

A Lean Framework for Education-Led Enterprise AI Transformation

Advancing AI Adoption in Software Engineering through Team Learning and Central Governance

Marco Fasanella

[LinkedIn](#)

October 2026

Abstract

Enterprise adoption of AI in software development is often treated as a matter of giving employees access to tools, with the expectation that delivery velocity and productivity will naturally improve. Yet access alone does not explain whether developers understand, critically evaluate or effectively integrate AI into their work.

This white paper examines how Lean Thinking, structured education and governance can support responsible and measurable AI adoption in an enterprise software-development context. It combines a practical onboarding programme with anonymous PRE/POST developer surveys to examine signals of use, perceived benefit, verification effort and dependency risk. It also proposes shared ways of working, responsible AI governance and delivery measures to guide continued adoption.

Drawing on multidimensional developer productivity literature, especially SPACE, and delivery performance practices from DORA, the paper frames AI adoption as an organisational intervention rather than a simple productivity accelerator. Its contribution is a practical methodology for understanding and supporting conscious, sustainable AI use. For engineering managers and transformation leaders, it offers a structured basis for connecting education, team practices and central governance in enterprise AI transformation.

Contents

1	Introduction	3
2	Background and Related Work	4
3	Organisational Context and Intervention	5
4	AI Onboarding and Adoption Framework	6
5	Questions for Enterprise Transformation	11
6	Measurement Model & KPI Framework	11
7	Reading the Programme Evidence	15
8	Findings from the Four-Wave Programme	18
9	Discussion	22
10	Practical Recommendations	24
11	Conclusion	27
A	Survey Blueprint and Retained Item Inventory	30
B	Quantitative KPI Clusters for Delivery Measurement	33

1 Introduction

Giving developers an AI assistant does not by itself change how a team delivers software. People still need to decide what to ask, which context to supply, how to check the answer, and who takes responsibility for it. Without that preparation, use can remain uneven: some developers abandon the tool, while others accept its output too readily. Faster task completion, where it occurs, is only part of the change.

This white paper examines an enterprise onboarding programme based on Lean Thinking. Its premise is that education provides the foundation for AI transformation. Critical thinking cannot be outsourced: employees must learn to validate and supervise AI-generated artifacts against shared standards and clear acceptance criteria. Tools change quickly; an understanding of AI fundamentals, limitations, context and verification can help teams make sound choices as those tools change. The programme also treats documentation as essential: business requirements, architecture, repository conventions and delivery rules give both developers and AI systems the context they need. A shared language is necessary for people and AI systems to collaborate effectively and create value together. The evaluation asks what signals of use and understanding appeared after onboarding.

The proposed approach starts with the team's existing work. Understand its needs and constraints, agree what is worth changing, and begin with a small pilot. Developers retain responsibility for every AI-assisted delivery artifact while they learn how to guide and assess the tool. Internal champions help colleagues apply the training and raise problems that a central programme team might miss. Over time, usable project context, shared standards and sound judgement may allow teams to delegate more work to agents within an AI-enabled software development lifecycle (SDLC).

The programme comprised four onboarding cohorts, also referred to as waves. Their survey responses describe reported use before (PRE) and after (POST) onboarding. Three findings give enterprise leaders a practical starting point:

Findings for transformation leaders

- **Reported use broadened.** The two highest usage bands rose from 37/163 responses (22.7%) PRE to 84/127 (66.1%) POST, and 27 of the 28 wave–task mean comparisons increased. Organise education around work activities, not licence activation alone.
- **Context-rich activities deserve explicit attention.** Documentation and unfamiliar-code exploration showed the largest pooled coded task increases; documentation ranked first in every cohort. Include these activities in practical learning and reusable examples, without assuming they deliver the highest financial return.
- **Adoption needs continuing support.** Validation effort increased in every cohort; critical evaluation and reliance when unsure both rose. Teach testing, rejection and escalation alongside effective use, and assess review burden as part of value.

These are unpaired aggregate self-reports, not individual transitions or measured productivity gains. The full findings retain cohort differences and mixed risk signals; response counts and comparability are explained in Section 7.1.

The distinction between use and value matters. Developer productivity involves more than output volume or speed (Forsgren et al. 2021), and AI's effects depend on the task, how people use it and the support around them (Houck et al. 2025; Choudhuri et al. 2026).

Long-term programme question

Under what conditions can structured education support broader, more deliberate AI use while sustaining delivery performance, engineering quality, job quality, and responsible human judgement?

The following sections describe the onboarding approach and the four-cohort survey findings, then set out recommendations for governance and key performance indicators (KPIs). Usage, perceived value, validation effort and dependency risk are considered separately. The intended readers are engineering managers, transformation leaders and practitioners responsible for enterprise AI adoption.

2 Background and Related Work

2.1 AI as a disruptive change in software delivery

AI-assisted development can alter how developers search for information, write code, generate tests, document systems, debug issues, and learn unfamiliar repositories. Controlled studies of AI coding assistants have shown measurable task-completion improvements in selected settings (Peng et al. 2023). However, these studies do not imply that every enterprise team will experience a uniform productivity gain. Effects depend on task type, repository context, skill, review practices, testing discipline, and the surrounding delivery system.

Introducing an assistant changes how people work, collaborate and learn, as well as the risks they must manage. Evaluating it therefore requires both developers' accounts of their experience and observations of the work delivered.

Measured outcomes can diverge from perceived benefits. METR's early-2025 randomised study of 16 experienced open-source developers completing 246 tasks found that AI access increased completion time by 19% in that setting, although developers believed it had accelerated their work (Becker, Rush, Barnes, et al. 2025). METR's February-2026 follow-up reported that selection effects and difficulties measuring concurrent agent work made its later estimates unreliable for quantifying current productivity effects (Becker, Rush, Cunningham, et al. 2026). Together, these studies motivate measuring accepted output and total effort alongside perception.

DORA's 2025 report describes AI as amplifying existing organisational strengths and weaknesses (DORA 2025). Education, context and organisational support are already recognised concerns. This case applies them through a common onboarding curriculum and examines the resulting pattern of reported use, benefit and burden across four cohorts. It then proposes how teams could investigate those findings and assign follow-up actions.

2.2 Developer productivity as a multidimensional construct

At a high level, the SPACE framework treats developer productivity as a balance across five dimensions: Satisfaction and well-being, Performance, Activity, Communication and collaboration, and Efficiency and flow (Forsgren et al. 2021). This is especially relevant for AI adoption because it allows evaluation

to include perceived cognitive effort, flow, documentation, and exploration of unfamiliar code alongside delivery throughput.

Recent AI-specific work applying the SPACE lens reports that developers often perceive AI as improving productivity, efficiency, satisfaction, and routine-task support, while the collaboration effects are more nuanced (Houck et al. 2025).

Self-reports and operational records answer different questions. Surveys describe perceived usefulness, satisfaction, frustration, trust and confidence. Delivery records can show throughput, cycle time, rework, review effort, defects and reliability. Perceptions can overstate impact; operational comparisons can also mislead when work, capacity or estimation practices change. This programme has aggregate survey evidence. Comparable delivery records remain a next step.

Other studies show why satisfaction and perceived effort deserve attention. GitHub’s Copilot research reports that 60–75% of users felt more fulfilled, less frustrated, or able to focus on more satisfying work, while 73% said Copilot helped them stay in flow and 87% said it preserved mental effort during repetitive tasks (Kalliamvakou 2022). SPACE-based AI research reports that 71% of AI adopters believe AI helps them deliver customer or business value, and 62% say it improves job satisfaction (Houck et al. 2025, p. 5).

3 Organisational Context and Intervention

The programme description draws on the organiser’s delivery account; survey findings are presented separately.

Table 1: Programme overview and evidence limitations

Aspect	Description
Setting	An internal R&D experiment involving enterprise software-development teams with different delivery practices. Four onboarding waves took place between February and July 2026.
Starting point	AI-tool subscriptions were already available. The organiser observed uneven installation and use; these were not systematically measured.
Programme delivery	Teams from the same product area were onboarded in batches. The organiser reports that all four cohorts received the same topics and slide materials. Individual attendance and completion are unverified.
Proposed shared practices	A shared Lean Standard of skills, contracts and practices was proposed.
Cohorts and evidence	Anonymous PRE and POST survey responses were collected across four cohorts. Individual responses could not be linked between surveys; the four waves do not track the same group over time.
Programme timing	The organiser reports approximately three weeks of onboarding and about two months between the PRE and POST surveys, including onboarding.

A common training curriculum was delivered across the four cohorts. The framework also sets out recommended practices for project context, team support, measurement, and governance; implementation of each practice across all cohorts is not independently documented.

