# Podcast: “Training for AI-Augmented Dev” – Experts Weigh In

## The Watercooler of Engineering Leadership

When engineering leaders need to make sense of rapid change, they turn to podcasts.

The AI platforms market has grown from $12.3B in 2022 to $109.9B in 2025 and is projected to reach $181.3B in 2026, with software engineering emerging as one of the most active deployment fronts. Nearly half of organizations now cite code generation as a priority generative AI use case, and 39.6% plan agentic AI deployments for software engineering within 18 months [1].

This explosive growth has created an urgent need for practical guidance. Engineering leaders are under mounting pressure to evaluate, deploy, and govern AI-assisted development tools at scale, and they're turning to podcasts as a trusted source of actionable insights [1].

What are these leaders saying about training for AI-augmented development? A clear consensus is emerging.

## The Consensus: Structured Training Wins

Across the engineering podcast ecosystem, one theme dominates: generic AI literacy programs are failing.

Josh Cavalier, one of the most recognized voices in AI for learning, describes a shift from casual prompting to structured workflow design. The most mature teams are creating repeatable processes that blend human insight with machine capabilities. They are automating assets while strengthening their ability to focus on more complex design and stakeholder alignment [2].

His advice for learning leaders is practical: "Start with the work, not the technology. Identify where speed, consistency, or cognitive load create bottlenecks. Build small but durable AI workflows around those tasks. And create communities inside the organization where teams can share prompt patterns, mistakes, and experiments without fear".

AI maturity grows faster when collaboration is intentional.

## The 2025 Engineering Leadership Survey

Lucas Mendes, CEO of Revelo and host of Tech Teams Today, synthesized insights from over 50 engineering leaders navigating the AI era. The findings reveal the shifting landscape [3]:

| Priority | What Leaders Are Doing |
| --- | --- |

| | |
|---|---|
| Pacing AI adoption with ROI focus | Most teams are still working on adoption as a first step, with ROI still a question mark |
| Building distributed teams with ownership culture | Remote is default, but culture and connection need to be intentional |
| Staying close to work without micromanaging | 51% of leaders maintain some technical engagement |

*"AI is the bricklayer, and every dev is evolving into this new 'architect' role,"* one leader noted. *"For the ROI to be interesting, AI will have to do more than just be a better version of StackOverflow"*.

The distinction is critical: 36% of leaders say the ROI of AI tools will be top of mind within 12 months, and 73% are shifting from traditional metrics to business outcomes [3].

## The Mindset Shift: From Features to Systems

Adam Bilišič, former CTO and now CEO of NodeonLabs, provides a framework for mastering AI-augmented coding through five distinct levels [4].

**The fundamental shift? Moving from feature-level thinking to systems-level thinking.**

*"When you start building systems, instead of thinking 'how can I solve this feature,' you are thinking 'how can I create either a skill, command, sub-agent, or other things which these tools offer, to then do this thing consistently again and again without repetition'"* [4].

The framework progresses from basic prompting to building autonomous multi-agent systems:

| Level | Skill |
|---|---|
| Level 1 | Master of Prompts |
| Level 2 | Configuration Expert |
| Level 3 | Context Master |
| Level 4 | Automation Master |
| Level 5 | The Orchestrator |

*"If you didn't catch up or change your opinion in the last 2-3 years, I would say we are getting to the point where it will be kind of last chance to do so, because the technology is evolving so fast"*.

## The Critical Distinction: Vibe Coding vs. Augmented Coding

Bilišič draws a crucial line between vibe coding and AI-augmented coding [4].

Vibe coding allows non-developers to create functional applications without understanding the underlying architecture. It's useful for product owners to create visual prototypes, but it's not a career strategy.

AI-augmented coding, however, is what professional software engineers need to master: using AI tools while maintaining full understanding of the system's architecture, security implications, and best practices [4].

The key difference is that augmented coding lets you delegate repetitive work while retaining deep knowledge of what's happening under the hood.

## The Human Factor: Confidence as Capability

Betty Dannewitz explored how imposter syndrome affects learning professionals during technological change [2]. Her interviews reveal that self-doubt often increases when AI appears to match or surpass tasks that once relied entirely on human skill.

*"Confidence is as important as capability. Psychological safety influences experimentation, collaboration, and performance. Investing in the emotional resilience of the team is not separate from operational excellence. It supports it".*

This insight applies equally to engineering teams adopting AI. When developers fear being replaced, they resist new tools. When they feel supported, they experiment, learn, and grow.

## The Hard Truth

The podcast ecosystem is speaking with a unified voice.

**Generic AI literacy programs are not working.** The teams making real progress are those with structured approaches, clear ownership, and deliberate frameworks for building AI fluency.

**The developer's role is shifting from builder to architect.** The premium is on judgment, evaluation, and system thinking—not just code generation.

**Training must be role-specific.** What a junior developer needs differs fundamentally from what a senior architect needs.

**Psychological safety matters.** Teams that feel safe to experiment and fail will learn faster than those that don't.

## The Choice Ahead

The engineering leaders shaping this conversation are clear: the question is no longer whether to adopt AI-assisted development tools, but how to use them effectively and at speed [1].

The podcast experts agree on what comes next. Success isn't about checking a training box or giving developers access to another tool. It's about building structured AI fluency—the skills, judgment, and workflows needed to turn AI capabilities into meaningful engineering outcomes.

**The question isn't whether you can afford to build AI fluency in your engineering team. It's whether you can afford to wait.**

Because AI adoption without capability creates activity, not advantage. The teams that invest in structured learning today will be better positioned to move faster, solve problems more effectively, and get greater value from their AI investments.

The teams that don't may eventually find themselves asking a difficult question:

Why did we invest in AI, but never realize its full potential?

The choice is being made now.

## References


1. https://futurumgroup.com/insights/are-podcasts-the-new-playbook-for-engineering-leaders-work-through-ai-in-the-sdlc/
2. https://www.intrepidlearning.com/blog/devlearn-2025/#brx-footer
3. https://leaddev.com/wp-content/uploads/2025/06/Engineering-leadership-playbook-2025-Three-lessons-every-engineering-leader-needs-this-year-Lucas-Mendes-LDX3-London-2025.pdf#1#1
4. https://html5-player.libsyn.com/embed/episode/id/40048970/height/45/theme/standard/thumbnail/yes/direction/forward/tdest_id/246431/