INTERTECH ENGINEERING CONVERSATIONS

AI Engineering Transition Handbook

Episode 1: From Individual AI Experimentation to Organizational Capability

Purpose: Help an IT or engineering leader discover how AI is actually being used today, evaluate whether those uses create real engineering value, and begin a repeatable learning loop without prematurely standardizing experimentation.

Core idea

An organization does not become AI-enabled simply because individual developers become faster. It becomes AI-enabled when it can identify what works, evaluate the outcome, share the learning, validate the practice, and eventually make proven practices repeatable across the engineering organization.

Episode 1 Deliverable

A visible inventory of current AI usage, a first-pass evaluation of value and risk, a small group of trusted reviewers, and a documented learning loop for deciding what deserves further validation.

Episode Summary

Most engineering organizations are no longer starting from zero. Developers are already experimenting with AI for tasks such as understanding legacy code, generating or improving tests, code review, diagnosis, documentation, and learning unfamiliar frameworks. The challenge is that these practices are often personal: each developer has different prompts, tools, review habits, and assumptions.

Episode 1 argues that the first leadership task is not to impose a large AI policy or standardize every workflow. It is to make current AI usage visible. Leaders should discover what developers are actually doing, identify

examples that genuinely improve engineering outcomes, collect examples that created hidden downstream work, and begin comparing those experiences as a team.

The organization then needs a lightweight way to distinguish experimentation from a practice that may be ready to share or standardize. Evaluation should look beyond apparent speed and consider quality, defects, testing, maintainability, architectural fit, security, review effort, repeatability, and the amount of senior engineering judgment required.

Experienced engineers remain central because AI can generate plausible options without automatically understanding the organization's history, architecture, business rules, security boundaries, or prior decisions. Their review questions and reasoning are themselves valuable organizational knowledge that should begin to be captured and reused.

The progression introduced in this episode is:

#	Stage	Leadership intent
1	Discover	Make current AI usage visible.
2	Evaluate	Determine what became better - and what hidden work or risk was created.
3	Share	Expose useful practices and failures to the broader engineering team.
4	Validate	Test whether a practice works beyond the person who discovered it.
5	Standardize	Adopt only practices that repeatedly create better engineering outcomes.
6	Govern	Protect the standards, information, architecture, quality, and security that matter.

IT Leader Action Checklist

Use this checklist to run the first stage of the transition. Do not treat every checked item as a new policy. The objective is visibility, learning, and evidence.

Prepare the discovery session

- Select a representative group of developers, technical leads, architects, and engineering managers.
- Include several trusted senior engineers who understand the systems, architecture, customers, security requirements, and engineering standards.
- Set the expectation that the session is for learning, not policing individual AI usage.
- Ask each participant to bring one example where AI genuinely improved engineering work.
- Ask each participant to bring one example where AI looked productive but created rework, risk, or downstream effort.
- Ask participants to be prepared to show what they actually did rather than describe an idealized workflow.

Make current AI usage visible

- Record the engineering task being performed - e.g., legacy-code discovery, testing, review, debugging, documentation, or framework learning.
- Record the AI tool or approach used, but keep the focus on the engineering outcome rather than the tool itself.
- Capture any reusable prompts, instructions, context, or workflow steps that appear important to the result.
- Identify where practices differ substantially among developers.
- Identify duplicated discoveries - cases where multiple developers independently solved the same AI-assisted problem.

Evaluate the claimed value

- **Ask:** What became better because AI was used?
- **Determine** whether the approach actually saved total engineering time, not merely the first developer's time.
- **Check** whether review, testing, correction, maintenance, or downstream integration added hidden effort.
- **Assess** whether the practice improved quality, test coverage, documentation, understanding, or risk reduction.
- **Determine** whether the result fits existing architecture, engineering standards, and business rules.
- **Determine** whether the practice exposes proprietary, sensitive, regulated, or otherwise protected information.
- **Ask** whether another developer could repeat the approach successfully.
- **Identify** how much experienced engineering judgment is required to use the practice safely.

Create the first learning loop

- Choose a small group or existing technical authority that will review promising practices.
- Define who can recommend that an experiment be shared more broadly.
- Define who can validate that a practice is ready to become a shared engineering practice.
- Capture the reasoning senior engineers use when accepting, modifying, or rejecting AI-assisted work.
- Document useful failures as well as successes.
- Create a simple shared location for validated practices, prompts/instructions, review questions, and lessons learned.
- Schedule a recurring review point so experimentation can continue while useful discoveries are periodically evaluated.

