Agentic SDLC in practice: the rise of autonomous software delivery

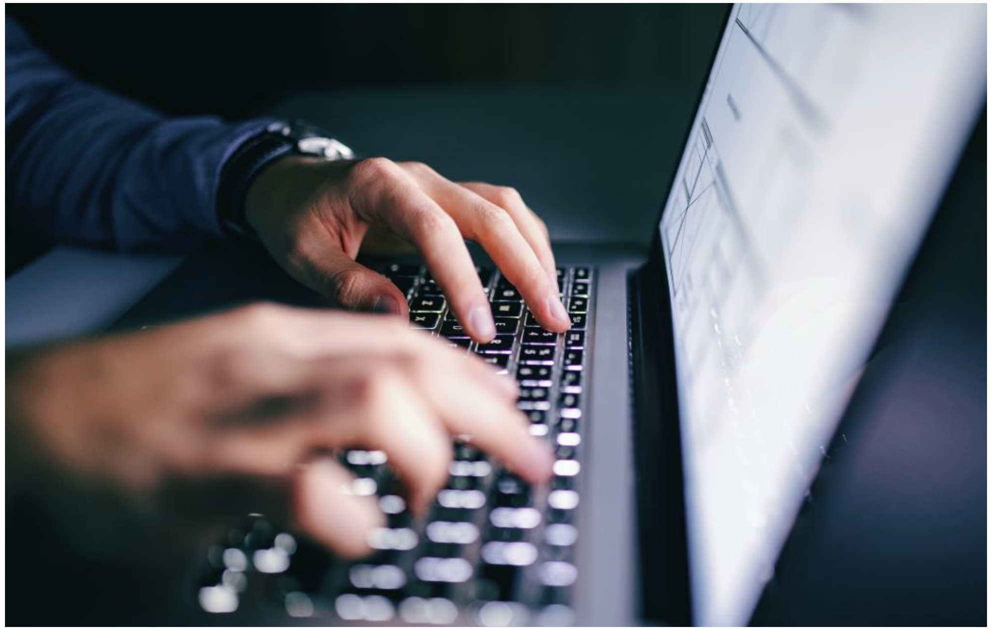
2026

Table of contents

Key terminologies introduced in this thought leadership

#	Terms	Definition
01	**Agentic roles**	New job profiles focused on orchestrating, governing, or optimizing autonomous GenAI agents (e.g., Prompt Engineer, AIOps Analyst)
02	**Agentic SDLC**	A software-delivery lifecycle where autonomous or oversighted AI agents plan, code, test, deploy and operate features with minimal human intervention, guided by high-level intent
03	**SDLC breadth**	The number of SDLC stages currently augmented by GenAI/AI agents
04	**SDLC depth**	The sophistication of AI augmentation within a single stage (e.g., from simple code-assist prompts to fully autonomous pull-request agents).
05	**Stage-coverage tiers**	We have defined 4-tier personas to present the current market efforts in leveraging GenAI in traditional SDLC lifecycle based on the SDLC breadth within each tier, those tiers are: Observer (≤1 stage AI-augmented), Experimenter (2-3 stages), Integrator (4-5 stages), Pioneer (≥6 stages)
06	**Vibe coding***	Informal term for generating or modifying code by describing the “vibe” or high-level intent in natural language, relying on LLM inference rather than exact specifications, this terms is widely adopted in the AI space and supported with many community members worldwide

* The term **“vibe coding”** was first coined by computer scientist and OpenAI co-founder **Andrej Karpathy**. He introduced it in a post on X (formerly Twitter) in February 2025, describing the idea of “fully giving in to the vibes” when using AI to generate and run code for quick, throw-away projects



01 Executive summary

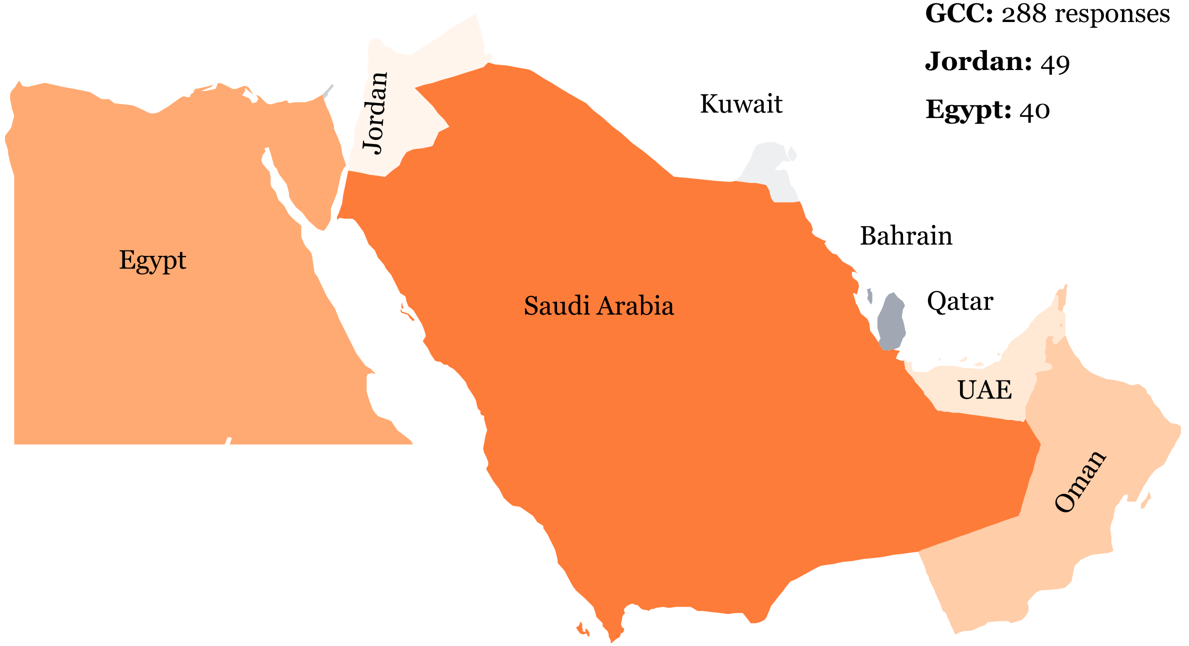
Middle East level insights

The adoption **hotspots** for AI within software development lifecycle are clearly indicating a strong shift soon impacting many roles in the play.