3.1 Baseline context

The initiative began with a practical problem: subscriptions were available, but teams had no shared approach to using AI. Initial programme observations indicated that some developers had not incorporated the assistant into daily work, while others needed guidance on task framing, context and verification. These observations prompted the programme; they were not a measured baseline. Onboarding combined practical exercises with an explanation of what AI can do, where it fails and why people remain responsible for its output.

3.2 Team composition and delivery diversity

Participating teams used Agile, waterfall-oriented and hybrid delivery practices. They did not have to replace their methodology to join the programme. The proposed transition lets teams learn within familiar workflows, test where assistance is useful, and identify which documentation or practices need to change before delegating more work to agents. Limiting the number of simultaneous changes was part of the design.

3.3 Resistance to change and adoption barriers

Developers approach AI with different expectations and concerns. One external study associated greater software-engineering experience with lower AI use, and prior AI experience with greater use (Choudhuri et al. 2026). The present programme has no demographic evidence with which to examine that pattern. Caution may reflect established habits, but it may also reflect legitimate concerns about quality, security or overstated benefits. Programme leaders should investigate those concerns.

Poor task framing or missing context can make an assistant seem ineffective. Weak validation can create the opposite problem: accepting an answer that should have been questioned. The training addresses both. The survey records reported use, perceived impact, scepticism, validation and dependency risk (Table 6), although it cannot explain why an individual answer changed.

4 AI Onboarding and Adoption Framework

4.1 Lean onboarding philosophy

The programme draws on Lean Strategy and Lean Thinking to avoid committing too early to a tool or delivery model (Christensen 1997; Womack and Jones 2003). It starts with capabilities that can outlast a product configuration: understanding AI and maintaining the project context needed to use it well.

Testing needs to be considered from the start. Projects differ in technology and risk, so they need different checks. Developer research identifies testing and documentation as useful areas for AI support (Choudhuri et al. 2026), but generated tests and documents still need to be checked against the project's acceptance criteria.

In practice, Lean means starting with the team’s existing work, making a small useful change and learning from the result.

Discovery establishes needs, constraints and current practices. The team and accountable leads agree what can change and what should remain, within centrally governed standards and documented approval for exceptions. Figure 1 shows the sequence.

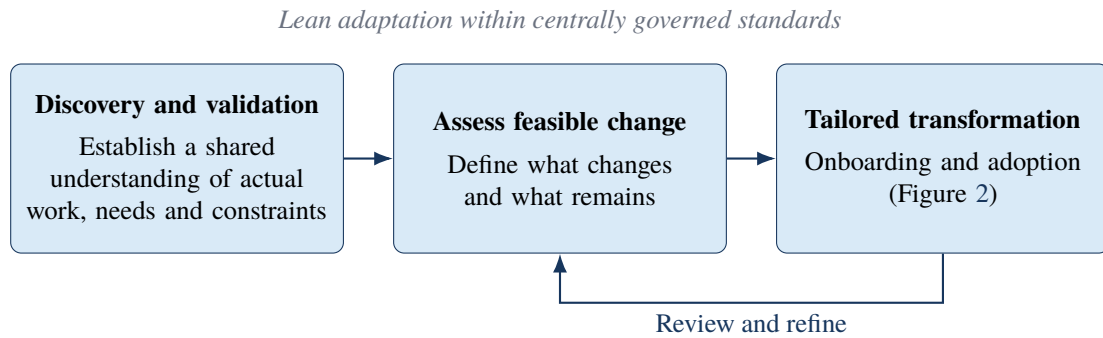


Figure 1: Proposed discovery and assessment before transformation. The team validates the understanding of its current work before assessing feasible changes and agreeing whether and how to proceed. Deviations from the applicable standard require documented justification and exceptional approval by the AI platform owner. The detailed onboarding and adoption pathway is shown in Figure 2.

4.2 Transition phase and target AI SDLC

A team cannot sensibly delegate work to agents before it knows how to guide them, assess their output and recognise their limits. The proposed transition therefore starts with assisted work under human review.

Following the conductor and orchestrator modes described by Osmani (Osmani 2026), this white paper treats the transition phase as a *conductor approach*: developers work directly with AI agents, frame the task, provide context, guide prompts and corrections, inspect intermediate outputs, and validate the final artifact before it becomes delivery evidence. The source describes complementary modes of work; their use in a staged adoption pathway is the design proposed here.

During this phase, teams document business requirements, architectural decisions, domain constraints, repository structure and delivery rules. Making that context usable by colleagues and agents is part of the work, alongside learning to judge the answers AI produces.

Each AI-assisted artifact has a named human reviewer. This includes code, tests, documentation, analysis, prompts, backlog refinements and architectural notes. Review is also a learning opportunity: developers encounter the tool’s limits while deciding whether its output is fit for use.

The POST survey takes place approximately two months after PRE, including around three weeks of onboarding, while the transition phase is already ongoing. In Figure 2, POST is a measurement checkpoint rather than the start or completion of transition. The proposed follow-up and readiness review remain within the transition phase, using repeated surveys and KPI review to identify team issues and recalibrate practices.

The *orchestrator approach* was not implemented or evaluated in the programme reported here. It is a possible future AI SDLC in which humans define goals, constraints, quality gates, and acceptance evidence, while agents coordinate larger parts of the lifecycle under explicit governance. Progression from continued human-directed assistance requires sufficiently usable and validated project context and delivery rules, local value evidence, quality gates and governance controls. Human responsibility remains as work shifts from fine-grained direction to governed orchestration (Osmani 2026). Follow-up continues under human direction while these prerequisites remain unmet, or where task risk or data restrictions make greater delegation unsuitable.

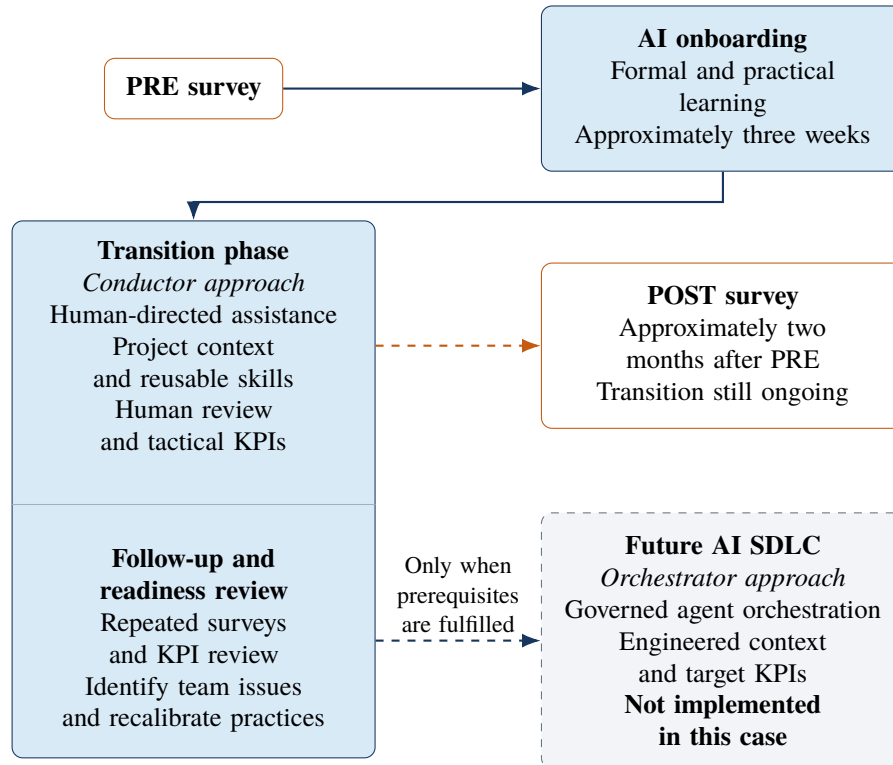


Figure 2: Proposed adoption pathway. The organiser reports around three weeks of onboarding and approximately two months between PRE and POST. The orange dashed arrow marks POST as a checkpoint while transition is ongoing. Follow-up and readiness review remain within the transition phase, with repeated surveys and KPI review to identify team issues and recalibrate practices. Progression to the future AI SDLC requires usable and validated project context and delivery rules, local value evidence, quality gates and governance controls. The dashed blue arrow and grey box identify a possible future extension, not an implemented or evaluated outcome.

4.3 Programme roles and principles

Roles

The AI transformation owner is accountable for the programme and coordinates adoption, training and follow-up. The AI platform owner manages AI tool subscriptions, including Codex where used, integrations and the central governed library of skills, artifacts and plugins. Team leaders provide technical leadership and guide adoption within their teams.

