

AI Coding Agents in the Norwegian Public Sector: Adoption, Governance, and Practitioner Concerns

Hans Kristian Flaatten
Norwegian Labour and Welfare Administration (Nav)
hans.kristian.flatten@nav.no

Anders Olsen Sandvik
Statens pensjonskasse (SPK)
anders.olsen.sandvik@spk.no

Abstract

Context

AI coding tools have moved from autocomplete assistants to autonomous agents that plan, execute, and review code changes across entire codebases. Despite widespread adoption in the software industry, to our knowledge no sector-level studies examine how public-sector organizations adopt and govern these tools.

Objective

This study investigates AI coding tool adoption patterns, governance structures, and practitioner concerns across the Norwegian public sector.

Methods

Cross-sectional survey of 115 respondents from 57 organizations in the Norwegian public sector (Q2 2026), distributed via the Offentlig PAAS community of practice. The instrument comprises 27 items with confidence indicators on key organizational questions and branching logic for non-users.

Results

We find broad personal adoption (92%, $n=106/115$, 95% CI: 86–96%), with 76% of users employing agentic capabilities (95% CI: 67–83%) — zero respondents use only autocomplete. GitHub Copilot dominates procurement (82% officially approved), but Claude Code shows substantial shadow adoption (42% personally used vs. 35% approved, on a common $N=115$ base). Shadow adoption depends on definition: 62% use tools beyond their approved list (broad, 95% CI: 53–72%), 30% when excluding browser-based LLMs and open-source tools (narrow, 95% CI: 22–40%). Governance lags: while 83% have some policy (95% CI: 76–89%), only 8% report a structured support program (e.g., champions networks) and 62% apply no special code review for AI-generated code. Governance maturity correlates moderately with adoption level (Spearman $\rho=0.439$, permutation $p<0.001$). Effect measurement is weak — only 11% have own metrics. Asked to pick at most three concerns, respondents most often ranked skill loss among them (71%, 95% CI: 62–79%), well ahead of security (34%) or privacy (27%); practitioners (79%) cite skill loss more often than technical and organizational leaders combined (64%), though this difference is not statistically significant ($p=0.104$). Compared with a permutation null that preserves answer distributions and organization sizes, respondents within the same organization agree more than chance on policy and vendor clarification, and disagree more than chance only on how effects are measured, and only marginally (96% of multi-respondent organizations vs. 83% expected; $P\approx 0.03$, uncorrected).

Conclusion

AI coding tool adoption in the Norwegian public sector has moved from individual experimentation to organizational procurement within months, but governance, measurement, and support lag behind. Shadow adoption of CLI-based agents shows that centralized procurement has not kept pace with how fast tools change. Policy and vendor arrangements appear to be shared organizational knowledge,

while effect measurement is the one area where colleagues disagree somewhat more than chance would predict. Organizations should consider moving beyond “procurement-as-governance” toward continuous capability support and outcome measurement that practitioners know about.

Keywords: AI coding agents, developer tools, public sector, governance, shadow IT, software engineering, agentic AI

1. Introduction

Software development is being reshaped by AI-powered coding tools. Since GitHub launched Copilot in 2021 as an autocomplete extension, the technology has progressed through at least two subsequent generations: AI assistants capable of multi-turn reasoning and contextual planning (2024–2025), and autonomous coding agents that can independently navigate repositories, execute tests, and iterate on failures with minimal human oversight (2025–2026). This progression — from single-line code suggestions to agents that open pull requests and run continuous integration pipelines — has compressed years of capability growth into months.

Adoption has followed a correspondingly steep trajectory. GitHub Copilot passed 20 million all-time users by July 2025 (TechCrunch, 2025). The Stack Overflow Developer Survey reported that 84% of developers use or plan to use AI tools (Stack Overflow, 2025), up from 76% in 2024 and 70% in 2023 (Stack Overflow, 2023). The Pragmatic Engineer’s 2026 survey found that 95% of respondents use AI tools at least weekly, with 55% regularly employing agentic features that go beyond assisted code completion (Orosz & Nilsson, 2026). What began as a productivity experiment for individual developers has become, within three years, a widespread feature of professional software engineering practice.

Yet organizational governance has not kept pace. Consumer-grade AI coding tools are freely available — many offer free tiers or can be installed without administrative approval — and this accessibility creates a gap between what individual developers use and what their organizations have sanctioned, evaluated, or even noticed. Enterprise-focused features such as compliance frameworks, audit logging, and air-gapped deployment remain immature across most tool vendors (Shah, 2025). The result is a governance deficit: organizations procure and permit AI coding tools faster than they establish policies for their use, training for their effective application, or measurement of their impact.

This deficit is especially consequential in the public sector. Public-sector software development operates under constraints that distinguish it from commercial settings: procurement regulations, data sovereignty requirements, transparency obligations, and heightened accountability for how public funds are spent. These constraints interact with AI coding tools in ways that current research has not examined. National AI strategies — including Norway’s 2020 National Strategy for Artificial Intelligence (Norwegian Government, 2020) and guidance from Digitaliseringsdirektoratet (Digdir, 2023–2026) — address AI adoption broadly but do not specifically consider developer-facing tools. The developer-as-user perspective is largely absent from public-sector AI governance discourse at the national level.

Existing research on AI coding tools focuses overwhelmingly on individual productivity effects. Controlled experiments measure whether developers complete tasks faster with AI assistance (Becker et al., 2025). Large-scale observational studies track code volume and quality metrics across developer populations (BlueOptima, 2024). Industry surveys report adoption percentages and tool preferences (Stack Overflow, 2025; Orosz & Nilsson, 2026). What is missing is the organizational perspective: how do organizations — not individual developers — adopt, govern, and support AI coding tools? And how do these organizational decisions interact with practitioner behavior and concerns? To our knowledge, no sector-level study examines these questions, and no peer-reviewed study addresses AI coding tool adoption in the public sector specifically.

The Norwegian public sector offers a productive setting for investigating these questions. The Offentlig PAAS community of practice, which connects developers and technical leaders across more than 80 Norwegian public-sector organizations, provides a sampling frame with high sector coverage. This community includes organizations from central government agencies to public enterprises, covering domains from tax administration to transport infrastructure.

This study investigates AI coding tool adoption, governance, and practitioner concerns across the Norwegian public sector through a cross-sectional survey of 115 respondents from 57 organizations. We address three research questions:

- **RQ1 (Adoption):** How are Norwegian public-sector organizations adopting AI coding tools, and to what extent does unofficial (“shadow”) adoption exist?
- **RQ2 (Governance):** What governance structures have organizations established for AI coding tools, and how do these relate to adoption maturity?
- **RQ3 (Barriers):** What concerns and barriers do practitioners experience, and do these differ by role?

2. Background and Related Work

2.1 AI Coding Tools: From Autocomplete to Agents

The evolution of AI-assisted software development can be divided into three phases. The *pre-agentic phase* (2021–2023) was defined by inline autocomplete: tools such as GitHub Copilot and Amazon CodeWhisperer generated single-line or block-level code suggestions within the developer’s editor, with the developer accepting or rejecting each suggestion individually. The *AI assistant phase* (2024–2025) introduced structured reasoning, multi-turn conversations, and tool use within IDE-integrated chat interfaces. Language models became capable of planning changes across files, explaining code, and generating tests on request — but the developer remained the primary executor of changes. The *agentic phase* (2025–2026) marks a qualitative shift: tools such as Claude Code, GitHub Copilot agent mode, and OpenAI Codex operate as autonomous agents that navigate repositories, make cross-file edits, execute tests, iterate on failures, and can push commits and open pull requests with minimal human intervention (Orosz & Nilsson, 2026).

The key distinction in this progression is between *AI-assisted* coding, where the developer remains in the loop for each action, and *AI agents*, which execute multi-step tasks autonomously. The Pragmatic Engineer’s 2026 survey found that 55% of developers regularly use agentic features, with staff-level and senior engineers being the heaviest users (63.5% compared to 49.7% for regular engineers) (Orosz & Nilsson, 2026).

Adoption has reached scale. The Stack Overflow Developer Survey found that 84% of developers use or plan to use AI coding tools (Stack Overflow, 2025). However, tool preferences differ by organizational context. Large organizations (10,000+ employees) prefer GitHub Copilot (56%), a pattern driven by existing Microsoft enterprise agreements and centralized procurement. Startups, by contrast, favor Claude Code (75%) and Cursor (42%), where individual developers select tools without centralized gatekeeping (Orosz & Nilsson, 2026).

AI coding tools can be categorized by their integration model into four groups: IDE extensions (e.g., GitHub Copilot, Amazon Q) that install as plugins in existing editors; AI-native IDEs (e.g., Cursor, Windsurf) that are purpose-built editors with agent-first interfaces; CLI agents (e.g., Claude Code, OpenAI Codex) that operate in the terminal without IDE dependency; and open-source or self-hosted tools (e.g., Aider, Continue) that offer data sovereignty advantages but limited enterprise support. Each category implies different procurement paths, governance surfaces, and shadow-adoption risk profiles — CLI tools and free tiers, for instance, are often invisible to centralized IT management, a pattern studied in the information-systems literature as shadow IT (Haag & Eckhardt, 2017).