The rapid evolution of Generative AI (GenAI) technologies is reshaping the software development landscape, offering unprecedented opportunities for efficiency, innovation, and scalability. **This research explores the current adoption levels, benefits, challenges, and potential of GenAI tools** in the software development lifecycle **(SDLC).**

We have conducted a survey from **May to June 2025** covering GCC, Jordan, and Egypt, where we have received **377 responses with more than 40 responses from each mentioned country** (with some above 50 responses).

Survey participants from eight countries

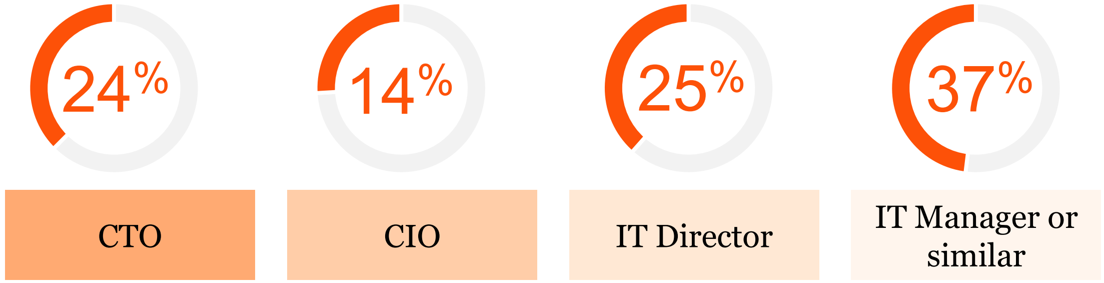


GCC: 288 responses

Jordan: 49

Egypt: 40

Primary audiences' roles who participated in the survey



Executive Summary

Software delivery is a race where speed and safety must work together. Every additional release, every defect prevented and every hour saved for engineers adds up across products, customers and cash flow. Artificial intelligence, especially GenAI, which produces new outputs like code, tests and documentation, is changing how this race is run. It is influencing the software development lifecycle (SDLC), by augmenting every stage – from requirements analysis and solution design to coding, testing, deployment, support and maintenance.

Across the Middle East, GenAI's role in the SDLC has moved from proof-of-concept to production-grade adoption, offering new levels of efficiency, innovation and scalability. To understand current penetration levels, benefits, challenges and long-term impact of GenAI in SDLC, PwC Middle East conducted a research study in the region, targeting Gulf Cooperation Council (GCC) countries, Jordan and Egypt

The study examines how GenAI is being adopted across the traditional SDLC and the impact it is already creating. It also outlines a future Agentic SDLC – defining the optimal blend of human and AI roles and identifying the evolving skills software teams will need to succeed in this new environment.

The transformation begins at the front end, where teams use GenAI to turn unstructured inputs into clearer requirements, functional specifications and rapid prototypes – shortening the journey from concept to buildable design. During coding and testing, AI agents generate workable code, surface defects earlier and improve test coverage, lifting quality throughout daily development. In deployment and operations, models support rollout planning and produce concise incident summaries that speed up resolution. And in maintenance, GenAI accelerates documentation, streamlines bug triage and helps teams manage backlogs more effectively. Together, these shifts signal a fundamental redefinition of how software is delivered.

The survey also explores how engineers and software developers are applying GenAI across the seven SDLC stages today:

- Ideation: Identifying business needs, defining project goals and gathering functional and technical requirements for what the software should do
- Design: Translating requirements into a system blueprint – outlining how the software will look, how users will interact with it and how different components will connect and operate
- Coding: Writing the actual code that brings the design to life, turning ideas and blueprints into working software components
- Testing: Detecting defects, assuring quality and validating alignment with requirements before release
- CI-CD (Continuous Integration and Continuous Delivery): Releasing the software to users or production environments, automating builds, integration and updates for faster, more reliable delivery
- Monitoring: Enhancing the software post-release to ensure it runs smoothly in real-world use
- Maintenance: Improving the software – fixing bugs, updating features, refactoring code and ensuring it remains efficient and secure over time

For chief information officers (CIOs) and chief technology officers (CTOs), the findings mark a clear inflection point. The region's software ecosystem is moving toward an agentic SDLC where governance, measurement and human-AI collaboration become core design principles. Organisations that invest early in structured observability, talent development and end-to-end AI integration will set the new performance benchmark for speed, quality and innovation in digital delivery.

Our analysis suggests:

Visionary look

Software development is pivoting from **handcrafted systems-of-record to agentic,** goal-driven applications that translate **intent directly** into **autonomous action**

Reality check (2025)

38 % of teams already **augment ≥6 SDLC stages** (we called them Pioneers). **Nearly 4 in 5 plan to raise GenAI investment within 24 months. 62 %** are exploring **agentic** AI apps

Business upside

84% of respondents report that embedding GenAI in their SDLC **significantly accelerates software delivery and enhances code quality,** while **Pioneers report 2×** release cadence, **90%** defect-rate reduction and measurable OPEX savings