Internal champions are practitioners who help colleagues apply AI to daily work, reinforce safe use and raise problems with the programme team. GitHub’s adoption guidance similarly connects early adopters with more cautious users through repeatable onboarding and practical examples (GitHub 2025). Research also points to the role of organisational support and peer learning in realising value from AI (Houck et al. 2025).

Onboarding took place in batches of teams from the same product area. This was intended to make examples and peer guidance relevant to daily work and encourage shared learning, without ranking individuals or pressuring them to use AI.

Principles

The programme combines education with changes to working practice. Its foundation is shared, while exercises and support are adapted to the team’s assessed needs. Six principles guide that design:

1. **Tool-agnostic AI literacy.** Training should explain how generative AI and large language models behave, what tokens and context windows are, how model choice affects output, where AI helps, and why its output must be validated. Learning these foundations should not require an account or login for a specific AI tool. This shared understanding should help teams learn new AI tools more quickly and underpin a Lean Standard whose practices can be carried forward as tools change.
2. **Human-in-the-loop first.** The transition phase should require human validation of every AI-generated artifact before it is used as delivery evidence, code, documentation, analysis, or decision support.
3. **Context quality.** Documentation, repository structure, workflow clarity, and artifact templates should be treated as inputs to AI effectiveness.
4. **Centrally governed standards.** The AI platform owner centrally owns, approves and versions skills, templates, and lifecycle rules for reuse across teams. Different delivery contexts may require different centrally owned standards. Teams apply the relevant approved standard; deviations require documented justification and exceptional approval by the AI platform owner before use.
5. **Commitment and measurement transparency.** Team leaders should support developers in practising and reviewing AI-assisted work. Developers should understand why data is collected, how it is aggregated, and what will not be used for individual surveillance.
6. **Discover before changing.** Use proportionate team discovery and assessment to understand real tasks, handoffs, pain points, risks, existing controls and readiness before selecting use cases or changing practice. Imposing an AI methodology, framework or tool without prior education can undermine team commitment and morale. Teams may perceive the change as a management demand to measure delivery velocity gains rather than support for improving delivery quality and throughput.

4.4 Training content

According to the programme delivery account, each of the four cohorts received the same onboarding topics and slide presentations. These cover AI and machine-learning foundations; deep learning, generative AI and large language models; context, prompting and agents; tool-specific hands-on work; and role-specific practice. The practical materials include repository documentation and sprint work, manual and automated test cases, requirements analysis, and support-ticket investigation. These are curriculum assignments, not evidence of successful completion or measured learning. The reported onboarding duration is around three weeks.

The programme combines tool-agnostic principles with tool-specific practice. The recommended course in Table 2 provides a reusable extension, including evidence to collect in future delivery.

Table 2: Recommended AI onboarding modules

Module	Purpose	Evidence to collect
AI fundamentals	Build basic understanding of large language models, probabilistic outputs, hallucination, context windows, tokens, model capabilities, model limitations, and cost-aware usage.	Attendance, quiz or knowledge check, training completion.
Prompting and task framing	Teach developers how to state intent, constraints, context, and expected output.	Practical exercise output, peer review notes.
Model and token usage	Explain how to choose an appropriate model for the task, when to use more or less context, how token consumption affects cost and latency, and how reusable prompts or context packs reduce waste.	Model-selection exercises, token-usage examples, reusable prompt or context-pack inventory.
Repository context and documentation	Show how README, architecture, workflow, and local documentation improve AI usefulness.	Documentation readiness checklist, repository context inventory.
Verification and testing	Apply checks appropriate to the artifact and risk: independent acceptance criteria, code and security review, static analysis, relevant tests, and source or domain verification.	Review checklist, requirement links, test and security evidence. Define and implement a project test strategy where one is not already in place.
Safe-use and governance	Explain acceptable use, data handling, confidentiality, human oversight, and escalation.	Policy acknowledgement, risk exceptions, governance logs.
Use-case labs	Apply AI to coding, testing, debugging, documentation, and code understanding tasks.	Lab completion, qualitative feedback, examples of useful and failed outputs.

Exercises should use tasks from participants' delivery environments and may include standardised connectors such as the Model Context Protocol (MCP), reusable skills and shared instructions.

The proposed *Lean Standard* holds these reusable skills, contracts and practices in a versioned set, under the ownership and exception rules described above.

Here, *skills* are reusable task instructions and supporting resources; *contracts* state a task’s inputs, constraints, expected outputs and acceptance evidence. They are working agreements, not a particular vendor feature. Human ownership applies throughout, while review depth should match the artifact’s intended use and risk. Passing AI-generated tests is insufficient when tests and code share a mistaken assumption. Code review and analysis complement executable testing, as distinguished in the NIST Secure Software Development Framework (Souppaya et al. 2022).

5 Questions for Enterprise Transformation

Four practical questions guide the reading of the survey evidence and its implications for the next programme cycle.

1. **Where did reported use expand?** Examine usage bands and the seven development activities, including documentation, code understanding, and testing.
2. **How did perceptions change?** Distinguish perceived productivity, quality, solutions, and effort from validation burden and scepticism.
3. **Which habits need reinforcement?** Read verification, critical evaluation, uncertain reliance, copying, and understanding as separate behaviours.
4. **What should further education address?** Use POST-only training and collaboration responses as follow-up topics alongside the observed PRE/POST patterns.

6 Measurement Model and KPI Framework

6.1 Measurement philosophy

Measure what happens to the team’s work as a whole. Clear requirements, testing and documentation may help a team turn AI assistance into useful output. Where the workflow is unclear, completing one step faster may do little to improve delivery.

The model in Figure 3 groups measures into three layers: knowledge; perceptions and behaviour; and delivery outcomes. Governance applies across all three. The layers specify what to examine, while the KPI groups below organise the decisions that arise during adoption and transition. Appendix B lists candidate operational measures.

The relationships between these layers remain hypotheses. The survey has not validated a maturity sequence. Teams should review the layers together and address gaps in understanding, review practices or support before granting greater autonomy.

1. **AI foundation knowledge:** formal AI literacy, understanding of AI limitations, and knowledge of the fundamentals of the provided AI coding tool.
2. **Perception, behaviour, and barriers:** perceived impact, scepticism, trust, verification, cognitive effort, training adequacy, dependency risk, and adoption barriers.
3. **Operational outcomes:** flow, throughput, review burden, defects, rework, test evidence, and delivery reliability.

The **governance control layer** defines how teams respond to findings. It covers risk records and checks, standard definitions, continuing education, reusable skills and templates, agent controls and responsibility for corrective action.

Risk watchers are agreed checks that trigger review of a specific risk; *agent harnessing* means the permissions, execution limits, logs and approval checks surrounding an agent. Neither term denotes a capability demonstrated by the survey.