A further dimension is the Model Context Protocol (MCP), released by Anthropic in November 2024 (Anthropic, 2024) and since adopted by many major tools. MCP allows AI coding agents to interact with external systems — GitHub, Kubernetes, databases, observability platforms — through a standardized protocol. This transforms coding agents from code-generation tools into unified developer interfaces that can span the entire development lifecycle, raising governance questions about what data leaves the organization and what actions the agent is authorized to take.

2.2 Organizational Adoption and Governance

While individual adoption of AI coding tools is well documented, organizational governance remains poorly understood. Enterprise-grade AI coding tools face documented feature gaps: consumer-focused tools lack compliance frameworks, adequate context analysis, and air-gapped deployment options, and organizations using them may face higher credential exposure and regulatory violations (Shah, 2025; note: this source is published by Augment Code, an AI coding tool vendor, and should be interpreted with awareness of its commercial context).

Known adoption barriers include security concerns, intellectual property and licensing uncertainties arising from models trained on public code, code quality concerns, and skill displacement fears — although 70% of professional developers in broad industry surveys do not perceive AI as a direct threat to their employment (Stack Overflow, 2024). Ox Security, an application-security vendor, likens AI coding tools to “junior developers who fundamentally lack architectural judgment and security awareness” (Ox Security, 2025).

No widely adopted maturity model exists specifically for AI coding tool governance. General AI maturity models (e.g., Gartner AI Maturity Model, MIT CISR Enterprise AI Maturity Model) provide multi-stage frameworks across dimensions of data, technology, people, and governance, but these are not specific to developer tooling. The closest framework is the DORA AI Capabilities Model (DORA, 2025b), published in December 2025, which identifies seven organizational capabilities that amplify benefits from AI-assisted software development: clear AI policy, a healthy data ecosystem, user-centric focus, resilient internal platforms, feedback loops, governance, and literacy. The premise is stated in the accompanying 2025 DORA report: “AI’s primary role is as an amplifier, magnifying an organization’s existing strengths and weaknesses” (DORA, 2025a).

The DORA 2025 report (DORA, 2025a) found no measurable impact of AI adoption on developer burnout, friction, or organizational delivery metrics. Software delivery instability actually increased with AI adoption in the measured sample. This finding — that AI tools may improve individual output without improving organizational outcomes — provides a baseline against which sector-specific studies can be compared.

2.3 The AI Productivity Paradox

A growing body of evidence challenges the productivity claims made by AI coding tool vendors, producing an AI productivity paradox: individual output metrics rise while organizational delivery metrics remain flat.

On the individual side, vendor-reported numbers have been optimistic. GitHub’s own research on Copilot and code quality relies heavily on developer questionnaires and on simplistic metrics such as line length (BlueOptima, 2024; BlueOptima is a developer-analytics vendor with its own commercial stake in this debate). Independent studies tell a different story. A randomized controlled trial by Becker et al. (2025), conducted by METR with experienced open-source developers using Cursor Pro and Claude Sonnet, found that AI tools actually *increased* task completion time by 19% — despite developers predicting a 24% speedup. BlueOptima, analyzing data from 218,000 developers, estimated gains of approximately 4% and reported that 88% of developers still rework AI-generated code before committing (BlueOptima, 2024).

On the organizational side, the disconnect is sharper. Faros AI, an engineering-analytics vendor, reports telemetry from about 10,000 developers across 1,255 teams in which “engineers are completing twice as many code changes, but company metrics remain flat” (Faros AI, 2025). The gains appear to dissipate between individual output and system-level delivery. One possible mechanism is quality effects: Jellyfish, another engineering-analytics vendor, reports data from hundreds of organizations in which bug fixes made up 7.5% of pull requests at companies with low AI adoption and 9.5% at companies with high adoption; although Jellyfish interprets this range as evidence of stable code quality, the upward trend is also consistent with AI generating additional defects that require subsequent correction (Jellyfish, 2025). The DORA 2025 report confirmed this pattern at scale, finding that AI positively correlates with individual effectiveness, code quality perceptions, and throughput, but shows no measurable impact on burnout, friction, or organizational delivery (DORA, 2025a).

The measurement dimension of this paradox is itself poorly understood. No study provides clean data on what proportion of organizations systematically measure AI coding tool impact. A broader finding — that 56% of CEOs report no significant financial benefit from AI to date (PwC, 2026) — suggests the measurement gap extends well beyond developer tools. Vendor dashboards track inputs (acceptance rates, completions per day, active users) rather than outcomes, meaning an organization can show high AI utilization while simultaneously accumulating technical debt and review burden (BlueOptima, 2024).

2.4 Public Sector Context

The public sector faces distinctive conditions that shape AI coding tool adoption. Procurement constraints, data sovereignty requirements, transparency obligations, and political accountability create a governance environment that differs materially from commercial software development.

Norway has taken an active stance on AI in the public sector. The Norwegian Government published its National Strategy for Artificial Intelligence in 2020, emphasizing data sharing, language technology, and a national resource centre under Digitaliseringsdirektoratet (Norwegian Government, 2020). Digdir has since published comprehensive guidance for responsible AI use and development in public administration (“Veiledning for ansvarlig bruk og utvikling av kunstig intelligens”), covering procurement of generative AI tools, responsible usage training, and hosting a cross-sector forum (“Fagforum for KI i offentlig sektor”) (Digdir, 2023–2026). However, this guidance addresses general AI applications — chatbots, citizen services, administrative tasks — and does not specifically cover developer-facing AI coding tools. The developer-as-user perspective is largely absent from national-level governance discussions.

At the organizational level, Oslo kommune’s Digitaliseringsetaten (which absorbed Oslo Origo on 1 January 2026) has published what appears to be the most comprehensive Norwegian governance framework specifically for AI coding tools (Oslo kommune, 2025–2026). The framework includes a structured tool portfolio with per-tool assessments covering vendor data processing guarantees, data center locations, and risk assessments; a formal developer sandbox with explicit vendor requirements (no-training, no-storage, no-sharing guarantees for code processed); a dedicated cross-functional AI team (KITT) that handles competence building, legal assessments, and prototyping; and three governance principles — legality, traceability, and meaningful human control. This open-source framework illustrates what mature AI coding tool governance looks like in a Norwegian public-sector context, but it appears to be an outlier — most organizations lack comparable documentation.

In the broader Nordic context, a Finansforbundet study across Nordic countries (n=1,211) found that the top barriers to AI adoption are unclear or uncertain potential (26%), AI not being a management priority (23%), and lack of digital skills (21%) — with Norway not an outlier among Nordic countries (Finansforbundet, 2024). Denmark appears more advanced in public-sector AI governance: Digitaliseringsstyrelsen has published comprehensive guides for responsible use of generative AI by public-

sector authorities, and Denmark has established both a regulatory sandbox for AI and a portfolio of 40 signature AI projects in the public sector (Digitaliseringsstyrelsen, 2024).

Norwegian academic research on AI coding tools is limited but relevant. Stray, Moe, and colleagues at the University of Oslo and SINTEF have published the most pertinent Norwegian-context studies, including a longitudinal mixed-methods case study of developer productivity with and without GitHub Copilot (Stray et al., 2025) and a study of human-AI collaboration in software development (Stray, Barbala, & Wivestad, 2025). These studies operate at the individual and team level, examining how specific developers experience and adopt AI tools within their projects. They do not address sector-wide adoption patterns, organizational governance, or the systemic phenomena — such as shadow adoption or governance-adoption gaps — that characterize organizational-level adoption.

2.5 Research Gap and Contribution

The preceding sections reveal three gaps. First, no sector-level organizational adoption survey exists for AI coding tools. Existing studies either measure individual productivity (Becker et al., 2025; Stray et al., 2025), analyze aggregate code-level data (BlueOptima, 2024; Faros AI, 2025), or survey self-selected developer populations without organizational context (Stack Overflow, 2025; Orosz & Nilsson, 2026). The organizational unit of analysis — how do organizations adopt, approve, support, and govern these tools? — is absent from the empirical literature.

Second, no public-sector-specific study of AI coding governance exists. Digdir’s guidance covers general AI use but not developer tooling. Oslo kommune’s framework demonstrates that structured governance is possible, but no study examines governance practices across multiple organizations or relates governance maturity to adoption patterns.

Third, no measurement maturity data exists at the sector level. The AI productivity paradox documented by DORA (2025a) and Faros AI (2025) shows that individual gains fail to translate into organizational outcomes, yet no study quantifies how — or whether — organizations measure AI coding tool impact at scale.

This study addresses all three gaps through a cross-sectional survey of developers and technical leaders across 57 Norwegian public-sector organizations, examining adoption patterns and shadow adoption (RQ1), governance structures and their relationship to adoption maturity (RQ2), and practitioner concerns across different organizational roles (RQ3).

3. Methodology

3.1 Research Design

We conducted a cross-sectional survey of developers and technical leaders in Norwegian public-sector organizations during Q2 2026 (April–May). This represents the first wave of a planned longitudinal study; the cross-sectional design captures a snapshot of adoption, governance, and concerns at a point when AI coding tools are transitioning from individual experimentation to organizational procurement. The study follows survey guidelines for empirical software engineering as described by Kitchenham and Pfleeger (2008) and Molléri et al. (2016).