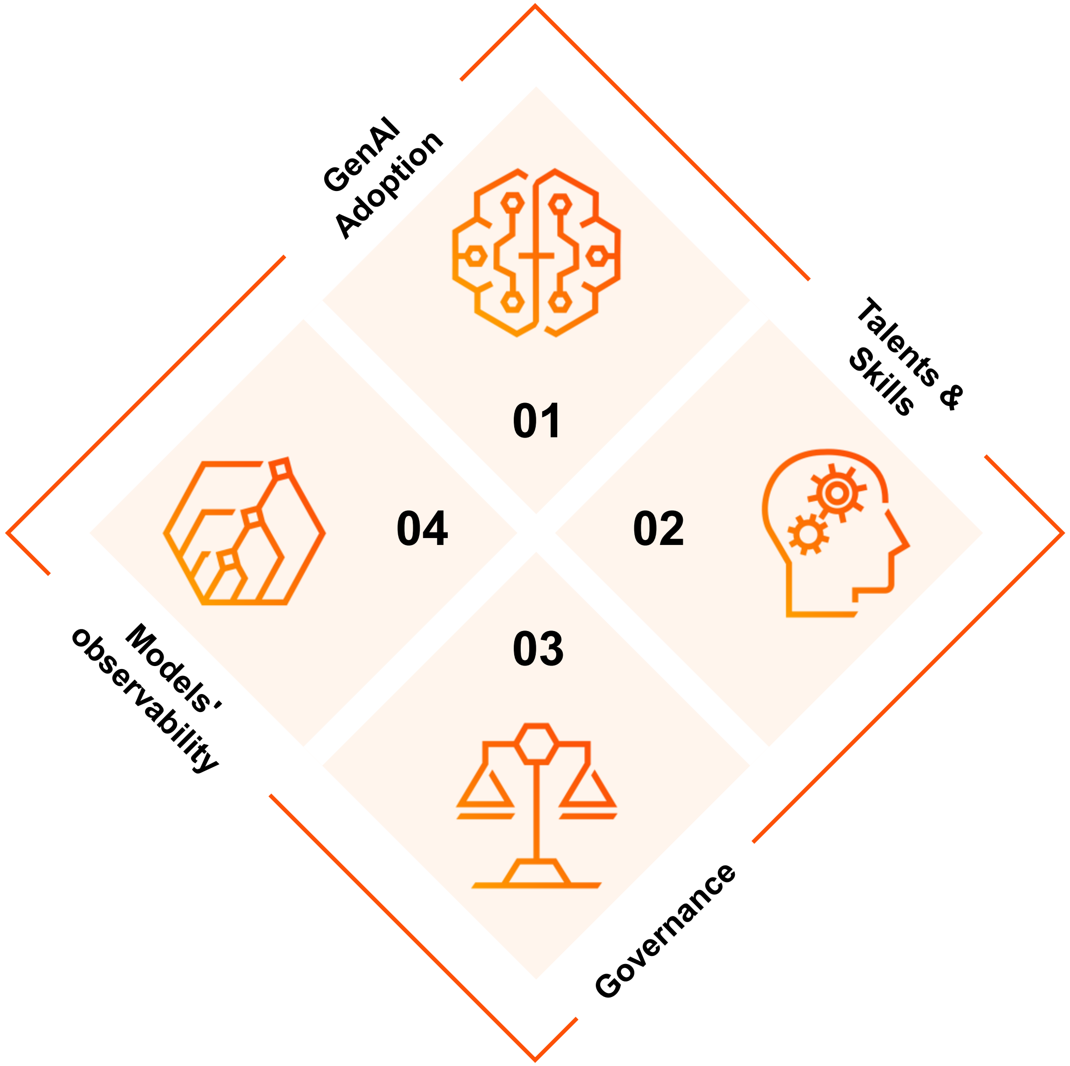
Predictable roadmap

By 2027 more than half of regional teams will run a fully Agentic SDLC; **by 2029 two-thirds will. Five enablers and six emerging roles** will shape the transition

Current barriers are open opportunities

Adoption challenges such as resource accessibility, compliance issues, and skill gaps persist; to harness GenAI's full potential, organizations must address these barriers and align their strategies with emerging trends

GenAI's trajectory is set, by decade's end **Agentic SDLC** will be mainstream across GCC & neighbors. Organizations that invest in **governance, observability** and **new talent models** today will capture the full speed-quality-cost dividend



02 Why GenAI is breaking the traditional SDLC

Github Octoverse 2024

AI Work Is Going Mainstream

Python #1, gen-AI repos +98% YoY, developer base surges

Stack Overflow 2025

Adoption Is Daily, Trust Lags on Risky Steps

51% of pros use AI daily; most won’t use it (yet) for deploy/monitor

METR 2025 RCT

Speed Gains Are Not Guaranteed

Experienced devs were ~19% slower with early-2025 AI tools—optimize for verification loops

JetBrains 2024

From Allowed Tools to AI-Native Products

80% of orgs permit 3rd-party AI; 18% of devs ship AI features

AI work is surging on GitHub while the global developer community expands; Python overtook JavaScript for the first time1

Why this breaks the traditional SDLC lifecycle?

1

Velocity

Contributions to GenAI projects jumped in 2024 **+59% YoY**; and **+98%** increase in the number of projects overall

This pace shifts toolchains and patterns faster than traditional release cycles can absorb

2

Talent

A swelling, AI-literate developer base (5.2B contributions across 518M projects)

This changes who builds software and how quickly they learn via AI

3

Stack drift

Python's rise to **#1 language** and Notebook usage **+92%** reflect a pivot to ML/agentic work, not just CRUD apps

518M Total projects on GitHub with 25% YOY growth	**5.2B** Contributions to all projects on GitHub in 2024	**>1M** Open source maintainers, verified students, and teachers have used GitHub Copilot at no cost
~1B Contributions to public & open source projects in 2024	**137K** Public GenAI projects with 98% YOY growth	**Python** Overtakes JavaScript as #1 language

[1]Source: GitHub Octoverse 2024: "AI is reshaping OSS and the dev base is exploding"

AI tools are now mainstream: **84%** use or plan to use them; **51%** of professional Devs use them daily; but most teams **still avoid AI** for **high-risk SDLC steps**[2]

Why this breaks the traditional SDLC lifecycle?