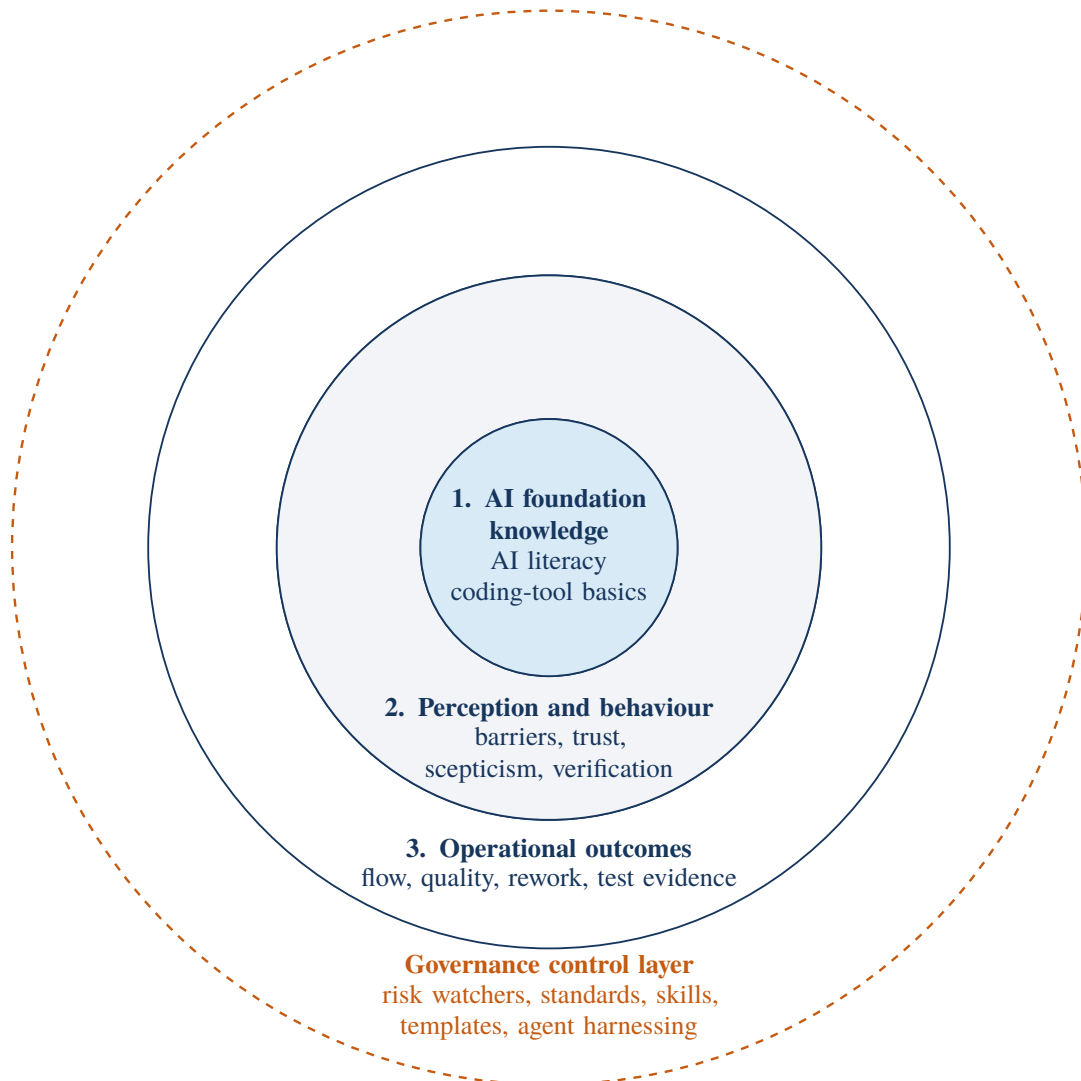


Figure 3: Measurement layers for AI-driven SDLC monitoring.

6.2 Why speed alone is insufficient

Comparing actual delivery lead time in the AI-assisted development lifecycle with earlier estimates has a short window of use as a tactical KPI in early adoption. Completing work sooner than estimated describes performance against a forecast; it does not establish an AI productivity gain. Estimates are not observed counterfactuals. Once onboarding is implemented and AI experience informs planning, estimates incorporate changed expectations, so this comparison no longer provides a stable basis for assessing efficiency gains.

For sustained assessment, the primary efficiency question is how much more accepted, useful work the team can deliver within the same time and available capacity, with quality maintained. Reduced delivery lead time alone does not capture this increase in workload handled. Event-based lead and cycle times remain useful operational measures alongside review effort, defects and rework under comparable workflow definitions. More documentation or tests count as value only when useful and accepted, not merely generated.

The measures need to fit the delivery method. An Agile team may examine committed work, additional accepted work, capacity, scope changes and quality. A waterfall or milestone-based team may use planned work packages, phase deliverables, effort budgets, change requests and acceptance records. The KPI clusters in Appendix B can be adapted to either setting.

6.3 Adoption, tactical, and target KPI groups

Adoption indicators describe tool use, reported understanding, perceived value, verification practices, and training experience. They can identify questions for further assessment. Practical assessments and delivery evidence are needed to evaluate competence, safety, and readiness for greater delegation.

During early transition, tactical KPIs separate forecast reliability from observed delivery performance. Preserve estimates made before AI adoption to describe differences against actual delivery lead time within this limited window; discontinue their use as an efficiency baseline once AI-informed estimation becomes established. Separately compare actual flow and quality where historical records exist. Report changes in work mix, scope, capacity and workflow definitions. Such comparisons describe performance; attribution to AI requires a design that addresses alternative explanations.

As AI becomes part of planning, target KPIs should prioritise additional accepted work delivered within the same time and available capacity, while tracking delivery reliability and quality. Record changes in estimation practices and work mix. Actual delivery lead and cycle times remain informative after transition when their definitions and delivery context are comparable; the limitation concerns comparisons against earlier estimates.

Governance applies to adoption, tactical and target KPIs. It defines who owns each measure, which evidence is required, how privacy is protected and what action can follow. Supporting governance metrics may be useful, but the main task is to make collection, interpretation and decisions accountable.

Table 3: Adoption, tactical, and target KPI groups

KPI group	Baseline logic	Main question	Linked KPI families
Adoption KPIs	Use PRE and POST survey evidence, training records, and usage data.	Which adoption patterns and training needs require further assessment before extending use?	Usage and exposure; perception and culture; safe-use behaviour; training experience.
Tactical KPIs	Compare earlier estimates with actual delivery lead time only in early adoption; retain observed flow and quality separately.	What changed in delivery and forecast reliability before AI-informed estimation became established?	Flow and delivery; quality and testing; usage and exposure context.
Target KPIs	Compare accepted output within the same time and available capacity; record planning and work-mix changes.	Can the team deliver more accepted work in the same time, with quality and reliability maintained?	Flow and delivery; quality and testing.
Governance control layer	Define rules, evidence requirements, ownership, privacy constraints, and auditability standards.	Are KPI collection, interpretation, and interventions trustworthy and responsible?	Applies across Adoption, Tactical, and Target KPIs.

6.4 Workflow measurement consistency

Agree ticket labels, work item types, status definitions and the events used to calculate lead time, cycle time, rework and reopened work. If these definitions vary, an apparent performance change may simply reflect how teams record or move tickets.

6.5 Capacity and work-mix context

Teams differ by size, seniority, skills, domain complexity, and AI experience. Delivery KPIs should therefore be interpreted with capacity and complexity context, using consistent definitions within each team and segmentation by work item type. Story points are local planning conventions, not a common productivity unit for comparing teams. Report changes in capacity, scope, and work mix alongside outcomes, and keep this programme's reporting at team or programme level rather than ranking individuals.

Table 4: From current survey evidence to the next measurement decision

Current evidence	Next measurement and decision	Suggested owner
Reported usage and task coverage	Retain item distributions and counts; investigate which uses produce accepted work before extending them.	AI transformation owner
Perceived benefit and validation effort	Add comparable task, review and correction effort alongside accepted quality; decide whether to continue, adapt or stop a use case.	Team leaders
Understanding, verification and reliance	Add practical review decisions and clearly anchored behaviour items; use the findings to target support and assess readiness.	Team leaders and internal champions
POST-only training and collaboration	Investigate reported needs through exercises and team feedback; define a baseline before making future change claims.	AI transformation owner

Current findings are in Section 8; the next measurements are proposals. Adoption, tactical and target groups describe when these decisions arise (Table 3). Operational KPI clusters are in Appendix B. Governance applies throughout.

7 Reading the Programme Evidence

7.1 Survey scope and interpretation limits

PRE records responses before onboarding and POST records responses after it, within each of the four waves. The workbooks contain 28 six-bin items at both timings, a separate usage-band distribution, and ten additional POST-only items. Reported programme timings are approximately three weeks for onboarding within an approximately two-month PRE-to-POST interval; transition is already ongoing at POST. The four waves are separate programme cohorts, rather than successive measurements of the same people.

The usage question asks, “How often do you use AI coding tools?” The surveyed population consists of software developers, so the question concerns AI use in their daily development work, including coding, code review, documentation creation and functional requirements analysis. Table 5 pools responses from

all four cohorts, showing the response count and percentage in each usage band: 163 responses PRE and 127 POST. Each completed survey represents a distinct participant in the onboarding programme.

Table 5: Reported AI usage bands before and after onboarding, pooled across four cohorts. Bold rows identify the two highest bands and their combined share.

Reported usage band	PRE, n (%)	POST, n (%)
0–20%	45 (27.6%)	16 (12.6%)
21–40%	35 (21.5%)	13 (10.2%)
41–60%	46 (28.2%)	14 (11.0%)
61–80%	25 (15.3%)	33 (26.0%)
81–100%	12 (7.4%)	51 (40.2%)
Two highest bands combined	37 (22.7%)	84 (66.1%)
All responses	163 (100%)	127 (100%)

Shares use the denominator for each timing. The combined row is a subtotal, not an additional category. Percentages are rounded independently.

The two highest bands, 61–80% and 81–100%, together account for 37/163 responses (22.7%) PRE and 84/127 (66.1%) POST, an increase of 43.4 percentage points. This summary makes the shift towards higher reported use visible. The percentages in the table describe the share of responses in each band; the band labels describe reported AI use.

Survey limitations and bias

The surveys were addressed to the population of each wave, but completing either survey, including completing both, was voluntary. According to the programme delivery account, roles and team membership changed during the approximately two months between measurements. These changes, together with voluntary participation at each timing, account for differences in the PRE and POST respondent groups. The aggregate extracts do not quantify the contribution of each reason or link individual responses. The lower POST count therefore cannot be interpreted as a measured dropout rate from a matched panel.