Avoid premature standardization

Do not require every developer to use the same prompt or workflow.

Do not create a large AI manual before the organization understands its actual usage patterns.

Do establish minimum guardrails around protected information, accountability, architecture, review, testing, and knowledge sharing.

Keep room for continued experimentation inside those guardrails.

Standardize only after a practice has demonstrated repeatable engineering value.

Leadership self-check

Question	Yes	Partly	No	Evidence / Notes
Do we know how our developers are actually using AI?	■	■	■	
Do we know which uses are producing better engineering outcomes?	■	■	■	
Do we have a way for one engineer's learning to benefit the rest of the organization?	■	■	■	
Do we know who decides when an AI-assisted practice is ready to become a shared practice?	■	■	■	

Working Session: AI Usage Discovery

Recommended format: 60-90 minutes with a representative engineering group. Each participant presents no more than one or two examples. Capture evidence, not sales pitches for AI.

Opening prompt

"Show us how you are actually using AI today. Bring one example where it genuinely improved engineering work and one example where it looked productive but created work somewhere else."

For each example, ask:

- What engineering task were you trying to accomplish?
- What did AI actually do?
- What became better because of it?
- What time was saved, and where was additional time spent later?
- Did quality, testing, documentation, understanding, or risk improve?
- What review or correction was required?
- Did the result fit our architecture and established patterns?
- Did the approach require sensitive or proprietary information?
- Could another developer repeat this safely and successfully?
- How much senior engineering judgment does this approach require?
- Should the rest of the engineering team know how to do this?

Session notes

AI Practice Evaluation Worksheet

Complete one copy for each practice that appears promising enough to investigate further.

Practice / use case	
Engineering task or problem	
Developer(s) who demonstrated it	
AI tool / model / environment	
What AI contributed	
Claimed engineering improvement	

Criterion	Evaluation question	Strong	Mixed	Weak
Total time	Did it reduce total effort after review, testing, correction, and integration?	■	■	■
Quality	Did it reduce defects or improve correctness?	■	■	■
Testing	Did it improve coverage or identify useful edge cases?	■	■	■
Understanding	Did it improve comprehension of code, architecture, or business rules?	■	■	■
Architecture	Does it respect existing boundaries, patterns, and intentional dependencies?	■	■	■
Security / data	Can it be used without exposing protected information?	■	■	■
Maintainability	Is the result understandable and supportable later?	■	■	■
Repeatability	Can another engineer reproduce the benefit?	■	■	■
Judgment dependency	How much senior engineering judgment is needed to use it safely?	■	■	■
Downstream work	Did it move effort to review, testing, maintenance, or another team?	■	■	■

Disposition

- Continue experimenting - promising, but more evidence is needed.
- Share with team - useful enough that others should learn and test it.
- Validate formally - candidate for a repeatable shared engineering practice.

- Do not promote - hidden effort, quality, architecture, security, or repeatability concerns outweigh the benefit.
- Escalate for security / architecture review before further use.

Reviewer reasoning / lessons to retain:

What to Retain for the Handbook

The value of this exercise increases when the outputs survive the meeting. Retain the following artifacts so later episodes can build on them rather than restarting the AI conversation.

- Current AI Usage Inventory - a list of real engineering use cases currently occurring.
- Validated Wins - practices that have demonstrated measurable or observable engineering value.
- Useful Failures - cases where apparent speed created rework, risk, architectural drift, or maintenance burden.
- Reusable Prompts / Instructions - only where they materially contribute to a proven workflow.
- Engineering Review Questions - questions experienced engineers use to judge AI-assisted work.
- Security and Information Boundaries - early findings about what information may or may not be supplied to AI systems.
- Practice Owners / Reviewers - the people or group responsible for evaluating promising practices.
- Open Questions - issues that need evidence, testing, architecture review, security review, or later governance decisions.

Episode 1 maturity checkpoint

- You are ready to move forward when you can reasonably say:
- We have made current AI usage visible across a representative portion of engineering.

- We are evaluating AI based on engineering outcomes rather than activity or generated code volume.
- We are capturing both benefits and hidden downstream costs.
- We have trusted engineering reviewers involved in evaluating promising practices.
- We have a place to retain shared learning.
- We have not confused early evidence with a final standard.