3.2 Survey Instrument

The survey instrument comprises 27 items organized into 10 sections, including three confidence follow-ups, the interview opt-in and a contact field. Question numbers in this paper follow the website form (e.g., Q14 = policy, Q20 = concerns, Q24 = interview opt-in); the supplementary instrument uses its own design numbering and opens with a table that maps one to the other. The instrument is structured in Norwegian (bokmål) and covers three thematic areas aligned with the research questions: organizational adoption and tool use (RQ1, Sections 3–5), governance and organizational support (RQ2, Section 6), and concerns and barriers (RQ3, Section 7), with additional sections for demographics (Section 2), follow-up interest (Section 8–9), and consent (Section 1).

Three branching points route respondents through the instrument. Organizations without developers (Q3 = “Ingen — vi har ikke utviklere”) are directed to the closing section; no respondent triggered this exit. Respondents who do not personally use AI coding tools (Q8 = “Nei”) skip the personal usage section and proceed to governance and concerns questions. Respondents who opt in to follow-up interviews (Q24 = “Ja”) are routed to a contact information section.

A distinctive feature of the instrument is the use of confidence indicators on key organizational questions. For questions where respondents report on organizational-level characteristics — such as the organization’s adoption level (Q5), approved tool list (Q6), and policy status (Q14) — a follow-up question asks respondents to rate their confidence in the answer as “certain” (*sikker*), “fairly sure” (*ganske sikker*), or “estimate” (*anslag*). These indicators enable sensitivity analysis: claims about organizational governance can be tested at different confidence thresholds to assess whether low-confidence responses inflate or deflate reported values.

Multi-select fields (e.g., approved tools, usage modes, concerns) allow respondents to select multiple options. The concern question (Q20) asks for the three most important concerns (*Velg de tre viktigste*) and accepts at most three; 79% of respondents (91/115) selected three. These are stored as semicolon-separated values in the dataset. Estimated completion time was 10–12 minutes.

3.3 Population and Sampling

The target population is developers and technical leaders in Norwegian public-sector organizations. We distributed the survey primarily through the Offentlig PAAS community of practice, which connects professionals from more than 80 member organizations across Norwegian central and local government. Distribution channels included the community’s Slack workspace, email networks, and broader Norwegian public-sector IT channels. This constitutes a convenience sample, with the acknowledged limitation that it overrepresents organizations with platform-engineering capacity, active technical communities, and interest in developer tooling.

3.4 Respondent Profile

We received 115 responses from 57 identified organizations; one further respondent entered a unit name instead of an organization, so this response is counted as organization unknown and left out of organization-level analyses. The first 35 responses (5–7 April 2026) were collected through a Google Forms version of the instrument and imported into the dataset; the remaining 80 were submitted through the survey form on the Offentlig PAAS website, which used the same questions and consent text. The data do not record why the channel changed. In four website-form responses (8–21 April), free text entered under “other” was attached to the last selected option; we moved it back to “other” so that every selected option is counted. Of the 57 organizations, 27 have multiple respondents, with the most represented organization contributing 7 responses.

Respondents reported their role from a set of seven options, which we recoded into three analytical groups: *practitioners* (developers, platform engineers; n=56, 49%), *technical leadership* (tech leads, team leads; n=44, 38%), and *organizational leadership* (CTOs, non-technical leaders; n=15, 13%). One response was submitted on behalf of an AI working group (*faggruppe for KI*); we keep it as one organizational-leadership respondent. Organization sizes by number of developers are distributed across the range, with the largest groups being 51–200 developers (41%) and 200+ developers (31%).

Nine respondents (8%) report not personally using AI coding tools. These non-users remain in organizations with documented adoption — four at the “most developers” level, three at “officially approved,” and two at “individual” — and they contribute data on organizational governance and personal concerns.

3.5 Analysis Approach

We employ a dual-level reporting strategy. Individual-level analysis (N=115) is used for questions capturing personal perceptions, concerns, and usage patterns, where each respondent represents one

practitioner’s perspective. Organization-level analysis (N=57) is used for organizational characteristics such as adoption level and policy status, where we aggregate multiple respondents from the same organization. For single-select fields, we use majority-vote consensus (the most common response represents the organization; ties yield no majority and are excluded from that variable’s analysis). For multi-select fields (e.g., approved tools), we use union aggregation — a tool is counted for an organization if any respondent mentions it. This approach avoids double-counting organizational attributes while preserving individual variation in perceptions and concerns.

Descriptive statistics are reported with 95% confidence intervals for proportions. Because the sample is a convenience sample, these intervals quantify sampling variability under random-sampling assumptions that do not strictly hold; they do not indicate population coverage. For role-based concern differences, we combine the two leadership groups (technical and organizational) into a single leadership category due to the small organizational-leadership cell size (n=15), then test practitioners versus combined leadership using Fisher’s exact tests across four concern types, applying Bonferroni correction at $p < 0.0125$. For the governance–adoption relationship, we compute Spearman rank correlation (two-sided permutation p-values, 10,000 draws, fixed seed) and Cramér’s V across governance indicators. We conduct sensitivity analyses by filtering responses at different confidence thresholds (all respondents, certain + fairly sure, certain only) to assess the robustness of organization-level findings.

Intra-organizational agreement is assessed for the 27 organizations with multiple respondents. For each governance indicator, we compute the proportion of organizations where at least two respondents give different answers. Because some disagreement is expected by chance whenever several people answer a multi-option question, we compare each observed rate with a permutation null: the answers to that question are shuffled across all 115 respondents (which preserves the answer distribution and every organization’s size), and the disagreement rate is recomputed over 10,000 draws with a fixed seed. Only a rate above the null indicates that colleagues disagree more than random respondents would.

3.6 Ethical Considerations

The survey is anonymous. Respondents provide informed consent in accordance with GDPR Article 6(1)(a) before proceeding. The consent text describes the study’s purpose, the voluntary nature of participation, and how data will be handled. Respondents who opt in to follow-up interviews provide contact information in a separate section; this information is stored separately from survey responses, excluded from the anonymized dataset, and will be deleted by 31 December 2026.

We disclose our position as insider-researchers: both principal investigators are active members of the Offentlig PAAS community through which the survey was distributed. This position provided access to the sampling frame and domain knowledge that informed instrument design, but also introduces potential biases in survey framing and interpretation. We address these threats in Section 6.

The anonymized respondent-level dataset (without organization identifiers, free text, timestamps or contact information) and aggregate organization-level tables are available for replication (see Data Availability). The full survey instrument (Norwegian, with English translations) is provided as supplementary material: <https://github.com/offentlig-paas/open-data/blob/main/state-of-ai-agents/2026/instrument.md>

4. Results

We present findings organized by research question. All statistics report 95% confidence intervals (CI) where applicable. We distinguish respondent-level (N=115) and organization-level (N=57) analyses where constructs differ, and note the base for all percentages. Sensitivity analyses in Section 4.4 examine robustness to respondent confidence levels.

4.1. RQ1: Widespread Adoption with Substantial Shadow Use

Organizational adoption. At the respondent level, 78% of respondents (90/115, 95% CI: 70–85%) report that their organization has reached at least the “officially approved” stage of AI coding tool adoption. This divides evenly: 39% report their organization at the “most developers use it” (*de fleste utviklere bruker det*) level, and 39% at “officially approved” (*offisielt godkjent*). A further 15% report adoption at the “some teams” level and 7% at the “individual” level. No respondent reports zero use of AI coding tools in their organization. Figure 1 shows the distribution of adoption levels.

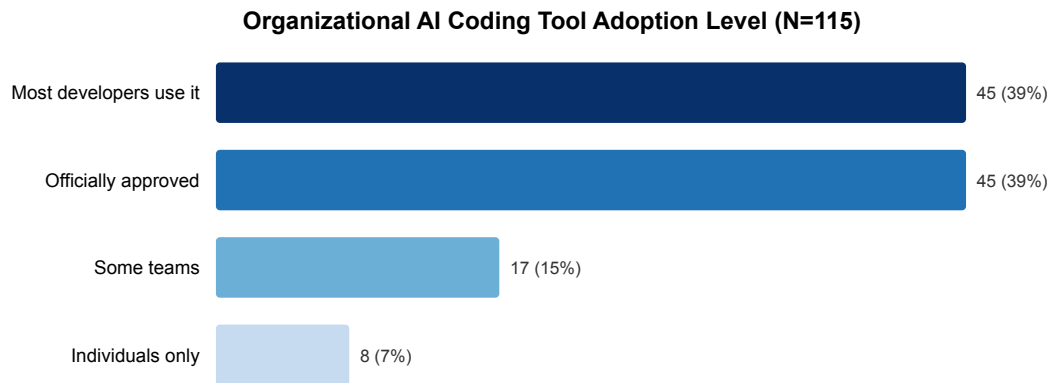


Figure 1: Organizational AI coding tool adoption level, respondent level (N=115)

When aggregated to the organization level using a majority-vote approach (N=57), the picture shifts: 32% of organizations (18/57) fall at the most-developers level, 26% (15/57) at officially-approved, and 26% (15/57) are classified as “disagreement” — organizations where respondents gave conflicting answers about the adoption stage. Among the 27 organizations with multiple respondents, 23 exhibit intra-organizational disagreement on the adoption level. Section 4.2 reports broader disagreement patterns.

Individual adoption. Personal use of AI coding tools is widespread in this sample: 92% of respondents (106/115, 95% CI: 86–96%) report using such tools. The nine non-users still represent organizations where AI tools have been adopted — four are in organizations at the most-developers level, three at officially-approved, and two at the individual level — indicating that organizational adoption does not imply uniform individual uptake.

