and isolation capabilities instead of trying to prevent every production issue
- Hiring for adaptability, judgment and problem-solving ability over specific tooling knowledge alone

HUMAN ACCOUNTABILITY REMAINS CRITICAL

- **Maintain confidence in your own coding ability.** If you lose the ability to code independently, you won't be able to judge the quality and accuracy of AI output.
- **Use AI for the right things.** Brainstorming, formatting, admin, generating ideas, but apply human judgment to evaluate the output.
- **Avoid blindly shipping AI-generated code.** Engineers still need to understand what they're deploying so they can troubleshoot issues when things go wrong.
- **Build qualitative feedback loops.** Ask your engineers where friction exists, whether they feel in control of the systems they're building, and where AI is genuinely improving productivity