1

Governance gap

Developers report lower positive sentiment vs 2024 and say tools struggle on complex tasks; resistance is highest for **Deployment/Monitoring (76% don't plan to) and Project planning (69%);** pressure to redesign controls and QA

2

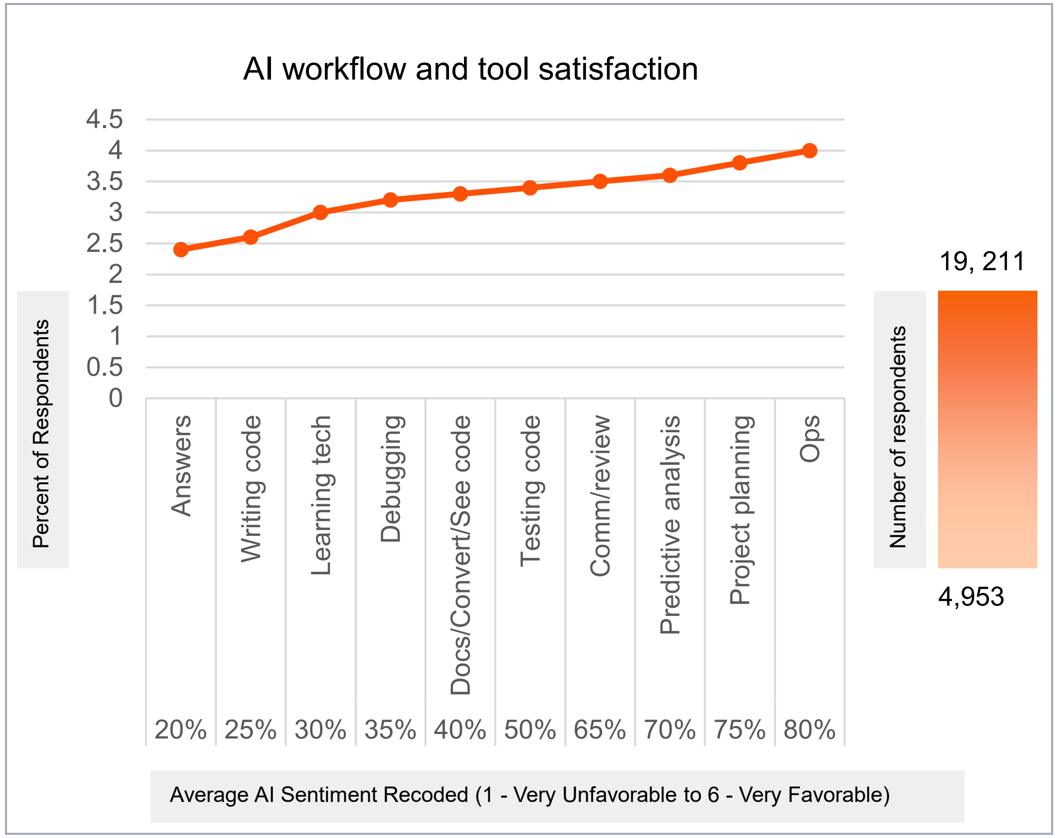
Talent & cost

Broad daily use **(47.1%)** means faster onboarding, fewer "blank page" costs, and shifting senior/junior leverage

3

Process redesign

AI is embraced for **writing/debugging/testing,** not yet for production gates—teams must re-cut responsibilities and sign-off models



2025 Developer Survey

[2]Source: Stack Overflow Dev Survey 2025: "AI is in daily use—yet trust and scope limits remain"

In a randomized trial with 16 experienced maintainers on their own repos, enabling early-2025 AI tools made them **19% slower** on real tasks. Expectation ≠ reality (yet)[3]

Why this breaks the traditional SDLC lifecycle?

1

Velocity variance

AI isn't "free speed"; time can shift into prompt iteration, verification, and fixing half-right code. Plan for **peaks and dips,** not linear gains

2

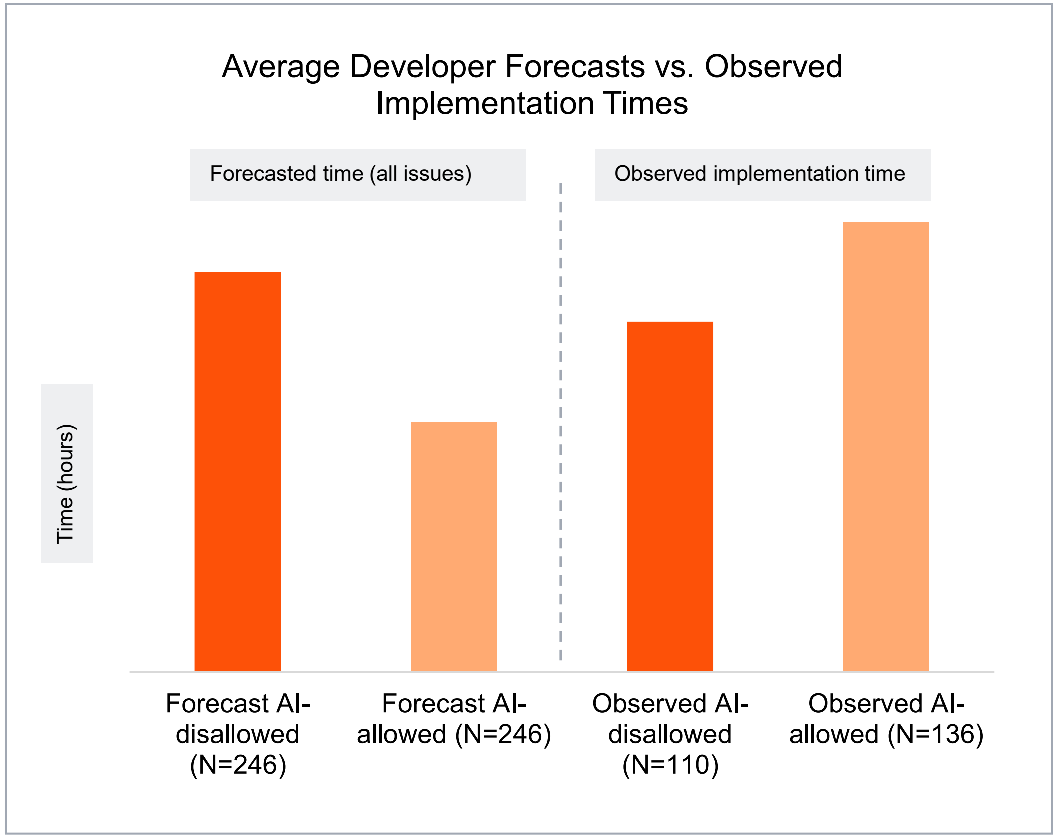
Cost & quality

Review/cleanup time is real; **governance and test automation must evolve** before autonomy pays off