Tool landscape. GitHub Copilot dominates procurement, with 82% of respondents (94/115) reporting it as an approved tool in their organization. Claude Code ranks second, reported as approved by 35% (40/115) and personally used by 42% (48/115; 45% of the 106 tool users). This gap between approval and personal use constitutes the clearest single-tool instance of shadow adoption and is discussed further below. Figure 2 illustrates the approved-versus-used comparison across tools; note that the figure reports personal use on the base of tool users (N=106).

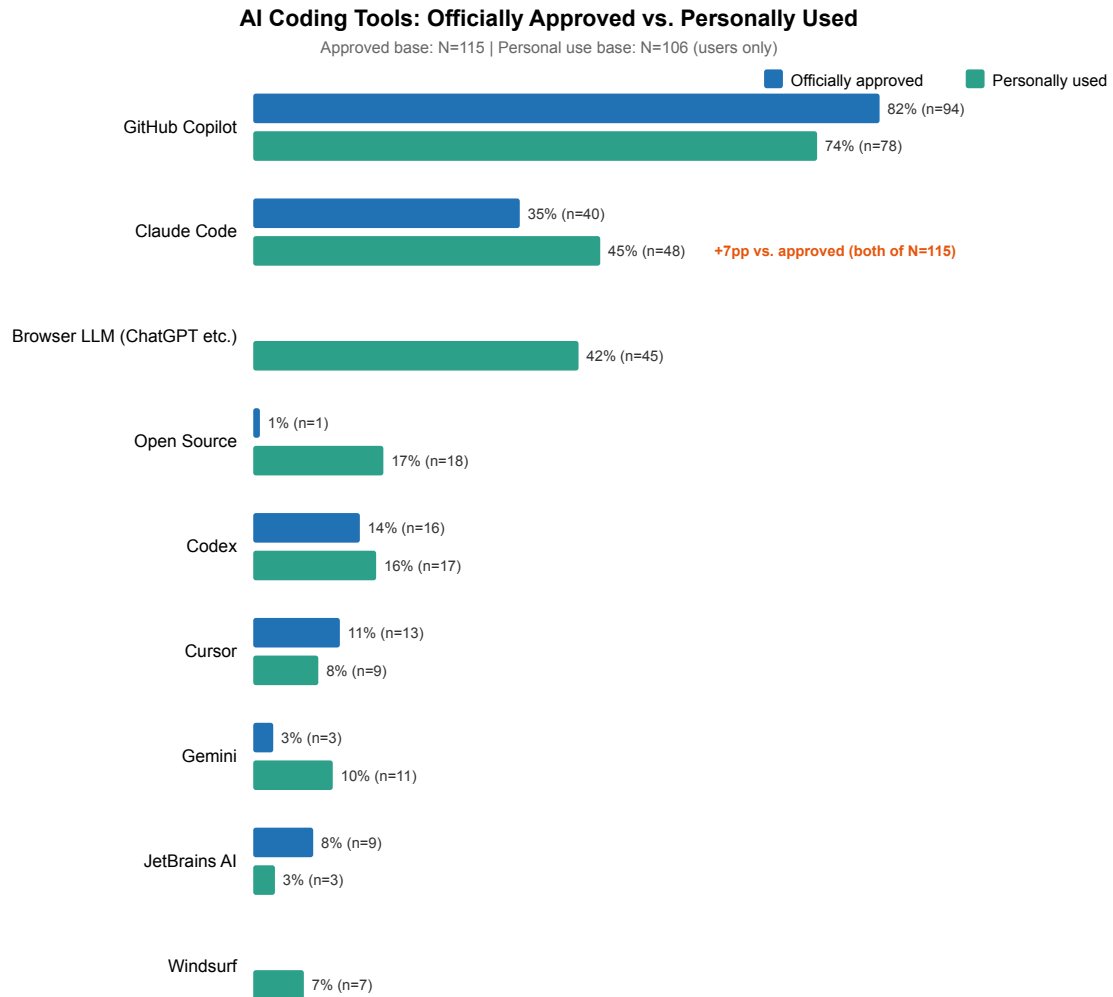


Figure 2: AI coding tools: officially approved (N=115) vs. personally used (N=106 users)

Shadow adoption. We operationalize shadow adoption as tool use outside an organization's approved list, measured among respondents whose organizations have at least one approved tool (n=96). Each respondent's personally used tools are compared against the approved tools that the same respondent reports, not against the organization-level union. We report two metrics to reflect differing governance-risk profiles. The broad measure counts any mismatch between approved and personally used tools, including browser-based LLMs and open-source tools: 62% of respondents (60/96, 95% CI: 53–72%) use at least one tool that is not on their organization's approved list. The narrow measure excludes browser-based LLMs and open-source tools, which carry a different risk profile since they typically do not process internal code through third-party IDE extensions: 30% (29/96, 95% CI: 22–40%). Both measures remain stable across confidence filters (58–63% for the broad measure, 30–32% for the narrow measure across all, certain+fairly-sure, and certain-only respondents). The primary shadow tool is browser-based LLMs (41 users), followed by Claude Code (17 users). Figure 3 visualizes the flow from approved tools to actual usage.

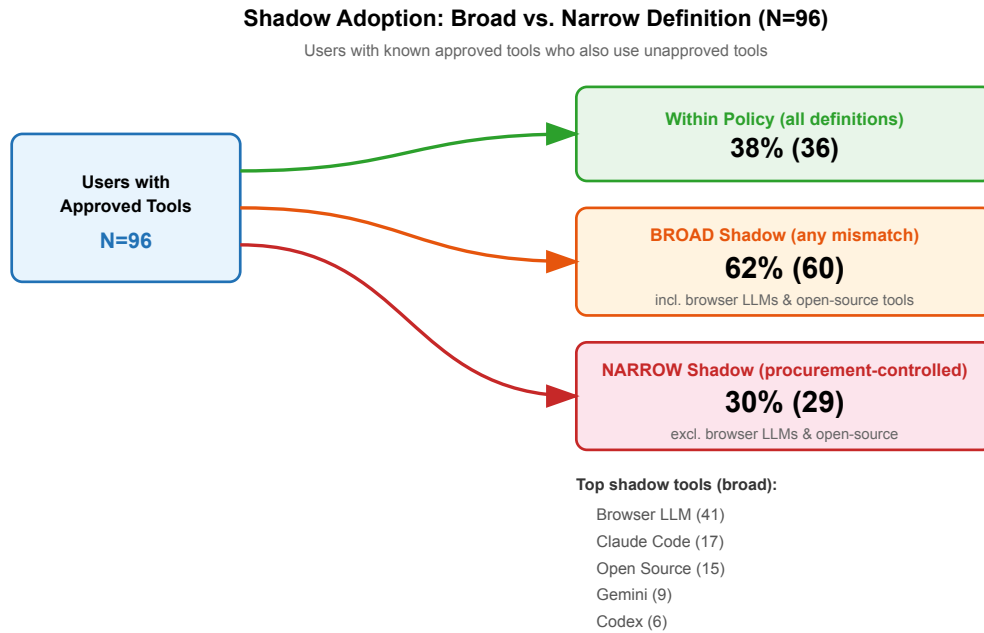


Figure 3: Shadow adoption: broad vs. narrow definition (n=96 users with known approved tools)

Agentic usage. Among the 106 respondents who personally use AI coding tools, 76% (81/106, 95% CI: 67–83%) report using local or cloud agents — tools that autonomously plan and execute multi-step changes. No respondent uses only autocomplete. The most common usage modes are chat (82%), local agent (73%), debugging (66%), planning (64%), and autocomplete (55%). Figure 4 presents usage modes beyond passive code completion.

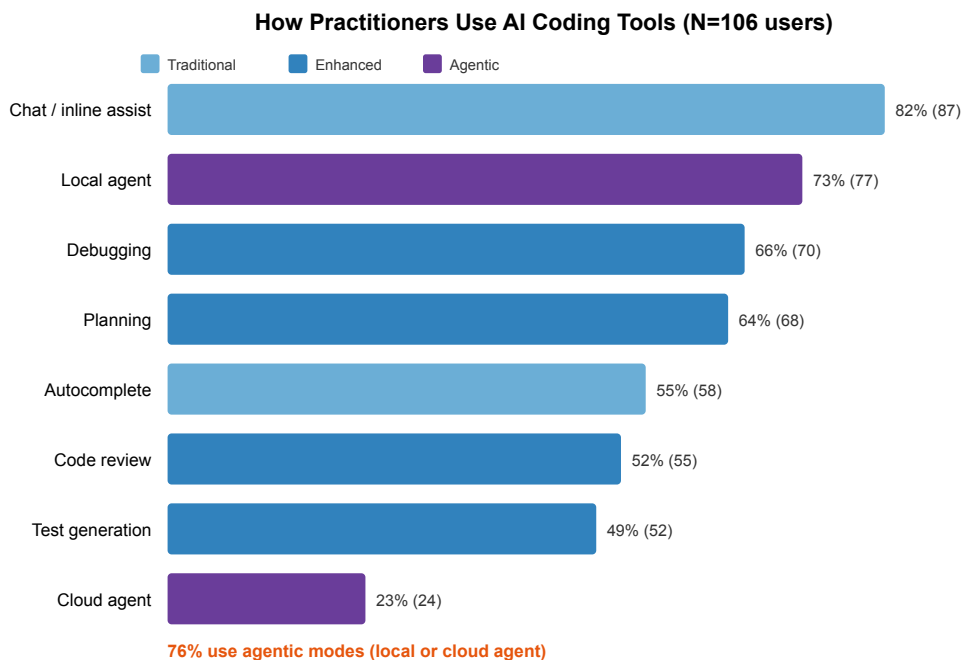


Figure 4: How practitioners use AI coding tools (N=106 users)

Integration, frequency, and perceived benefits. Nearly half of tool users (49%, 52/106) report connecting their AI tools to external systems such as source control, CI/CD pipelines, and monitoring platforms. Most users engage daily: 69% (73/106) use AI coding tools every day, and a further 14% use them several times per week. When asked about the primary benefit, speed was the most common response (36%), followed by reduced toil (25%) and learning (13%).