Selective participation may bias the distributions if those who respond differ from those who do not, while role changes may alter the work activities represented. Self-reported use may also be affected by recall and social desirability. PRE/POST differences describe the available responses at each timing; they cannot establish individual improvement or isolate an effect of onboarding.

Training and collaboration questions are available only after onboarding. Appendix A distinguishes the blueprint from the observed evidence.

The programme aims to help people use AI correctly and make deliberate choices about it. The survey approaches that aim through separate questions on understanding, verification, testing, critical evaluation and reliance when correctness is uncertain. POST-only questions address understanding limitations, knowing when not to use AI and using tools correctly. These are self-reports, not demonstrations of competence or a validated combined score. More frequent use therefore has to be read alongside both favourable and adverse behavioural signals.

Cognitive impact is explored through self-reported understanding, reliance on AI instead of independent reasoning, and critical evaluation. Explanation-seeking and unfamiliar-code exploration provide complementary context (Appendix A.3). These items describe perceived effects and reported behaviours; they do not constitute a validated cognitive-impact scale or establish changes in cognitive ability. The profile reuses items from existing survey categories and is not independent additional evidence.

Table 6: Survey domains and interpretation

Domain	Timing	Interpretation
AI usage behaviour	PRE/POST	Usage-band distributions and seven task-use items; reported exposure rather than measured productivity.
Perceived impact	PRE/POST	Separate items for reported benefit, effort, and validation burden.
Scepticism	PRE/POST	Separate usefulness and overhype items; neither is a standalone test of adoption readiness.
Verification and understanding	PRE/POST	Separate self-reports of understanding, verification, and testing before integration.
Trust and critical evaluation	PRE/POST	Items on evaluation, trust, perceived correctness and reliance under uncertainty; not a validated calibration score.
Reported risk-related behaviour	PRE/POST	Aggregate self-reports of risk-related practices; reported descriptively without ranking or causal interpretation.
Reliance and code-acceptance practices	PRE/POST	Separate signals for copying directly, relying instead of thinking, and reduced understanding.
Cognitive impact (self-reported)	PRE/POST	Profile of four existing understanding, reasoning, and critical-evaluation items.
Collaboration	POST only	Available POST aggregate only; PRE and difference are N/A.
Training adequacy	POST only	Available POST aggregate only; PRE and difference are N/A.

7.2 Aggregation and interpretation

For each item and timing, counts are pooled across cohorts, weighting responses equally. Task-use and positive-benefit summaries equally weight seven task-use means and four benefit means (productivity, code quality, better solutions, and effort reduction), respectively; neither has a single participant denominator. Burden and scepticism remain separate, with no overall maturity or safety score. Means and differences are displayed to two decimals; differences are calculated before rounding and may differ from subtraction of the displayed means.

All six values are valid survey responses (0–5) included in item denominators and means. Agreement items use 0 = strongly disagree and 5 = strongly agree; task-use items use 0 = never and 5 = always. Published means retain the original coding, so higher agreement with negatively worded items is interpreted as less favourable. Coded differences are descriptive, not validated scale scores or percentage productivity gains.

For task use, a complementary distribution check compares PRE and POST shares at or above each coded threshold, 1 through 5. This checks the direction of the pooled shift without assuming equal distances between categories.

7.3 Evidence and programme experience

Context on prior tool access, education, repository guidance, champions and leadership participation comes from the organiser's observations. These form part of the programme delivery account, rather than survey variables or findings from systematic interviews.

The survey documents changes accompanying onboarding. It cannot isolate the contribution of education from practical support, shared working practices, or other changes during the programme.

7.4 Anonymity, reproducibility and author disclosure

The surveys were fully anonymous to address concerns about surveillance or individual evaluation and encourage candid feedback. No personal identifiers or matching keys were collected. The analysis therefore compares aggregate responses and cannot track individual change between PRE and POST.

The private analysis workbook retains counts, formulas, source hashes and descriptive summaries for all four cohorts. Source workbooks and internal notes are withheld for confidentiality. Published tables and definitions describe the analysis, but the absence of public source data prevents readers from fully reproducing it.

The author organised the internal R&D programme and provides the delivery account; this involvement may influence the selection and interpretation of observations. Analysis and editorial preparation were AI-assisted. The paper presents a practitioner case study, and organisational identities are withheld.

8 Findings from the Four-Wave Programme

Broader reported AI use is the clearest change after onboarding. Respondents also reported greater benefit overall, but more validation work and several signs of dependency. These findings suggest questions for further support: whether teams need more practice, better context, stronger review or limits on unsuitable uses.

8.1 Reported use expanded across development activities

The share in the two highest usage bands rises from 37 of 163 responses (22.7%) PRE to 84 of 127 (66.1%) POST, a difference of 43.4 percentage points. Reported AI use increased across nearly all development activities in the four cohorts. Of the 28 comparisons, 27 showed higher mean responses after onboarding.

Documentation and exploration of unfamiliar code showed the largest pooled coded increases. Documentation also had the largest task-use mean increase within each of the four cohorts. Reported use also

Table 7: Programme-wide reported usage and perceived benefit.

Measure	PRE	POST	Change
Two highest usage bands	37/163 (22.7%)	84/127 (66.1%)	+43.4 pp
Task-use summary	2.63	3.51	+0.88
Positive perceived-benefit summary	2.99	3.37	+0.38

All four waves. pp = percentage points. Summaries equally weight the seven task items or four positive-benefit items after pooling counts within each item. Coded means are descriptive, not validated scales. Differences use unrounded means.

Table 8: Reported AI use across software-development activities.

Activity	PRE <i>n</i>	POST <i>n</i>	PRE	POST	Change
Documentation	156	114	2.40	3.81	+1.41
Unfamiliar code	155	112	2.75	3.84	+1.08
Tests	149	113	2.23	3.14	+0.91
Explanations	160	118	3.09	3.94	+0.85
Debugging	152	112	2.64	3.48	+0.84
New code	151	109	2.56	3.13	+0.57
Existing code	150	109	2.75	3.21	+0.46

All four waves; source-coded means on 0–5 bins. Ordered by pooled mean change. 27 of 28 wave–task comparisons increase; these are not independent significance tests.

increased for testing, explanations, debugging, code generation, and modification. These activities are useful candidates for practical training.

For all seven pooled task-use items, POST shares were higher at every coded threshold from 1 through 5. The upward pattern therefore extends beyond the mean differences; it describes the available response distributions, not individual change.

8.2 Perceived benefits increased, with additional validation work

All four positive-benefit items increase in the pooled responses, with the largest changes concerning better solutions and code quality. The positive-benefit summary rises from 2.99 to 3.37. These are changes in perceptions, not observations of delivered quality or output.

Table 9: Perceived benefits, effort and reservations after onboarding.

Item	PRE <i>n</i>	POST <i>n</i>	PRE	POST	Change
AI improves my productivity	154	117	3.25	3.56	+0.32
AI improves code quality.	146	112	2.63	3.16	+0.53
AI helps me find better solutions.	156	118	3.00	3.53	+0.53
AI reduces effort.	152	119	3.09	3.23	+0.14
AI sometimes creates more problems than it solves	154	115	2.22	2.10	-0.12
Using AI requires additional validation effort	152	116	2.97	3.58	+0.60
I am skeptical about AI usefulness	151	117	1.15	1.24	+0.09
AI tools are overhyped	154	116	1.95	1.75	-0.20

All programme waves. Means retain original coding 0–5. Changes are descriptive and samples are unpaired.

Perceived code quality was the only positive-benefit item whose mean increased in every cohort.

Additional validation effort rises in every cohort (+0.60 pooled). Trying AI more often and experiencing less work are different outcomes. An assessment of value should include reviewing, correcting, testing, and integrating output.

Usefulness scepticism rises slightly in pooled coded responses (+0.09), while the perception that AI is overhyped declines (-0.20). Attitudes did not move uniformly towards enthusiasm.

8.3 More critical evaluation alongside greater uncertain reliance

Reported critical evaluation increases in every cohort (+0.58 pooled), and direct copying declines in every cohort (-0.43 pooled). Increased reported evaluation aligns with the programme’s emphasis on active human judgement; less direct copying alone does not establish safer acceptance.

Reliance on AI when unsure of correctness also increases in every cohort (+0.39 pooled). Other pooled indicators concerning unreviewed acceptance, reliance instead of independent reasoning, reduced understanding, and self-reported bugs increase.

Table 10: Verification, trust and dependency indicators.