3

Talent mix

The most value may come from AI-literate reviewers/architects ("AI overseers") rather than pure code generation



[3] Source: METR (July 2025 RCT): "Early-2025 AI slowed experienced OSS devs by ~19%"

80% of companies permit third-party AI tools; **18%** of developers already integrate AI **into their products**, AI is shifting from assistance to capability[4]

Why this breaks the traditional SDLC lifecycle?

1

Architecture

More products embedding AI means more **agentic flows,** non-determinism, and runtime evaluation loops, SDLC must include **model lifecycle** and **observability**

2

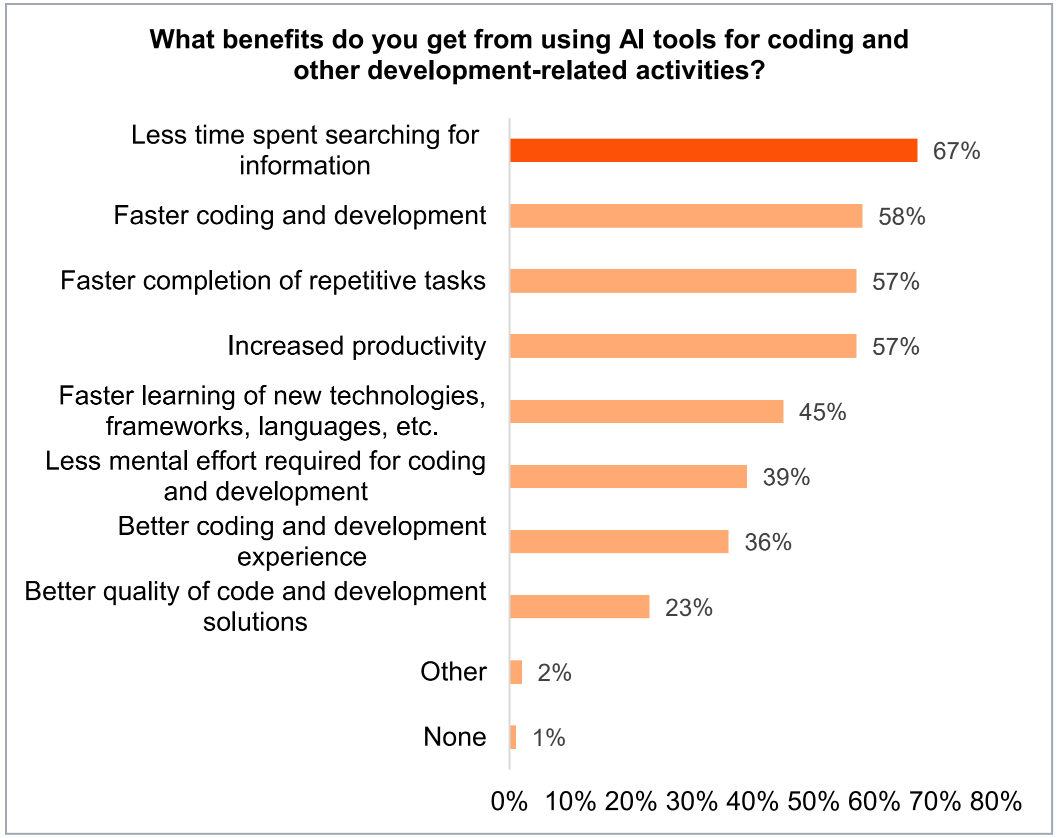
Talent & culture

With formal permissioning**, AI usage moves from shadow IT to standard practice;** upskilling and policy become gating items

3

Vendor ecosystem

Tooling and IDEs standardize around AI features, compressing certain roles while elevating evaluation/safety roles



What benefits do you get from using AI tools for coding and other development-related activities?

JETBRAINS

[4] Source: JetBrains State of Developer Ecosystem 2024: "Org guardrails are opening—AI is moving into products"

Software development leads in AI adoption: 36% of such roles employ AI for at least a quarter of tasks, yet only 4% use it extensively—and most of that usage is augmentative (57%), not outright replacement[5]

Why this breaks the traditional SDLC lifecycle?

1

Velocity + cost

Partial AI adoption speeds up routine tasks—but mixed usage means variable efficiency gains across the cycle

2

Talent shift

AI tools increasingly assist developers, enhancing work rather than subsuming jobs—creating a dual-mode workflow where human decision-making remains central

3

Hybrid workflows

SDLC frameworks must now account for collaboration between human developers and AI tools—not just handoffs between teams

[5]Source: Anthropic Economic Impact Index: "AI in Software Development—Primarily Augmentation, Not Automation"
Anthropic Economic Index: AI's impact on software development

AI-skilling yields tangible economic benefits; workers command a 56% wage premium, productivity rises are up to 4×, and job opportunities are increasing even in roles most exposed to automation[6]

Why this breaks the traditional SDLC lifecycle?

1

Cost and value

AI-augmented roles aren't cheaper—they're more valuable, signaling a shift to quality over quantity in engineering spend

2

Talent premium

Attracting developers now demands AI fluency and advanced capabilities—organizations must contend for premium-skilled talent

3

Role evolution

As AI handles routine tasks, humans will focus on high-value oversight, architecture, and system orchestration—**requiring redesign of role responsibilities and training**

[6] Source: PwC Global AI Jobs Barometer 2025: "AI Drives Greater Productivity, Wages, and Role Evolution"

3 common takeaways seen from different global parties that holds great reputation in software development along with massive access from worldwide developers and software engineers

Velocity

Octoverse shows the **scale and speed** (AI work +59% contributions; Python to #1). Stack Overflow shows **daily use now**. METR reminds leaders the **path to net speed** requires new QA, not just more prompts. Anthropic shows how partial AI boosts cadence.