4.2. RQ2: Governance Lags Behind Adoption

Policy coverage. A large majority of respondents report that their organization has some form of policy governing AI coding tools: 83% (96/115, 95% CI: 76–89%). Of all respondents, 41% (47/115) report an AI-tool-specific policy and 43% (49/115) a general AI policy that also covers coding tools, an approximately even split. At the organization level (majority vote, N=57), 40% (23/57) have a specific policy, 35% (20/57) a general policy, 7% (4/57) no policy and 5% (3/57) unknown, while respondents disagree in 12% (7/57). However, the existence of a policy document and the presence of operational support are markedly different: only 8% (9/115) report a structured support program (*strukturert program*, e.g., champions networks, dedicated resource persons, or feedback routines).

Organizational support. The dominant support model is informal: 43% of respondents (50/115) report that their organization relies on informal channels (e.g., Slack discussions, peer help) as its primary support mechanism. Shared resources — such as wikis, prompt libraries, or documented guidelines — are reported by 23% (26/115), and organized training or courses (*organisert opplæring eller kurs*) by 19% (22/115). Figure 5 compares policy existence with the other governance indicators.

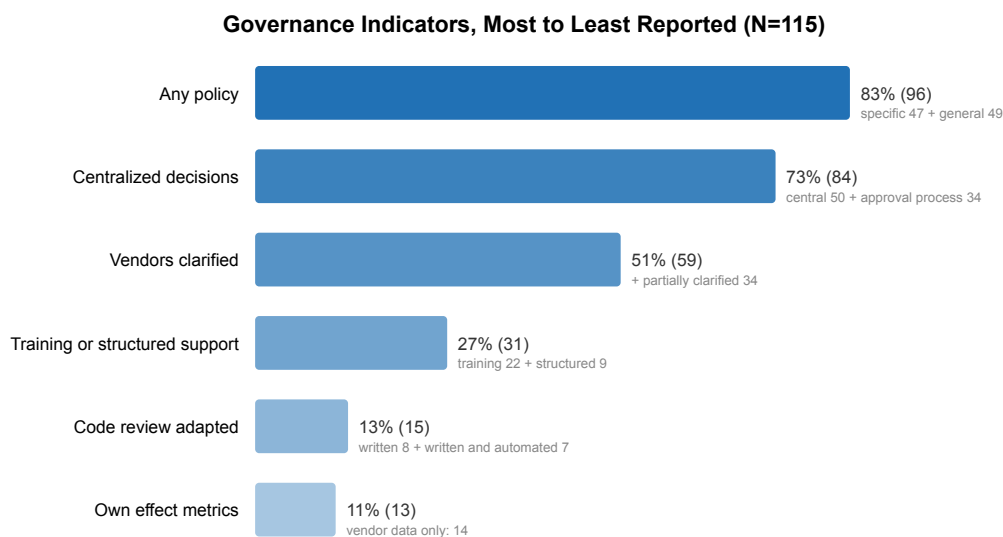


Figure 5: Governance indicators, most to least reported (N=115)

Decision-making and vendor management. Decision-making about AI tool adoption is primarily centralized: 43% of respondents report centralized decisions and 30% describe a formal approval process. Regarding vendor and data processing agreement (DPA) clarity, 51% (59/115) report that their organization has clarified data processing terms with vendors, while 30% report partial clarity.

Effect measurement. Measurement of AI tool effects is the weakest governance dimension. Only 11% of respondents (13/115) report that their organization has developed its own metrics. The largest group, 43%, describe informal assessment, 12% rely on vendor-provided data, and 26% do not know how effects are measured.

Code review practices. Despite widespread adoption, 62% of respondents (71/115) report that AI-generated code receives no special handling in code review. Among the remainder, 12% report verbal guidelines, 7% written policies, and 6% written policies with automated enforcement.

Governance-adoption association. Governance maturity and adoption level show a moderate positive association (Spearman $\rho=0.439$, permutation $p<0.001$). At the organization level, among the 42 organizations with a majority-vote adoption level (excluding the 15 with disagreement), the association is similar ($\rho=0.441$, permutation $p\approx 0.004$). Cramér's V values range from 0.22 to 0.37 across individual governance indicators, indicating small-to-medium effect sizes. The composite governance index sums six binary indicators: an AI-tool-specific policy (general AI policies do not count), organized training or a structured support program, centralized decisions or a formal approval process, vendors clarified (partial clarification does not count), own metrics or vendor-provided data for effect measurement, and written code-review guidelines (with or without automated enforcement). It has a mean of 2.29 and a median of 2 out of a possible 6: most respondents report fewer than half of the measured indicators. Counting any policy (specific or general) instead raises the mean to 2.71 (median 3) and leaves the association of similar size ($\rho=0.393$, permutation $p<0.001$; organization level $\rho=0.395$, $n=42$, $p\approx 0.009$). Centralized decisions and vendor-provided data are counted because they show that the organization has taken a position on tool choice and on measurement, even if neither is evidence of outcome-oriented governance. Counting only a formal approval process and only own metrics gives a mean of 1.73 (median 2) and again a similar association ($\rho=0.389$, permutation $p<0.001$; organization level $\rho=0.467$, $n=42$, $p\approx 0.002$).

Intra-organizational disagreement. Among the 27 organizations with multiple respondents, the share with at least two differing answers ranges from 52% (policy, vendor clarity) to 96% (effect measurement). Table 1 compares each rate with the rate expected if answers were assigned to respondents at random.

Table 1: Share of the 27 multi-respondent organizations with any disagreement, observed and under a permutation null (10,000 draws; answers shuffled across all 115 respondents, preserving answer distributions and organization sizes).

Indicator	Observed	Null mean (95% range)	P(null $\geq$ obs)	P(null $\leq$ obs)
Adoption level (Q5)	85%	81% (67–93%)	0.374	0.809
Policy (Q14)	52%	79% (63–93%)	1.000	0.001
Support (Q15)	67%	84% (70–96%)	0.996	0.014
Decision-making (Q16)	74%	83% (70–96%)	0.944	0.148
Vendor clarity (Q17)	52%	77% (63–89%)	1.000	0.001
Effect measurement (Q18)	96%	83% (70–96%)	0.033	0.995
Code review (Q19)	70%	72% (59–85%)	0.693	0.516

Colleagues agree clearly more than chance on whether a policy exists and whether vendors are clarified, and somewhat more on support. Decision-making, code review and adoption level are close to the null. Only effect measurement exceeds it, and only marginally ($P\approx 0.03$, one of seven uncorrected comparisons). Policy, vendor arrangements and support therefore look like shared organizational knowledge; measurement is the one dimension where respondents in the same organization describe practice differently more often than random respondents would.

4.3. RQ3: Skill Loss Is the Most Frequently Selected Concern

Concern ranking. The concern question asked respondents to choose at most three items, so the percentages below show how often a concern was ranked among a respondent's top three, not how many hold it at all. Respondents selected a mean of 2.67 concerns, close to the cap of three; 79% (91/115) used all three slots. Skill loss (*kompetansetap*) was ranked among the top three by 71% (82/115, 95% CI: 62–79%), exceeding code quality at 57% (65/115, 95% CI: 47–65%), hallucinations at 40% (46/115, 95% CI: 32–49%), security at 34% (39/115, 95% CI: 26–43%), and privacy at 27% (31/115, 95% CI: 20–36%). Cost was selected by 17% (20/115), intellectual property by 7% (8/115), and workflow disruption by 7% (8/115). No respondent ranked vendor lock-in among their top three; this shows that it was outranked by other concerns, not that respondents are unconcerned about it. Figure 6 presents the full ranking of concerns.