Item	PRE <i>n</i>	POST <i>n</i>	PRE	POST	Change
I understand the AI-generated code I use.	144	112	3.93	3.99	+0.06
I verify AI-generated code before I use it	145	111	4.30	4.50	+0.20
I test AI-generated code before I integrate it	141	111	4.04	4.41	+0.36
I trust AI-generated code without modification	144	106	1.13	1.42	+0.30
I critically evaluate AI suggestions before accepting them	145	110	3.42	4.00	+0.58
AI-generated code is often correct on the first try	143	107	2.09	2.60	+0.51
AI-generated code often contains subtle issues	142	104	2.54	2.62	+0.08
I rely on AI even when I am unsure that the result is correct	146	109	1.35	1.74	+0.39
I accept AI suggestions without fully reviewing them	148	114	0.91	1.03	+0.12
I copy AI-generated code directly into working code	145	109	1.68	1.26	-0.43
I rely on AI instead of thinking through the problem myself	149	113	1.16	1.36	+0.20
AI reduces my understanding of the code I am working with	146	108	1.46	1.70	+0.24
I sometimes introduce bugs because of AI suggestions	145	107	1.37	1.61	+0.24

All programme waves. Means retain original coding 0–5. Changes are descriptive and samples are unpaired.

Verification and testing rise in the pooled data, but not uniformly across cohorts. Recognising the need to review an answer is not the same as resolving uncertainty about it. Practical education should include rejecting unsupported output, testing assumptions, and seeking help when correctness cannot be established.

The cognitive-impact items suggest a mixed pattern (Appendix A.3). Self-reported understanding changes little, while more frequent explanation-seeking and unfamiliar-code exploration suggest greater engagement with code and its interpretation. This may create opportunities for learning, particularly when AI use is accompanied by critical evaluation and independent reasoning.

8.4 Beyond the mean: towards more nuanced perceptions of AI

Pooled means can conceal greater concentration in intermediate responses, as illustrated by the items in Table 11.

Table 11: Greater concentration in intermediate categories on two AI-perception items.

Survey item	PRE / POST <i>n</i>	Mean	Categories 0–1	Categories 2–3	Categories 4–5
AI-generated code often contains subtle issues	142 / 104	2.54 → 2.62	23.9% → 16.3%	48.6% → 60.6%	27.5% → 23.1%
I trust AI-generated code without modification	144 / 106	1.13 → 1.42	69.4% → 58.5%	23.6% → 35.8%	6.9% → 5.7%

All four cohorts pooled. Values show PRE → POST. Categories retain source coding 0–5; percentages use each item’s observed denominator. Samples are unpaired.

For both items, intermediate responses become more common while both outer bands decline. The subtle-issues item demonstrates how a nearly unchanged mean can conceal this redistribution; trust shows a related pattern alongside a larger mean increase. Practical experience may encourage more nuanced judgments, as users encounter both useful outputs and limitations and become more selective about which results to trust. This may be an encouraging signal of a more balanced, context-sensitive understanding of AI’s capabilities and limitations. Changes in respondent composition or interpretation of the questions remain alternative explanations.

8.5 Training and collaboration identify follow-up topics

The ten items are administered after onboarding because assessing training sufficiency, informed tool use, and effects on collaboration requires direct experience of AI tools and the programme. Self-reported understanding of limitations has a pooled coded mean of 4.05, training sufficiency 3.72, and confidence in complex or critical tasks 3.37. Different wording and coverage prevent interpreting these differences as a validated training gap or proof of competence.

Table 12: Training and collaboration: self-reports after onboarding.

Item	POST <i>n</i>	Coded mean
AI improves my ability to communicate technical ideas	105	3.09
AI improves the quality of documentation I produce	109	3.82
AI reduces the need to ask colleagues for routine help	107	3.29
AI-generated code is harder for others to review	100	2.42
AI reduces knowledge sharing within the team	106	2.44
I received sufficient training to use AI tools effectively	106	3.72
I understand the limitations of AI tools	108	4.05
I know when I should not use AI tools	109	3.81
I feel confident using AI in complex or critical tasks	106	3.37
I am using AI tools correctly in my daily development work	102	3.86

All four waves; source coding 0–5. Items assess experience of AI tools and onboarding and support ongoing follow-up. Self-reports do not establish competence or a validated training gap.

Collaboration items concern technical communication, documentation quality, routine requests for help, review difficulty, and knowledge sharing. Perceived documentation quality is prominent in these POST responses, complementing increased documentation use in the questions available at both timings. Reduced need for routine help should prompt discussion of which interactions remain valuable; it does not itself show improved or diminished collaboration. These ten items should now be repeated in continuous

onboarding follow-up within teams and across the programme population, guiding subsequent exercises, team conversations, and training support.

8.6 Different experiences within one programme

Table 13: Variation across the four programme waves.

Wave	PRE <i>n</i>	POST <i>n</i>	Usage bands	Task use	Benefits
O1	17	14	+63.4 pp	+1.54	+0.69
O2	11	16	+7.4 pp	+0.72	-0.21
O3	52	46	+56.9 pp	+1.08	+0.71
O4	83	51	+35.2 pp	+0.49	+0.07

Counts refer to usage-band responses. Last three columns are POST minus PRE. Task use and benefits are equal-item coded summaries; waves are cohorts, not successive observations of one panel.

Cohort O2 shows increased task use, a smaller shift towards higher usage bands and a decrease in the positive-benefit summary. Its usage counts were 11 PRE and 16 POST; these unlinked samples do not establish individual changes. Possible explanations and follow-up questions are discussed in Section 9.5.

9 Discussion

9.1 Implications of broader use for practical training

The broader reported use of AI in Section 8 supports extending practical training beyond code generation to documentation, code exploration, explanation, testing and debugging. Exercises should emphasise task framing, context provision and answer validation.

9.2 Perceived value changed, but not uniformly

The pooled benefit scores conceal different experiences. Effort reduction fell in O2 and O4 even as task use increased, while perceived code quality increased in every cohort. Local reviews with participants and team leaders should distinguish quality benefits from effort savings, examining which tasks became easier and which required more correction or integration.

9.3 From scepticism to calibrated use

Perceived benefits and critical evaluation (+0.58) increased alongside slightly higher usefulness scepticism (+0.09) and lower perceived overhype (-0.20). One possible interpretation is that respondents recognised benefits while questioning usefulness in particular tasks.

Unlinked responses cannot show that the same people became more informed, found greater value and became more critical. Reliance when unsure also increased. The results therefore leave open whether reported scrutiny translated into better decisions.

9.4 Human validation and dependency risk

Validation effort rose in every cohort. Some of that work may be the checking expected during the transition; some may be avoidable correction. Reviewing tasks with the teams would help establish where that effort went.

More critical evaluation and less direct copying were reported alongside more reliance when unsure. These patterns may concern different respondents. Training should address how to resolve uncertainty and when to ask for help. Keeping the behaviours separate makes those needs easier to examine than a single safety score would.

These items are not measures of incidents or error rates per AI-assisted task. Broader use may create more opportunities for review, uncertainty and errors, so the survey cannot separate changing exposure from changing behaviour per opportunity. Direct copying or accepting unchanged code is not inherently unsafe when appropriate validation has occurred; those items do not establish whether acceptance was justified. Future assessment should combine clearly anchored behaviour questions with review decisions, task exposure and accepted-quality evidence.

The accountable owner of each KPI is responsible for monitoring it and interpreting its meaning within the specific team's context. KPIs should be read in pairs or groups with complementary indicators to support a correct interpretation of the data. Given the diversity of teams, no single KPI should serve as a uniform standard for assessing all teams.

9.5 Understanding differences between cohorts

Cohorts O2 and O4 illustrate why pooled perceptions require local follow-up. O2 reported a decrease in the positive-benefit summary despite increased task use. O4 reported a small positive-benefit change alongside improvements in several risk items and a lower POST response count. The four-cohort comparison in Table 13 preserves these differences without treating any cohort as a control group.

According to the programme manager, commitment was low among O2's team leaders and the person accountable for transformation. This retrospective observation was not systematically measured. Leadership support is one hypothesis to investigate alongside task fit, practical support, baseline conditions, exposure timing and respondent composition. With 11 PRE and 16 POST usage responses, the cohort comparison cannot distinguish these explanations.

Champions and team leaders can record barriers and support provided during follow-up, so that future reviews can examine these explanations against task context and response coverage.

9.6 Documentation, context, and the path to an AI SDLC

Documentation, unfamiliar-code exploration and explanations showed larger coded increases than new-code generation or code modification. These activities warrant further training, although the survey does not show that project context improved or caused better outcomes.

The proposed conductor phase uses requirements, architecture, repository conventions and acceptance criteria to guide assisted work (Section 4.2). Whether to continue direct assistance or consider orchestration depends on task suitability and controls; usage volume cannot decide that question.