We in PwC quantifies impact at scale

Talent & Quality

Daily AI use and permissive org policies change **skill profiles;** fewer rote tasks, more **AI oversight, evaluation, and integration.** Anthropic underscores a collaborative workflow.

We in PwC highlights the premium on AI-fluency

Cost/Risk/SDLC redesign

Productivity upside exists (many studies show speedups), but real-world variance and governance gaps (deployment, monitoring) mean **process redesign** is essential. Anthropic calls for hybrid process models.

We in PwC demands talent, structure adjustments, and governance by design

So is **AI faster or slower** in software development activities? **usage is mainstream**, but **production-grade speed** depends on governance, testing, and agent scaffolding; hence we are trying to put a forward thesis around **Agentic SDLC in this thought leadership**

03 Where the market is today

Why now? GenAI's tipping-point in SDLC

PwC's AI Jobs Barometer 2025 forecasts a **66 % faster skill-shift** in AI-exposed roles (function-calling, agent frameworks) compress innovation cycles

GenAI moving from PoC to production

●●○○○
●●○○○

4 in 10 regional teams already augment ≥6 SDLC stages

75% plan to raise GenAI spend within 24 months,

driving **2x** release-cadence uplift revealed in our cadence-vs-tier analysis

GCC transformation agendas reward efficiency

Pioneers achieve **44%** cost-goal uptake and **90%** **defect-rate reduction**

GCC digital-transformation agendas demand faster, cheaper, safer delivery

We have grouped our 377 survey respondents into **4 archetypes**, we called it "Stage-Coverage Tiers" as an index distribution to analyze quantitative similar behaviors of various teams participated in the survey

	01 Observers	**02 Experimenters**	**03 Integrators**	**04 Pioneers**
Definition	Teams that are slow to adopt and careful in exposing their SDLC to GenAI	Teams that are experiment the GenAI adoption on specific tasks but maybe not in a sustained way	Teams that have adopted GenAI into their SDLC workflows and are focusing on specific tasks in a sustained way	Teams that have been already adopting GenAI on every aspect or project in their SDLC workflows with full augmentation
Data slicing criteria	Based on the number of SDLC Stages augmented by GenAI **(0 to 1 stage)**	Based on the number of SDLC Stages augmented by GenAI **(2 to 3 stages)**	Based on the number of SDLC Stages augmented by GenAI **(4 to 5 stages)**	Based on the number of SDLC Stages augmented by GenAI **(6 to 7 stages)**

3.1 Adoption hotspots | Current state of GenAI adoption in SDLC

The seven SDLC stages across the software development lifecycle

01 **Ideation**: Identifying business needs, define project goals and gather functional and technical requirements for what the software should do

02 **Design:** Translating requirements into a system blueprint – outlining how the software will look, how users will interact with it and how different components will connect and operate

03 **Coding:** Writing the actual code that brings the design to life, turning ideas and blueprints into working software components

04 **Testing:** Detecting defects, assuring quality and validating alignment with requirements before release

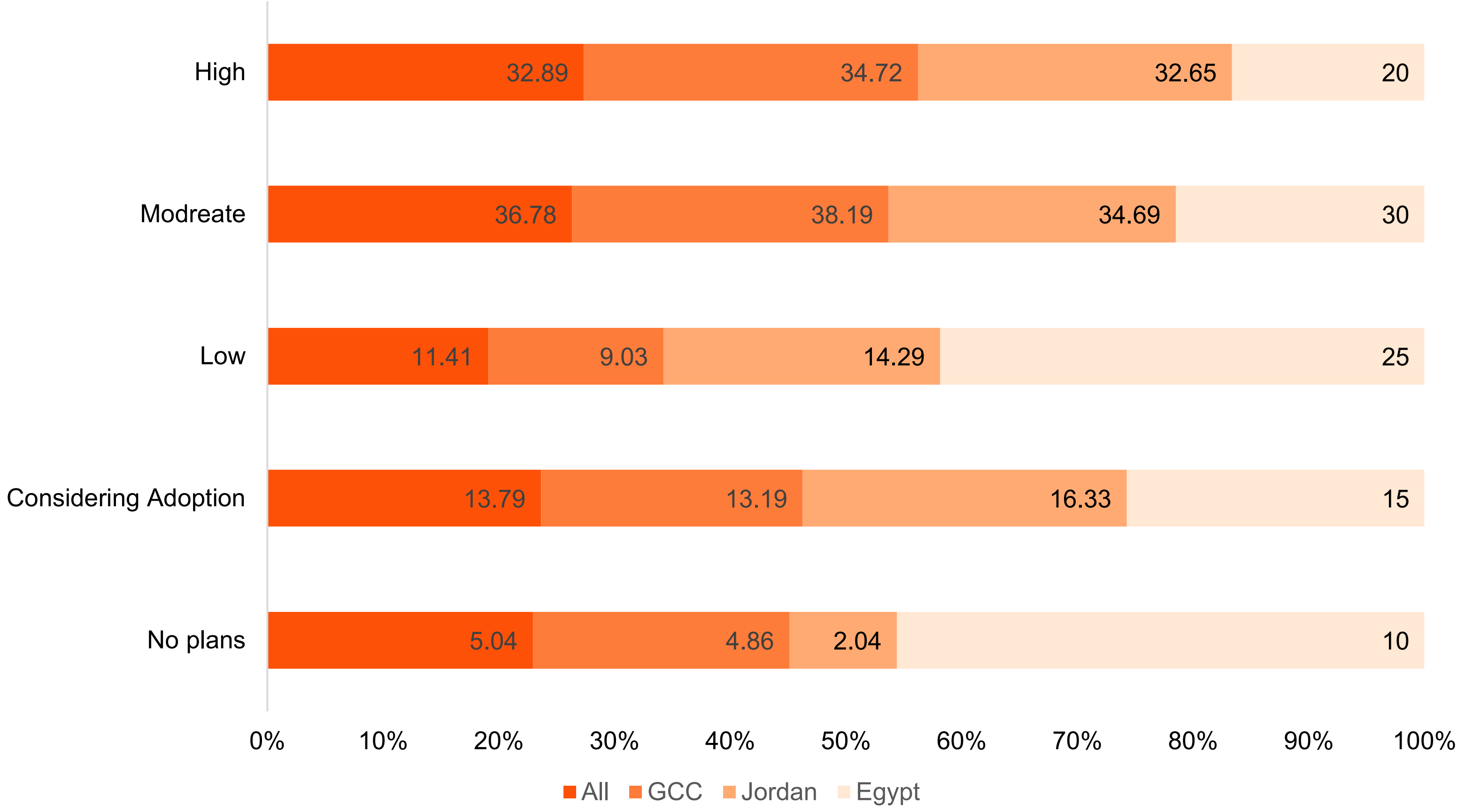
05 **CI-CD (Continuous Integration and Continuous Delivery):** Releasing the software to users or production environments, automating builds, integration and updates for faster, more reliable delivery