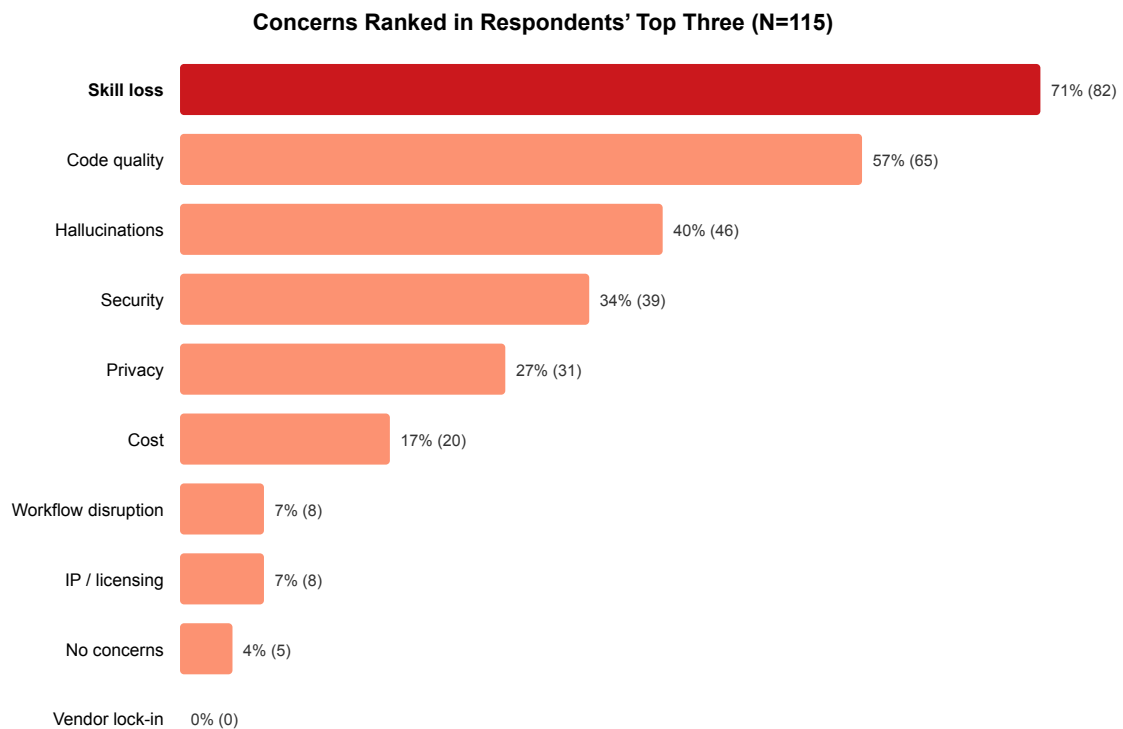


Figure 6: Concerns ranked among respondents' top three (N=115)

Role differences. We examined concern frequency across three recoded role groups — practitioners (n=56), technical leadership (n=44), and organizational leadership (n=15) — using Fisher's exact tests. No comparison reaches the Bonferroni-corrected significance threshold of $p < 0.0125$, and we therefore treat these differences as descriptive. Practitioners selected skill loss at a higher rate (79%) than technical leaders (61%) and organizational leaders (73%); the test of practitioners versus combined leadership (64%) is not statistically significant ($p = 0.104$). Code quality was selected at similar rates by practitioners (59%) and technical leaders (59%), but less frequently by organizational leaders (40%); practitioners versus combined leadership: $p = 0.707$. Hallucination concern was highest among practitioners (46%), followed by technical leaders (39%) and organizational leaders (20%); practitioners versus combined leadership: $p = 0.187$. Security concern showed a different pattern, with organizational leaders selecting it most often (47%) compared with practitioners (32%) and technical leaders (32%); practitioners versus combined leadership: $p = 0.844$. Figure 7 displays concern frequency across role groups.

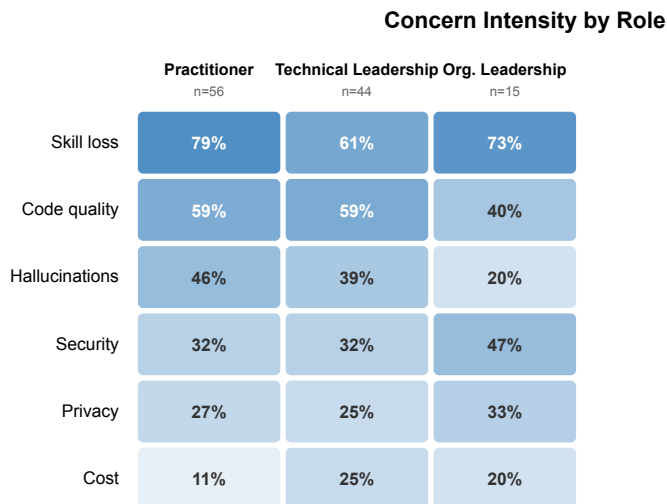


Figure 7: Concern intensity by role (practitioners n=56, technical leadership n=44, organizational leadership n=15)

Personal review stance. When asked about their personal approach to AI-generated code, 68% of respondents (78/115) report that they always or mostly review AI output before committing it: 24% always review and 43% mostly review. A further 16% perform a quick check, and 8% report that they usually accept AI suggestions. The remaining 9% (10/115) answered not applicable.

Perceived policy alignment. Respondents' views on their organization's policy are divided: 42% consider it appropriately calibrated, 20% view it as too liberal, 16% as too restrictive, and 18% report having no formed approach to compare against.

Public-sector-specific themes. Open-text responses (not quantified, and not included in the public dataset) raise themes particular to the public-sector context: concerns about data sovereignty when using cloud-hosted AI services, the need for transparency and public trust when government agencies produce AI-assisted code, dependency on US-based technology vendors, and the mismatch between rapid tool evolution and the pace of public-sector procurement processes.

4.4. Sensitivity Analysis

To assess robustness, we re-estimated key findings using only respondents who reported higher confidence in their organizational answers.

Adoption at the “officially approved” level or above is 78% across all respondents and 82% among those reporting certain confidence — a modest increase indicating stability. Policy coverage shows a stronger shift: 83% across all respondents rises to 93% among certain-plus-fairly-sure respondents and 94% among certain-only respondents. This suggests that respondents with lower confidence are disproportionately likely to report “no policy” or “unknown,” and that the 83% figure may underestimate policy prevalence (or, alternatively, that uncertain respondents accurately reflect gaps in policy communication). Shadow adoption measures are stable across all confidence filters: the broad measure ranges from 58% to 63% and the narrow measure from 30% to 32%, providing additional confidence in these estimates.

5. Discussion

We discuss the findings in light of our three research questions, situating them within this sample of technically engaged public-sector practitioners. Given the community-based sampling frame, we avoid generalizing to the Norwegian public sector as a whole.

5.1. Adoption Outpaces Governance

The central finding in this study is the gap between adoption and governance. AI coding tools have been procured, policies have been written, and 92% of respondents (106/115) use AI coding tools personally. Yet structured support remains scarce — only 8% (9/115) report a structured support program — and effect measurement is weak, with just 11% (13/115) using their own metrics. Code review processes have not adapted: 62% (71/115) report no special handling for AI-generated code.

The moderate positive association between governance maturity and adoption level (Spearman $\rho=0.439$, permutation $p<0.001$) suggests that organizations further along in adoption tend also to have more governance mechanisms in place. However, the cross-sectional design precludes causal interpretation. Governance may enable adoption by reducing organizational friction, adoption may prompt governance as a reactive measure, or both may reflect a common underlying factor such as organizational digital maturity. The composite governance index (mean 2.29 out of 6) indicates that even in a sample drawn from a technically progressive community, most organizations satisfy fewer than half of the governance criteria we measured.

5.2. The Shadow Adoption Signal

Shadow adoption of AI coding tools in this sample is substantial, though its magnitude depends on how the concept is defined. The broad measure (62%, 60/96) includes any use of tools outside the approved list, while the narrow measure (30%, 29/96) excludes browser-based LLMs and open-source tools. These two metrics represent different governance risks. Browser-based LLM use — the most common shadow tool (41 users) — typically involves pasting code snippets into a web interface, a practice that predates AI coding tools and resembles existing Stack Overflow or documentation search behavior. CLI agents such as Claude Code (17 shadow users) are different: they operate within the local development environment and may process internal source code, configuration, and infrastructure definitions. The gap between Claude Code’s approval rate and personal use rate provides the clearest single-tool example of approval lists lagging behind actual practice. Conversely, Copilot shows the reverse pattern: on the common $N=115$ base, 82% report it as approved but 68% (78/115) use it personally. One plausible explanation is that organizational Copilot subscriptions are often tied to work accounts and cannot easily be used on personal or side projects, making personal use lower than the approval rate would suggest — the friction of subscription management may itself suppress reported personal use.

The stability of both shadow measures across confidence filters (58–63% broad, 30–32% narrow) suggests these are not artifacts of respondent uncertainty. For organizations concerned about governance risk, the narrow measure offers a more defensible basis for action, as it isolates tools that interact directly with internal codebases.

5.3. Intra-Organizational Disagreement

Raw disagreement rates among the 27 multi-respondent organizations look high (52–96%), but most of this is what the any-disagreement measure produces by chance when several people answer a multi-option question. Against the permutation null (Table 1), respondents in the same organization agree *more* than random respondents on policy, vendor clarification and support. These answers behave like shared organizational knowledge: colleagues tend to know whether a policy exists and whether vendors have been clarified. For decision-making, code review and adoption level, the observed rates are close to chance and say little about communication either way.

Effect measurement is the exception. 96% of multi-respondent organizations contain at least two respondents who describe measurement differently, above the 83% expected by chance ($P\approx 0.03$, uncorrected). Combined with the 26% who do not know how effects are measured and the 11% reporting own metrics, this suggests that where measurement happens at all, it is not visible to many of the people using the tools. The evidence is modest and should be read as a hypothesis for the planned interviews and the next wave.

5.4. *The Agentic Transition*

The absence of autocomplete-only users in this sample is notable. All 106 users report employing capabilities beyond passive code completion — chat (82%), local agents (73%), debugging (66%), and planning (64%). Combined with the finding that 76% (81/106) use local or cloud agents, these results describe a sample that has moved to interactive and agentic use patterns.

