INTERTECH ENGINEERING CONVERSATIONS

AI Engineering Transition Handbook

Episode 3: Engineering Standards Without Killing Experimentation

Purpose: Help an IT or engineering leader turn proven AI-assisted practices into shared engineering expectations without freezing individual technique, slowing experimentation, or allowing every developer to maintain a different definition of good.

Core idea

Standardize what must remain consistent - not everything that can vary. A useful standard protects an engineering outcome that matters to the organization. Prompts, interaction style, and personal technique can remain flexible unless variation creates meaningful risk, cost, or confusion.

Episode 3 Deliverable

A small first set of AI engineering standards built from Episode 2 evidence, each with a clear protected outcome, classification as experimental/ recommended/required, practical guidance, an owner and review point, and a lightweight path for engineers to challenge or improve the standard.

Episode Summary

Episodes 1 and 2 deliberately delayed standardization. Episode 1 made current AI usage visible. Episode 2 traced promising practices through the lifecycle to determine whether they produced reliable leverage. Episode 3 begins only after that evidence exists.

The central distinction is between an engineering expectation and a personal technique. An organization may require that an engineer accepting AI-generated production code understands the

implementation, verifies architectural fit, and remains accountable for the result. It does not automatically need to prescribe the exact prompt, number of interactions, or preferred workflow used to get there.

A second distinction is between a standard and the guidance around the standard. The standard defines what must be protected. Guidance can contain examples, prompts, workflows, and recommended techniques that can change much faster as AI capabilities improve.

Practices should not all become mandatory. A lightweight classification - experimental, recommended, or required - lets the organization spread useful learning without turning every successful idea into a rule. Standards should also have an owner, a reason for existing, a review trigger, and an explicit way for engineers to propose something better.

The progression introduced in this episode is:

#	Stage	Leadership intent
1	**Select**	Choose a few Episode 2 practices with strong evidence or meaningful inconsistency.
2	**Separate**	Define what must remain consistent and what can safely remain flexible.
3	**Explain**	State the organizational outcome or risk the expectation protects.
4	**Classify**	Mark the practice experimental, recommended, or required.
5	**Own & surface**	Assign ownership/review and place the expectation close to the engineering work.
6	**Improve**	Give engineers a lightweight path to challenge the standard with evidence.

IT Leader Action Checklist

Use this checklist after completing the Episode 2 lifecycle evaluation. The objective is not to create a complete AI policy. The objective is to establish a small, usable standards process that converts evidence into shared engineering practice.

Select the first standards candidates

- Start with the **Reliable Leverage Candidates** and conditions for success retained from Episode 2.
- Choose approximately three practices. Prefer strong evidence or areas where inconsistent behavior creates meaningful engineering or information risk.
- Do not begin with the most impressive AI demonstration simply because it is impressive.
- Confirm that the evidence includes the context, engineering judgment, validation, and downstream effects required for success.
- Keep practices with insufficient evidence experimental rather than forcing a standard.

Separate the standard from the technique

- For each candidate, ask: What needs to remain consistent here, and what can safely remain flexible?
- Write the engineering expectation at the highest level that still protects the required outcome.
- Do not automatically standardize exact prompts, number of AI interactions, chat style, or individual workflow.
- Capture useful prompts and workflows as guidance or examples when they materially help engineers satisfy the expectation.
- Ask whether a qualified engineer could use a different technique and still produce the protected outcome. If yes, preserve that flexibility.

State what the standard protects

- Complete the sentence: 'We do this because...' If the reason is unclear, reconsider whether the item should be a standard.
- Identify the protected outcome - for example information protection, architectural integrity, maintainability, accountability, quality, or another documented engineering need.
- Distinguish an organizational requirement from a team's preference.
- Carry forward the conditions that made the Episode 2 practice succeed; do not standardize the visible shortcut while losing the engineering reasoning behind it.

Classify the practice

- **Experimental:** promising or actively used, but the organization does not yet have enough evidence to promote it broadly.
- **Recommended:** evidence shows consistent value and the organization wants engineers to use it where appropriate, while allowing reasonable alternatives.
- **Required:** variation creates enough risk, cost, or engineering inconsistency that a common expectation is necessary.
- Do not make 'required' the default. Require the outcome when possible rather than unnecessarily requiring the technique.
- Allow practices to move in either direction as evidence, tools, systems, or risk change.

Make each standard maintainable

- Assign a named owner or clearly accountable role.
- Record when the standard should be reviewed and what events should trigger an earlier review, such as a major tool, model, architecture, security, or regulatory change.
- Place the expectation close to the work. Code-review expectations belong in review workflows; system-entry practices may belong in team engineering guidance or onboarding.

- Maintain a central standards record, but do not rely on developers remembering to search a large AI document.
- Communicate changes with the reason and evidence behind them.

Keep standards challengeable

- Tell engineers explicitly that the current standard represents the best evidence available today, not the end of improvement.
- Provide a lightweight challenge path: show the alternative, explain what it improves, and validate it against the same protected engineering outcome.
- Use lifecycle evidence to compare the proposed alternative with the current practice rather than debating preferences.
- If the alternative is better, update the standard or guidance and communicate why.

Leadership self-check

Question	Yes	Partly	No	Evidence / Notes
Can we explain what each first standard protects?	▪	▪	▪	
Have we separated required outcomes from personal AI technique?	▪	▪	▪	
Are practices classified as experimental, recommended, or required based on evidence?	▪	▪	▪	
Does each standard have an owner and review point?	▪	▪	▪	
Can an engineer propose a better approach without quietly ignoring the standard?	▪	▪	▪	

Working Session: Create the First Standards

Recommended format: 60-90 minutes with a small group that already has engineering credibility - for example a technical lead, engineering manager, and architect where appropriate. Bring the Episode 2 evidence for approximately three practices. The objective is to create usable first standards, not a comprehensive AI rulebook.