9.7 Implications for the next measurement cycle

The next cycle should retain separate measures for task use, perceived value, validation burden, scepticism and distinct dependency risks. Choose a documented repeated-cross-section design or, under a separate approved protocol, privacy-safe respondent linkage; keys must remain outside publication exports. Record stable scale anchors, response-rate context and limited delivery/exposure fields. POST-only items describe experience, not improvement. Add operational flow and quality evidence only when definitions, ownership and lineage are validated.

10 Practical Recommendations

10.1 Preserve education as the durable foundation

Teach AI fundamentals, limitations, context and verification before focusing on a particular product. Give participants work from their own setting on which to practise, and retain useful instructions, contracts and delivery rules in the versioned Lean Standard. These can be revised and reused as tools change.

10.2 Use a staged transition

Retain human ownership during transition and continued follow-up. Use follow-up to review readiness, and consider orchestration only once project context and delivery rules are usable and validated, local value evidence supports it, and task risk, data restrictions, quality gates and governance permit it.

For an enterprise applying this approach, the recommended sequence is:

1. **Discover the team's actual work.** Use the discovery phase in Figure 1 to map tasks, handoffs, pain points, tools, controls and context gaps. Check this account with team members and leaders against available artifacts.
2. **Assess and agree the starting point.** Reuse suitable standards or agree a tailored set for piloting. With the team and accountable leads, identify opportunities, risks, readiness gaps, success signals, ownership and the smallest useful change.
3. **Build a shared foundation and adapt it locally.** Teach capabilities, limitations, context and accountability through representative tasks. Preserve effective practices and develop any new standards with the team and AI platform owner.
4. **Pilot, validate and learn.** Test the standards in actual work, recording accepted value, review effort, quality and risk. Use the findings to refine or stop the pilot and identify practices suitable for reuse.

5. **Extend only when evidence supports it.** Resolve material issues across usage, perceived value, validation burden and risk before extending the pilot or granting greater autonomy. Submit validated new standards for central approval, ownership and versioning before reuse by teams with compatible needs.

Keep discovery proportionate to complexity and risk. Deviations from an applicable centrally governed standard require documented justification and approval by the AI platform owner before implementation.

10.3 Set minimum controls before expanding permissions

Table 14 is a proposed local decision checklist, not an assurance result from these surveys. NIST’s voluntary Generative AI Profile informs the provider, supplier, incident and fallback provisions (National Institute of Standards and Technology 2024). The accountable AI transformation owner should require the named owners’ approval before extending a pilot; reviewers should be able to halt a task when its acceptance conditions are unmet.

Table 14: Minimum controls for an AI-assisted development pilot		
Control	Suggested owner	Required decisions and practices
Tools, data, integrations and configuration	AI platform owner	Select approved tools, permitted data and integrations; define connector and agent access, execution limits and logging. Review and approve model, configuration and permission changes, with checks of affected behaviour before enabling them.
Skills and artifact governance	AI platform owner	Enable teams to create skills and artifacts for project needs. Centrally review, approve, version and maintain the applicable sets of skills, artifacts, plugins and best practices, with clear ownership and conditions for reuse.
SDLC checks, error handling and fallback	Team leaders, with AI platform owner support	Define checks and acceptance criteria at the relevant SDLC steps. Review outputs against requirements and tests; stop and escalate unresolved errors. Record failures, apply rollback or manual fallback, and improve project context or agent instructions before retrying and validating the correction.

10.4 From discovery to reuse or a new standard

The following **illustrative exercise** shows how discovery connects a team’s needs to the Lean Standard. When approaching a new team, the adoption team identifies relevant use cases and examines each task’s purpose, required inputs, expected outputs, constraints and acceptance criteria. This assessment leads to two possible paths.

Adopt an existing standard. Suppose the team needs to produce maintenance documentation for a repository. If the Lean Standard already covers this task and provides an applicable task contract (as defined in Section 4.4), reusable skills and agent configuration, the adoption team proposes that standard. Together, they confirm that it fits the team’s needs, supply the required project context and validate the outputs against its acceptance criteria. The team can then adopt the existing practice.

Develop a new standard. If the task is new, or its requirements differ materially from those covered by an existing standard, discovery continues with a closer examination of the work. The team clarifies the

task's needs, inputs, outputs and constraints, then defines a contract specifying how the work should be performed and what evidence is required for acceptance. That contract guides the creation of supporting artifacts, including reusable skills and agent configurations. The team pilots the practice on actual work, reviews the results and refines both the contract and its artifacts.

Once validated, the new practice is submitted to the AI platform owner for central review, approval, ownership and versioning. It then joins the existing Lean Standard, making it available to other teams with compatible needs. Discovery therefore supports both the adoption of established practices and the development of new ones.

10.5 Invest in capability and assess long-term value

Budget explicitly for teaching, practice, review, correction and renewal of shared artifacts. These are investments in capability whose benefits may emerge over time through reusable practices and project context.

Evaluate AI initiatives on a technology-neutral basis, using net present value (NPV), internal rate of return (IRR) where appropriate, and strategic value. NPV and IRR assess expected cash flows over time (ACCA 2026); strategic assessment should also recognise the longer-term contribution of education and shared capabilities. Make investment decisions on expected value creation, independently of a preference for CapEx or OpEx, then determine the appropriate accounting treatment under applicable standards (IFRS Foundation 2026). Investment in capability is a management perspective, not an automatic basis for capitalisation.

Record learning and support effort alongside accepted output, quality and rework, and revisit the investment case as evidence develops. This programme has no measured financial return; future appraisal should make assumptions and uncertainty explicit.

10.6 Support teams through champions

Establish a champion network across teams, coordinated by the AI transformation owner with area managers and team leaders. Regular exchanges should carry guidance to practitioners and local problems to those responsible for the programme.

Give champions time to demonstrate AI use, support colleagues, collect examples and raise problems. Team leaders and managers should ensure that this support reaches their teams. The AI transformation owner coordinates training and follow-up; the AI platform owner incorporates approved lessons into the Lean Standard.

Commitment from team leaders and internal champions helps turn onboarding into sustained practice. Team leaders should actively guide adoption, follow up frequently and encourage colleagues by recognising learning and useful applications of AI. Champions reinforce this support through practical examples, help with task framing and context, and guidance on validating uncertain output. Together, they help teams build confidence and resolve difficulties as AI becomes part of daily work.

11 Conclusion

This programme established a practical approach to enterprise AI adoption that combines education, discovery of team practices, guided use and centrally governed standards. Across four cohorts, the share of responses in the highest usage bands rose from 22.7% to 66.1%, reported use expanded across development activities, and perceived benefits increased. Reported critical evaluation, verification and testing also increased. These unpaired self-reports show broader reported use after onboarding, alongside mixed signals of informed engagement; they do not isolate the programme's contribution.

Beyond the surveys, the organiser reports an approximately eight-month rollout from February 2026, an accepted methodology and a successful programme, with stronger leadership commitment and a shared AI adoption culture. These are organisational observations. Sustaining the approach requires team support and standards that fit the work, with new practices validated before wider adoption.

The next task is to connect adoption signals with accepted quality, review effort, defects and rework through team-level delivery measures and practical assessment. Follow-up should examine cohort differences and whether increased critical evaluation leads to better decisions, particularly when correctness remains uncertain.

Reported understanding changed little despite broader use. Continued teaching, practice and review therefore remain central to the proposed approach, while informed use and delivery value require further evaluation.