This finding requires careful qualification. The survey does not distinguish a single experimental session from daily production use, and the technically engaged sampling frame likely overrepresents sophisticated usage patterns. These numbers should not be interpreted as representative of the broader public sector. They do, however, confirm that among early-adopting practitioners, agentic capabilities are not an edge case — they are the reported norm.

5.5. *Skill Loss as a Professional Concern*

Skill loss was the most frequently selected concern (71%, 82/115), exceeding code quality (57%), hallucinations (40%), security (34%), and privacy (27%). This ordering is striking: practitioners in this sample express more concern about their own capabilities than about the technical reliability or security properties of the tools. However, as noted in Section 6.2, social desirability effects may be at play — skill-loss concern may be a professionally acceptable response within this technically engaged community, and the 71% figure may partly reflect community norms rather than individual worry.

The skill-loss construct, however, is underspecified in the current instrument. It may capture genuine worry about deskilling (loss of ability to write code without AI assistance), concerns about over-reliance on AI output, anxiety about junior developer training, or broader professional identity concerns in the face of automation. Follow-up interviews, for which 51% of respondents have opted in (raw data; the opt-in column is not part of the public dataset), will be needed to disentangle these interpretations. The elevated rate among practitioners (79%) compared with technical and organizational leaders combined (64%) is suggestive but does not reach statistical significance (Fisher's exact $p=0.104$, Bonferroni threshold $p<0.0125$). Given the small organizational-leadership subgroup ($n=15$), the role comparisons should be treated as hypothesis-generating rather than confirmatory.

5.6. *The Measurement Paradox*

The combination of high policy coverage (83%) and low own-metrics adoption (11%) creates a measurement paradox: organizations have decided to use AI coding tools and written policies about their use, but most cannot independently assess whether the tools produce the intended effects. This pattern mirrors findings from our earlier study of internal developer platforms in the same sector (Flaatten, 2026), where platform teams similarly adopted tools without developing outcome metrics. In both cases, governance appears oriented toward permission-granting (procurement, policy documents) rather than ongoing assessment (measurement, structured support, communication).

The parallel between AI coding tools and internal developer platforms extends further. Both domains show governance trailing adoption, with measurement as the weakest dimension. AI tools, however, appear to follow a similar arc on a compressed timeline. Whether governance will catch up is an open question for longitudinal follow-up.

5.7. *Scope and Contribution*

This study captures a cross-sectional view of AI coding tool adoption, governance, and practitioner concerns among technically engaged public-sector practitioners. The most robust finding is that structured support — training, measurement, and governance communication — appears much less mature than individual tool use, even within a sample drawn from a technically progressive community. Headline adoption numbers (92% personal use, 76% agentic features) likely represent an upper bound for the Norwegian public sector. Organizations outside the Offentlig PAAS community, those with lower technical maturity, or those where AI tools are not yet relevant may show substantially different patterns. The contribution lies not in prevalence estimation but in identifying the structural

gap between tool procurement and operational governance, and in providing the first sector-level empirical data against which future changes can be measured.

6. Threats to Validity

6.1. External Validity

The sampling frame is centered on the Offentlig PAAS community and related technical networks. This community comprises organizations with platform-engineering capacity, modern developer tooling, and active interest in technology adoption — characteristics that are not representative of the Norwegian public sector as a whole. Organizations with lower technical maturity, those where AI coding tools are irrelevant to operations, and those that have not engaged with community-of-practice networks are likely underrepresented.

Non-response bias presents a related concern. Individuals with experience of or interest in AI coding tools were more likely to respond to a survey about AI coding tools. The 92% personal adoption rate is a within-sample finding and should not be interpreted as a sector-wide prevalence estimate. Organizations that tried AI tools and abandoned them, were blocked by procurement or security constraints, or never considered adoption are likely missing from the sample — a form of survivorship bias that inflates apparent adoption rates.

6.2. Internal Validity

Both principal investigators are active members of the Offentlig PAAS community, which introduces insider-researcher positionality. This position provided domain expertise, survey access, and contextual understanding, but may have influenced survey framing, the communities reached during distribution, and the interpretive lens applied to results. We acknowledge this dual role and mitigate it through transparent reporting of methods, data, and analysis scripts.

Social desirability effects may also be present. The community norms of the sampling frame favor innovation and technical experimentation. Respondents may over-report adoption, exaggerate the sophistication of their tool use, or select professionally acceptable concerns (such as skill loss) over less socially sanctioned responses (such as confusion or reluctance to adopt). The cross-sectional design prevents us from distinguishing experimentation from sustained production use: a respondent who tried a local agent once and a respondent who uses one daily both appear identically in the data.

The clustered structure of the data — 115 respondents nested in 57 organizations — means that individual responses are not independent for organizational constructs such as policy, approval status, and governance practices. The effective sample size for organizational governance claims is closer to 57 than to 115, and the Spearman $\rho=0.439$ for the governance-adoption association may be inflated by within-organization correlation.

6.3. Construct Validity

Several of our key constructs require qualification. Shadow adoption, as operationalized through the broad measure (62%), combines low-risk browser-based LLM use with genuine policy bypass by CLI agents that process internal code. The narrow measure (30%), which excludes browser-based LLMs and open-source tools, is more defensible for governance-risk claims but may undercount shadow adoption of tools that do interact with internal code in ways not captured by our categorization. Both measures compare each respondent's personal tools with the approved list that the same respondent reports. Some apparent shadow use may reflect a respondent's unawareness of a tool's approval rather than use outside policy. As a robustness check, we compared each respondent's tools with the union of tools that any respondent in the same organization lists as approved. The broad measure is unchanged (62%, 60/96) and the narrow measure falls slightly (28%, 27/96); only 2 of the 17 Claude Code shadow users are in an organization where another respondent lists Claude Code as approved.

The “policy exists” measure may capture awareness of a document rather than operational governance; that colleagues agree on it more than chance shows the document is known, not that it is followed. Similarly, skill-loss concern, the most frequently selected item, is underspecified in the current instrument. It may capture deskilling, over-reliance, junior developer training concerns, professional identity threats, or general change resistance. Without follow-up, we cannot distinguish these interpretations. Agentic usage — reported by 76% of tool users — may range from a single experimental session to daily production use. The survey captures self-reported usage mode, not verified sustained practice. The intra-organizational disagreement measure uses an “any disagreement” criterion that is sensitive by construction. We therefore interpret it only relative to a permutation null (Table 1). The null treats respondents as exchangeable across organizations and does not model role differences in what people can know, so it is a reference point rather than a full model of expected disagreement.

6.4. Conclusion Validity

Several cross-tabulations involve small subgroups, most notably organizational leadership ($n=15$). Statistical power for detecting role-based differences is limited, and all Fisher’s exact tests are treated as exploratory. The Spearman $\rho=0.439$ for the governance-adoption association may be inflated by within-organization correlation in the clustered data (115 respondents in 57 organizations); the effective sample size for this test is closer to 57 than to 115. The organization-level estimate ($\rho=0.441$, $n=42$, permutation $p\approx 0.004$) is of similar magnitude, which suggests that within-organization correlation does not drive the association. The confidence indicators attached to key organizational questions (e.g., 49% certain, 41% fairly sure and 10% estimate for adoption level, of 115; one respondent left the adoption confidence blank, so $n=114$) provide partial mitigation by allowing sensitivity analysis, but they do not eliminate the underlying measurement uncertainty for respondents reporting on organizational characteristics they may not fully know.

6.5. Scope of Claims

This study is a cross-sectional snapshot, not a sector-wide prevalence estimate. Headline findings (92% adoption, 76% agentic, 62% shadow adoption) describe this sample of technically engaged practitioners and should not be extrapolated to the Norwegian public sector without qualification. The governance-adoption association ($\rho=0.439$) reflects cross-sectional correlation and does not support causal interpretation in either direction.

7. Conclusion

This study presents the first sector-level empirical examination of AI coding tool adoption, governance, and practitioner concerns in the Norwegian public sector, based on a cross-sectional survey of 115 respondents from 57 organizations.

RQ1 (Adoption). AI coding tool adoption in this sample is widespread at the individual level (92%, 106/115) and has reached organizational procurement in most represented organizations (78% at “officially approved” or above). GitHub Copilot dominates the approved tool landscape (82%), but substantial shadow adoption exists: 62% of respondents with approved tools also use unapproved tools (broad measure), and 30% use unapproved tools that directly interact with internal codebases (narrow measure). Agentic capabilities — local and cloud agents that autonomously plan and execute code changes — are reported by 76% of tool users, with zero respondents using autocomplete alone. These figures describe a technically engaged segment of the sector and likely represent an upper bound for public-sector adoption overall.

RQ2 (Governance). Governance structures exist but are shallow. While 83% of respondents report some policy and 51% report vendor/DPA clarification, only 8% report a structured support program, 11% have own effect metrics, and 62% apply no special code review for AI-generated code. Governance maturity is positively associated with adoption level (Spearman $\rho=0.439$, permutation $p<0.001$), but

the direction of this relationship cannot be determined from cross-sectional data. Measured against a permutation null, colleagues agree more than chance on policy and vendor clarification, so these are shared organizational knowledge. Effect measurement is the only dimension where they disagree more than chance, and only marginally (96% vs. 83% expected; $P \approx 0.03$, uncorrected).