Opening prompt

"Take one practice that survived our lifecycle evaluation. What must remain consistent to protect the engineering outcome, what can remain flexible, and what evidence justifies the decision?"

For each practice, ask:

- What reliable leverage did Episode 2 demonstrate, and under what conditions?
- What engineering outcome must remain consistent even if the developer's technique changes?
- What part is guidance or a useful example rather than a requirement?
- What does the organization protect by creating this expectation?
- Should the practice be experimental, recommended, or required - and why?
- Who owns it, when should it be reviewed, and where should engineers encounter it in their normal workflow?
- What evidence would justify changing or retiring it?

Examples of outcomes that may justify consistency

Information: protected, proprietary, regulated, or customer information is handled within defined boundaries.

Engineering integrity: AI-assisted changes respect architecture, quality expectations, testing, and maintainability.

Accountability: the engineer accepting AI-assisted work understands it and remains responsible for the result.

Examples that may remain guidance rather than standards

- Prompt structures that help establish system context.
- Useful sequences for asking AI to analyze before generating a change.
- Examples of verification questions or review techniques that experienced engineers find effective.

Session notes

AI Engineering Standard Worksheet

Complete one copy for each candidate. This worksheet turns Episode 2 evidence into a concise engineering expectation while preserving room for better techniques.

Practice / use case	
Episode 2 evidence source	
Protected engineering outcome	
What must remain consistent	
What may remain flexible	
Why this standard exists	
Classification	Experimental Recommended Required
Owner / accountable role	
Review date / trigger	
Where the expectation appears in workflow	

Standard quality check

Criterion	Evaluation question	Yes	Revise
Evidence	Is the expectation supported by Episode 2 evidence rather than enthusiasm or preference?	▪	▪
Outcome	Can we state clearly what organizational or engineering outcome it protects?	▪	▪
Scope	Have we avoided prescribing technique that can safely vary?	▪	▪
Conditions	Have we preserved the context, judgment, and validation required for success?	▪	▪
Proportion	Is the classification no stronger than the risk or consequence requires?	▪	▪
Usability	Will engineers encounter the expectation where the work occurs?	▪	▪
Ownership	Is someone responsible for reviewing and changing it?	▪	▪
Challenge	Is there a clear way to propose a better approach?	▪	▪

Standard Improvement / Challenge Log

Purpose: Use this worksheet when an engineer or team believes an existing AI engineering standard can be improved. The goal is to allow better practices to emerge without abandoning the standards that are already protecting the organization.

Standard being challenged or improved:
Write the current recommended or required practice here.

Proposed improvement:
Describe the new approach you would like to test.

Why do you believe it may be better?
What do you expect it to improve—for example, engineering time, quality, consistency, maintainability, or cost?

What must not get worse?
Identify the important outcome the current standard is protecting. For example, if the standard exists to protect code quality, a faster alternative cannot be considered better if it produces lower-quality code.

Test the Proposed Improvement

Try the alternative on representative engineering work and compare it with the current practice. Record:

- Did it save engineering effort?
- Did it produce an equal or better result?
- Did it introduce additional review, rework, risk, cost, or technical debt?

Decision

After reviewing the evidence, select one:

- **Keep the current standard** — The proposed approach did not demonstrate enough improvement.
- **Update the current standard** — The experiment revealed a useful improvement that should be incorporated into the existing standard.
- **Replace the current standard** — The new approach produced a better result and should become the new recommended or required practice.
- **Continue experimenting** — The idea shows promise, but there isn't enough evidence yet to change the standard.

Why did we make this decision?
Record the evidence that supported the decision.

What needs to be updated or communicated?
If the standard changed, identify the handbook guidance, development

instructions, AI context, templates, workflows, or team communications that also need to change.

Communication / workflow updates required

That last item—**"Communication / workflow updates required"**—can be confusing. It simply means: If we change the standard, where else do we need to make that change so developers and AI are actually working from the new standard?

For example, if the old standard appears in your engineering handbook **and** in the context supplied to an AI coding assistant, changing the handbook but forgetting the AI instructions means the AI will continue following the old rule.

What to Retain for the Handbook

Episode 3 should convert selected Episode 2 evidence into shared practice without discarding the evidence or flexibility that produced it. Retain the following artifacts so later modules can build on the standards rather than recreate them.

- **First AI Engineering Standards** - concise expectations describing what must remain consistent.
- **Protected Outcome / Reason Statements** - the 'we do this because...' explanation attached to each standard.
- **Standards Classification Register** - experimental, recommended, or required status for each reviewed practice.
- **Guidance and Examples** - prompts, workflows, review questions, and techniques that help engineers satisfy the expectation without being confused with the standard itself.
- **Standard Owners and Review Triggers** - accountability for keeping each expectation current.
- **Workflow Placement** - where each expectation is surfaced in code review, onboarding, team guidance, or another engineering process.

- **Challenge / Improvement Log** - proposed alternatives and the evidence used to retain, revise, or retire standards.

Episode 3 maturity checkpoint

- You are ready to move forward when you can reasonably say:
- We have converted a small number of proven practices into clear shared expectations.
- We can distinguish a standard from the guidance or personal technique used to satisfy it.
- We use experimental, recommended, and required deliberately rather than making every good idea mandatory.
- Each standard states what it protects, has an owner, and has a review mechanism.
- Engineers have a visible, evidence-based way to challenge and improve current standards.
- Our standards are close enough to engineering work to influence behavior rather than simply exist in a document.