06 **Monitoring:** Enhancing the software post- release to ensure it runs smoothly in real-world use

07 **Maintenance:** Improving the software – fixing bugs, updating features, refactoring code and ensuring it remains efficient and secure over time.

Middle East software teams are past the curiosity phase; two-thirds now embed GenAI in day-to-day delivery

Indicator 1 - Overall adoption level (%) of GenAI in software development by Region



Key findings:

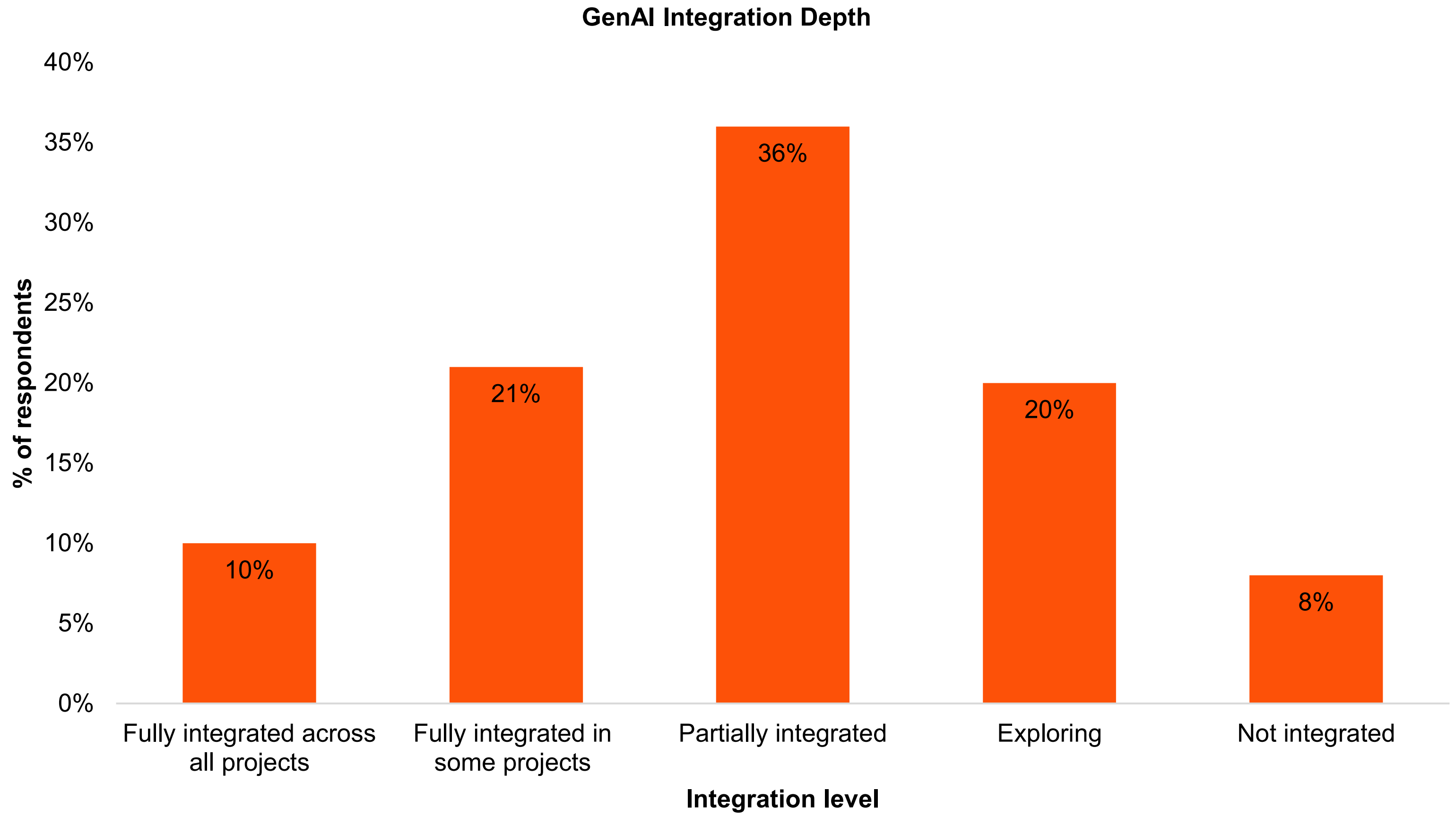
- **Tipping-point:** Nearly 70 % already sit in **"Moderate"** or **"High"** usage, signaling that GenAI is no longer an early-adopter play across our region
- **81.2 %** of respondents fall into High + Moderate + Low adoption triggering a tangible use of GenAI in software development

Notes:

- Country counts vary (GCC = 288 vs. Levant = 49 vs. NA = 40)
- "Moderate" = partial integration of GenAI within SDLC; "High" = extensive use across most SDLC stages

True end-to-end integration is still elusive: barely one in ten teams runs GenAI across all projects

Indicator 2 - How integrated are GenAI tools into your organization's SDLC?