RQ3 (Concerns). Skill loss is the concern most often ranked among respondents' top three (71%), exceeding code quality (57%), hallucinations (40%), security (34%), and privacy (27%). This result is unexpected: we expected security and privacy to rank highest, given the regulatory context. Role-based differences in concern frequency are suggestive but do not reach statistical significance. A majority of tool users report that they always or mostly review AI-generated code before committing it.

Practical implications. These findings suggest that public-sector organizations should move beyond "procurement-as-governance" — the assumption that approving a tool and writing a policy constitutes sufficient governance. Structured training and capability support should accompany tool deployment, rather than relying on informal peer channels. Organizations need to develop their own outcome metrics rather than depending on vendor-supplied data or informal impressions. Measurement in particular should be made visible to practitioners: it is the one governance dimension where colleagues in the same organization disagree somewhat more than chance would predict.

Future work. Three directions follow from this study. First, a second survey wave (planned for 2027) will enable longitudinal tracking of how adoption, governance, and concerns evolve as tools mature and organizational experience accumulates. Second, qualitative interviews — for which 51% of respondents have opted in (raw data) — will allow us to explore the mechanisms behind the patterns observed here, particularly the nature of skill-loss concerns and the reasons for intra-organizational governance disagreement. Third, the survey instrument itself requires refinement: the skill-loss construct should be decomposed into its constituent concerns, agentic usage should distinguish frequency and depth of use, and the measurement question should separate what is measured from who sees the results.

This study contributes the first sector-level empirical data on AI coding tool governance in the public sector. While the technically engaged sample limits generalizability, the structural pattern it reveals — widespread individual adoption paired with immature organizational support and measurement — provides a baseline for tracking how public-sector organizations respond as AI coding tools continue to evolve.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors conducted this study as part of their employment at the Norwegian Labour and Welfare Administration (Nav) and Statens pensjonskasse (SPK).

Declaration of Competing Interests

Both authors are employed by Norwegian public-sector organizations within the surveyed population. The first author serves as chairman of Offentlig PAAS, the not-for-profit community of practice through which the survey was distributed, and both authors are active members of the community. Section 5.6 cites the first author's unpublished study of platform engineering in the same sector (Flaatten, 2026).

Data Availability

The anonymized survey dataset and its codebook are available in the Offentlig PAAS open-data repository: <https://github.com/offentlig-paas/open-data/tree/main/state-of-ai-agents/2026>. This paper is published at <https://offentlig-paas.no>. To protect participants, as promised in the consent text, the public file contains no organization identifier, not even a pseudonym, and roles are given as the three analytical groups used in the paper. The file contains categorical answers only: every free-text

“other” answer is replaced by its analysis category (for tools) or by “other”, and the timestamp, the free-text question on public-sector concerns (Q21), the device and collection-channel columns, contact information and the interview opt-in are removed. The three confidence columns are labelled Q5b, Q6b and Q14b. All respondent-level statistics and figures can be reproduced from the public file, except where rare categories were merged for privacy: the developer-count bands, and one approved open-source tool response that affects Figures 2 and 3. The original values for these are published in `respondent-level-aggregates.csv` in the same folder. Results that require grouping respondents by organization (the organization-level adoption distribution in §4.1, Table 1 and the organization-level Spearman correlations) are published as aggregate tables in `org-level-tables.csv` in the same folder. The analysis script (`analyze_wave1_v2.py`) and the figure script (`generate_svgs.py`), which uses the same code, are available at <https://github.com/offentlig-paas/open-data/tree/main/state-of-ai-agents/2026/scripts>.

CRediT Authorship Contribution Statement

Hans Kristian Flaatten: Conceptualization, Methodology, Investigation, Formal analysis, Writing – original draft. **Anders Olsen Sandvik:** Methodology, Validation, Writing – review & editing.

Declaration of Generative AI Use

During the preparation of this work the authors used Claude (Anthropic) for review, source checking and editing. The authors reviewed and edited the output and take full responsibility for the content and analysis of this publication.

References

- Anthropic. (2024, November 25). Introducing the Model Context Protocol. <https://www.anthropic.com/news/model-context-protocol> (accessed 12 October 2026).
- Becker, J. et al. (2025). Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity. *arXiv preprint*, arXiv:2507.09089.
- BlueOptima. (2024, November 25). Debunking GitHub’s Claims: A Data-Driven Critique of Their Copilot Study. <https://www.blueoptima.com/post/debunking-githubs-claims-a-data-driven-critique-of-their-copilot-study> (accessed 11 October 2026).
- Digdir — Digitaliseringsdirektoratet. (2024–2026). Veiledning for ansvarlig bruk og utvikling av kunstig intelligens. First published 6 June 2023 (open beta). <https://www.digdir.no/kunstig-intelligens/veiledning-ansvarlig-bruk-og-utvikling-av-kunstig-intelligens/4601> (accessed 12 October 2026)
- Digitaliseringsstyrelsen. (2024, January). Guide til offentlige myndigheter om ansvarlig anvendelse af generativ kunstig intelligens. <https://digst.dk/media/29727/guide-til-offentlige-myndigheder-om-ansvarlig-anvendelse-af-generativ-kunstig-intelligens.pdf> (accessed 12 October 2026).
- DORA/Google Cloud. (2025a). State of AI-Assisted Software Development. <https://dora.dev/research/2025/dora-report/> (accessed 11 October 2026).
- DORA/Google Cloud. (2025b). DORA AI Capabilities Model. <https://dora.dev/ai/capabilities-model/report> (accessed 11 October 2026).
- Faros AI. (2025). AI Engineering Report 2025: The AI Productivity Paradox. <https://www.faros.ai/ai-productivity-paradox> (accessed 11 October 2026).
- Finansforbundet. (2024). Are Nordic Organisations Ready for AI? Think Tank Mandag Morgen. https://finansforbundet.dk/media/2z4mgccl/ai-i-norden_engelsk.pdf (accessed 12 October 2026).
- Flaatten, H. K. (2026). The Evolution of Platform Engineering in the Public Sector: A Longitudinal Study of Adoption, Maturity, and Motivations (2024–2026). Unpublished manuscript.

- Haag, S. & Eckhardt, A. (2017). Shadow IT. *Business & Information Systems Engineering*, 59(6), 469–473. doi:10.1007/s12599-017-0497-x.
- Jellyfish [Albarran, N.]. (2025, December 22). 2025 AI Metrics in Review: What 12 Months of Data Tell Us About Adoption and Impact. <https://jellyfish.co/blog/2025-ai-metrics-in-review/> (accessed 11 October 2026).
- Kitchenham, B. & Pfleeger, S. L. (2008). Personal Opinion Surveys. In *Guide to Advanced Empirical Software Engineering*. Springer.
- Molléri, J. S., Petersen, K., & Mendes, E. (2016). Survey Guidelines in Software Engineering: An Annotated Review. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Ciudad Real, Spain, 1–6.
- Norwegian Government. (2020, January 14). National Strategy for Artificial Intelligence. Ministry of Local Government and Modernisation. <https://www.regjeringen.no/en/documents/nasjonal-strategi-for-kunstig-intelligens/id2685594/> (accessed 12 October 2026).
- Orosz, G. & Nilsson, E. (2026, March 3). AI Tooling for Software Engineers in 2026. *The Pragmatic Engineer*. <https://newsletter.pragmaticengineer.com/p/ai-tooling-2026> (accessed 12 October 2026).
- Oslo kommune / Digitaliseringsetaten (formerly Oslo Origo). (2025–2026). Brolagt sti for kunstig intelligens — KI-verktøy portefølje og sandkasse for utviklere. <https://ki.utvikler.oslo.kommune.no/> (accessed 12 October 2026).
- Ox Security. (2025). The Army of Juniors: The AI Code Security Crisis. <https://www.ox.security/army-of-juniors> (accessed 11 October 2026).
- PwC. (2026). 29th Annual Global CEO Survey. <https://pwc.com/gx/en/news-room/press-releases/2026/pwc-2026-global-ceo-survey.html> (accessed 11 October 2026).
- Shah, M. (2025, October 3). AI Coding Tools: Enterprise Feature Gaps That Block Adoption. Augment Code. <https://www.augmentcode.com/guides/ai-coding-tools-enterprise-feature-gaps-that-block-adoption> (accessed 11 October 2026).
- Stack Overflow. (2023). 2023 Developer Survey. <https://survey.stackoverflow.co/2023/> (accessed 12 October 2026).
- Stack Overflow. (2024). 2024 Developer Survey. <https://survey.stackoverflow.co/2024/> (accessed 12 October 2026).
- Stack Overflow. (2025). 2025 Developer Survey. <https://survey.stackoverflow.co/2025/> (accessed 12 October 2026).
- Stray, V., Barbala, A., & Wivestad, V. T. (2025). Human-AI Collaboration in Software Development: A Mixed-Methods Study of Developers’ Use of GitHub Copilot and ChatGPT. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering (FSE Companion ‘25)*, Trondheim, Norway, 1325–1332. doi:10.1145/3696630.3730566.
- Stray, V., Brandtzæg, E. G., Wivestad, V. T., Barbala, A., & Moe, N. B. (2025). Developer Productivity With and Without GitHub Copilot: A Longitudinal Mixed-Methods Case Study. *arXiv preprint*, arXiv:2509.20353.
- TechCrunch. (2025, July 30). GitHub Copilot crosses 20M all-time users. <https://techcrunch.com/2025/07/30/github-copilot-crosses-20-million-all-time-users/> (accessed 11 October 2026).