References

- ACCA (2026). *Investment Appraisal*. URL: <https://www.accaglobal.com/uk/en/student/exam-support-resources/prodipsust-study-resources/technical-articles/investment-appraisal.html> (visited on 10/08/2026).
- Becker, J., N. Rush, B. Barnes, and D. Rein (July 10, 2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*. METR. URL: <https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/> (visited on 09/13/2026).
- Becker, J., N. Rush, T. Cunningham, D. Rein, and K. Mahamud (Feb. 24, 2026). *We Are Changing Our Developer Productivity Experiment Design*. METR. URL: <https://metr.org/blog/2026-02-24-uplift-update/> (visited on 09/13/2026).
- Choudhuri, R., C. Badea, C. Bird, J. Butler, R. DeLine, and B. Houck (2026). *AI Where It Matters: Where, Why, and How Developers Want AI Support in Daily Work*. arXiv: 2510.00762 [cs.SE]. URL: <https://arxiv.org/abs/2510.00762>.
- Christensen, C. M. (1997). *The Innovator's Dilemma: When New Technologies Cause Great Firms to Fail*. Boston, MA: Harvard Business School Press.
- DORA (2025). *State of AI-Assisted Software Development 2025*. URL: <https://dora.dev/research/2025/dora-report/> (visited on 09/13/2026).
- Forsgren, N., M.-A. Storey, C. Maddila, T. Zimmermann, B. Houck, and J. Butler (2021). "The SPACE of Developer Productivity: There's More to It Than You Think". In: *ACM Queue* 19.1, pp. 20–48. URL: <https://www.microsoft.com/en-us/research/publication/the-space-of-developer-productivity-theres-more-to-it-than-you-think/>.
- GitHub (Feb. 2025). *Training and Onboarding Developers on GitHub Copilot*. Whitepaper. GitHub. URL: <https://github.com/resources/whitepapers/training-and-onboarding-developers-on-github-copilot>.
- Houck, B., T. Lowdermilk, C. Beyer, S. Clarke, and B. Hanrahan (2025). *The SPACE of AI: Real-World Lessons on AI's Impact on Developers*. arXiv: 2508.00178 [cs.HC]. URL: <https://arxiv.org/abs/2508.00178>.
- IFRS Foundation (2026). *IAS 38: Intangible Assets*. URL: <https://www.ifrs.org/issued-standards/list-of-standards/ias-38-intangible-assets/> (visited on 10/08/2026).
- Kalliamvakou, E. (Sept. 2022). *Research: Quantifying GitHub Copilot's Impact on Developer Productivity and Happiness*. URL: <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/> (visited on 07/10/2026).
- National Institute of Standards and Technology (July 2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. Tech. rep. NIST AI 600-1. Voluntary risk-management guidance. National Institute of Standards and Technology. doi: 10.6028/NIST.AI.600-1.
- Osmani, A. (June 16, 2026). *The New Software Lifecycle*. URL: <https://addyosmani.com/blog/new-sdlc-vibe-coding/> (visited on 09/30/2026).

- Peng, S., E. Kalliamvakou, P. Cihon, and M. Demirer (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot*. arXiv: 2302.06590 [cs.SE]. URL: <https://arxiv.org/abs/2302.06590>.
- Souppaya, M., K. Scarfone, and D. Dodson (Feb. 2022). *Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities*. Tech. rep. NIST SP 800-218. National Institute of Standards and Technology. DOI: [10.6028/NIST.SP.800-218](https://doi.org/10.6028/NIST.SP.800-218).
- Womack, J. P. and D. T. Jones (2003). *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Revised and updated. Free Press.

A Survey Blueprint and Retained Item Inventory

This appendix lists the survey domains and item wording retained in the source workbooks, alongside recommendations for future administration. The workbooks contain question text and six valid response values, 0–5; the organiser has confirmed the scale endpoints described in Section 7. The inventory does not reproduce the complete administered questionnaire. Future surveys should retain wording, anchors and timing so that comparisons can be checked.

A.1 Proposed administration protocol

The following are blueprint recommendations, not confirmed administration dates for the reported programme.

- PRE survey: administer 3 to 5 business days before onboarding, training, or formal rollout.
- POST survey: administer after theoretical and practical onboarding and the approved enterprise reporting period.
- Retain the approved aggregate response distribution for each available measure and wave, with a derived mean only when the response bounds and anchors are validated; do not submit response-level records or linking keys.
- Keep scales unchanged across waves.
- Record an unavailable measure as *Not available*; do not substitute zero.

This aggregate reporting boundary also applies to publication exports. If a future study uses privacy-safe respondent linkage, it requires a separate approved collection and access protocol; linking keys must remain outside those exports. The current anonymous counts cannot be retroactively linked.

A.2 Domains represented in the source workbooks

The table records timing availability in the retained extracts. Its rows organise the observed items and interpretive profiles; they are not a reconstruction of the original questionnaire sections.

Table 15: Survey domains and availability before and after onboarding

Domain	PRE	POST	Purpose
AI usage behaviour	Yes	Yes	Adoption frequency and task-level usage.
Perceived impact	Yes	Yes	Reported perceptions of usefulness at each timing.

Domain	PRE	POST	Purpose
Scepticism and perceived barriers	Yes	Yes	Reservations, concerns, and adoption barriers.
Verification and understanding	Yes	Yes	Reported verification and understanding at each timing.
Trust calibration	Yes	Yes	Overtrust and critical evaluation.
Cognitive impact (self-reported)	Yes	Yes	Profile of existing understanding, reasoning, and critical-evaluation items; not a separate questionnaire section or validated scale.
Risk-related practices	Yes	Yes	Reported acceptance, reliance, and understanding.
Collaboration	N/A	Yes	POST-only collaboration experience; PRE and difference are N/A.
Training adequacy	N/A	Yes	POST-only adequacy of onboarding and residual needs; PRE and difference are N/A.

No respondent-level profile variables are available; anonymous wave labels identify aggregate cohorts only.

A.3 Items informing the cognitive-impact profile

The following four items are present at PRE and POST in the supplied extracts. They provide a descriptive profile of perceived cognitive impact and engagement, reusing items already reported under understanding, critical evaluation, and dependency-related behaviour.

Table 16: Existing items informing the cognitive-impact profile

Dimension	Survey item
Perceived loss of understanding	AI reduces my understanding of the code I am working with
Substitution for independent reasoning	I rely on AI instead of thinking through the problem myself
Self-reported understanding	I understand the AI-generated code I use.
Critical evaluation	I critically evaluate AI suggestions before accepting them

The first two items address perceived loss of understanding and reliance on AI instead of independent reasoning. The latter two describe reported understanding and evaluation practices; they do not establish whether AI improved those capabilities.

The questions “how often do you ask AI for explanations?” and “how often do you explore unfamiliar code with AI support?” provide complementary context on learning-related use. Their frequency alone does not demonstrate improved understanding. The profile is not a validated cognitive-impact scale and does not establish objective changes in cognitive ability, retained knowledge, or long-term skill development. Overlapping items are not treated as independent additional evidence or combined into a cognitive-impact score.

B Quantitative KPI Clusters for Delivery Measurement

The following KPI clusters can be used with Agile sprints, waterfall phases, hybrid models or other managed delivery systems. Each organisation needs to define formulas in its operational KPI catalogue to match its workflow, data and governance rules.

Table 17: General quantitative KPI clusters for delivery measurement

KPI cluster	Measurement intent	Typical evidence	Interpretation guardrail
Capacity-normalised output	Measure completed delivery value against the real capacity available to the team or delivery group.	Accepted scope, completed work packages, released features, completed backlog items, available capacity, allocation, and planned effort.	Higher output is positive only when quality, scope mix, and delivery reliability remain stable.
Planned-work reliability	Measure whether the team delivers the work that was planned or committed at the beginning of the measurement window.	Sprint commitments, milestone scope, phase deliverables, baseline plans, accepted completion evidence, and delivery dates.	Additional work should not be counted as productivity if planned work is missed or deferred.
Additional useful work	Measure extra valuable work completed in the same time window and capacity baseline.	Additional documentation, test cases, enhancements, clarified requirements, accepted change requests, or extra backlog/work-package items.	Extra work must be additive and useful, not a replacement for missed scope or a symptom of uncontrolled churn.
Scope stability and change control	Measure how much work is added, removed, replanned, or re-estimated after the baseline has been set.	Pull-in and pull-out work, change requests, replanning events, deferred scope, estimation changes, and exception records.	High change volume may indicate volatility rather than AI-enabled productivity.
Flow and wait-state reduction	Measure whether work moves faster through analysis, implementation, review, testing, approval, and release states.	Phase lead times, cycle times, readiness times, review duration, queue time, deployment lead time, and approval waiting time.	Compare actual events under stable definitions; disclose changes in work mix, capacity and workflow. Estimates are forecasts, not counterfactuals.
Quality and rework control	Measure whether faster or greater delivery is achieved without degrading correctness, maintainability, or release stability.	Defects, rework, reopened items, failed validations, test results, first-pass acceptance, incidents, and release failures.	Productivity should be treated as positive only when quality is stable or improving.

KPI cluster	Measurement intent	Typical evidence	Interpretation guardrail
Delivery evidence completeness	Measure whether completed work includes the evidence needed for review, auditability, maintenance, and handover.	Documentation, test evidence, acceptance records, traceability, review notes, decision logs, and release evidence.	More output is valuable only when the evidence needed to trust and maintain it is complete.
AI adoption and usage context	Measure whether teams have sufficient AI exposure to interpret delivery changes as AI-related.	Tool usage, task coverage, active users, usage intensity, training completion, and survey-based adoption evidence.	AI usage is an exposure signal, not productivity proof.
Safe-use and governance context	Measure whether AI-assisted delivery follows required rules, verification practices, and oversight expectations.	Safe-use checks, human review evidence, policy acknowledgement, risk records, exception handling, and audit trails.	Governance evidence constrains KPI interpretation; it should not be treated as a standalone productivity measure.