Key findings:

- **Scarcity of full integration:** Only 10 % of teams run GenAI end-to-end on every software development project
- **Pipeline of near-pioneers:** Beyond the 10 % "all-projects" pioneers, an additional **21 %** already run at least some projects fully on GenAI, giving a combined **31 %** proven-integration cohort
- **Upgrade runway:** With 36 % still only partially integrated and 20 % merely exploring, over half the market sits on the verge of deeper adoption
- **Residual resistance:** 8 % remain entirely uninvolved or unsure; potential targets for awareness and capability building.

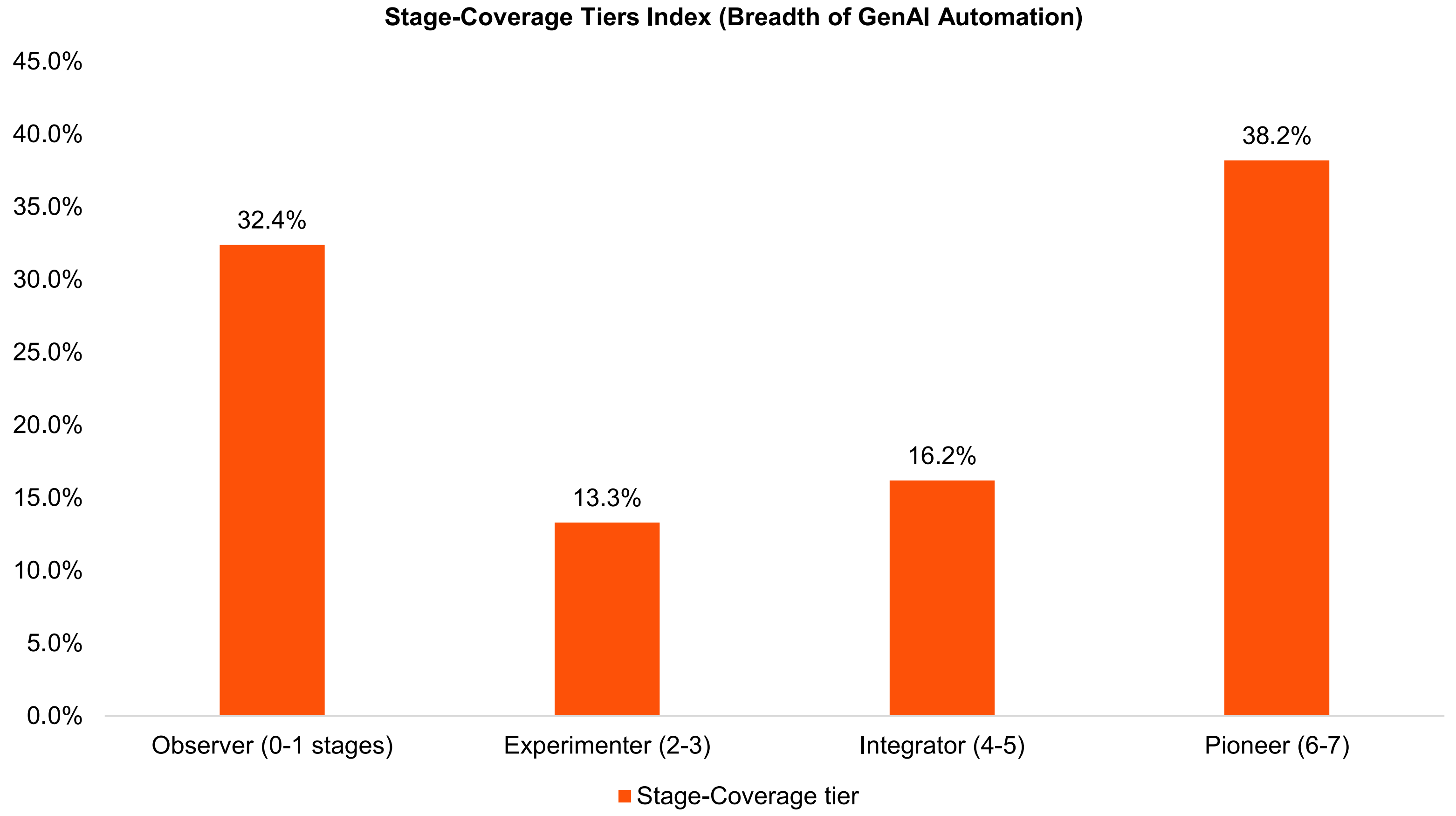
Notes:

- "Fully All Projects" denotes comprehensive GenAI use across every SDLC stage and every project; "Fully Some Projects" captures pockets of complete integration but to selected projects only

3.2 4 archetypes| Stage-coverage maturity tiers

Automation breadth is polarized: two-fifths of teams are already ‘Pioneers’ automating six or more SDLC stages, while a third remain ‘Observers’ with one or none

Indicator 3 - Stage-Coverage Tiers Index (Breadth of GenAI Automation)



Key findings:

- **Pioneers outnumber Observers:** 38 % vs 32 %, indicating a sizeable cohort with broad GenAI reach
- **Hollow middle:** Only 29 % sit in Experimenter or Integrator tiers, suggesting many leapfrog directly from minimal to near-full automation
- **Implication for capability building:** Training programs may need bifurcated tracks: “getting started” for Observers and “scaling architecture” for Pioneers

Notes:

- Compute shares depends on n=377, where frequency of each tier ÷ n, rounded to 0.1 %
- Equal weighting: all stages count equally, though in practice coding/test automation may matter more than deployment
- Threshold rationale: tiers were chosen to create intuitive quartiles; alternative cut-points would shift counts but not the polarized shape

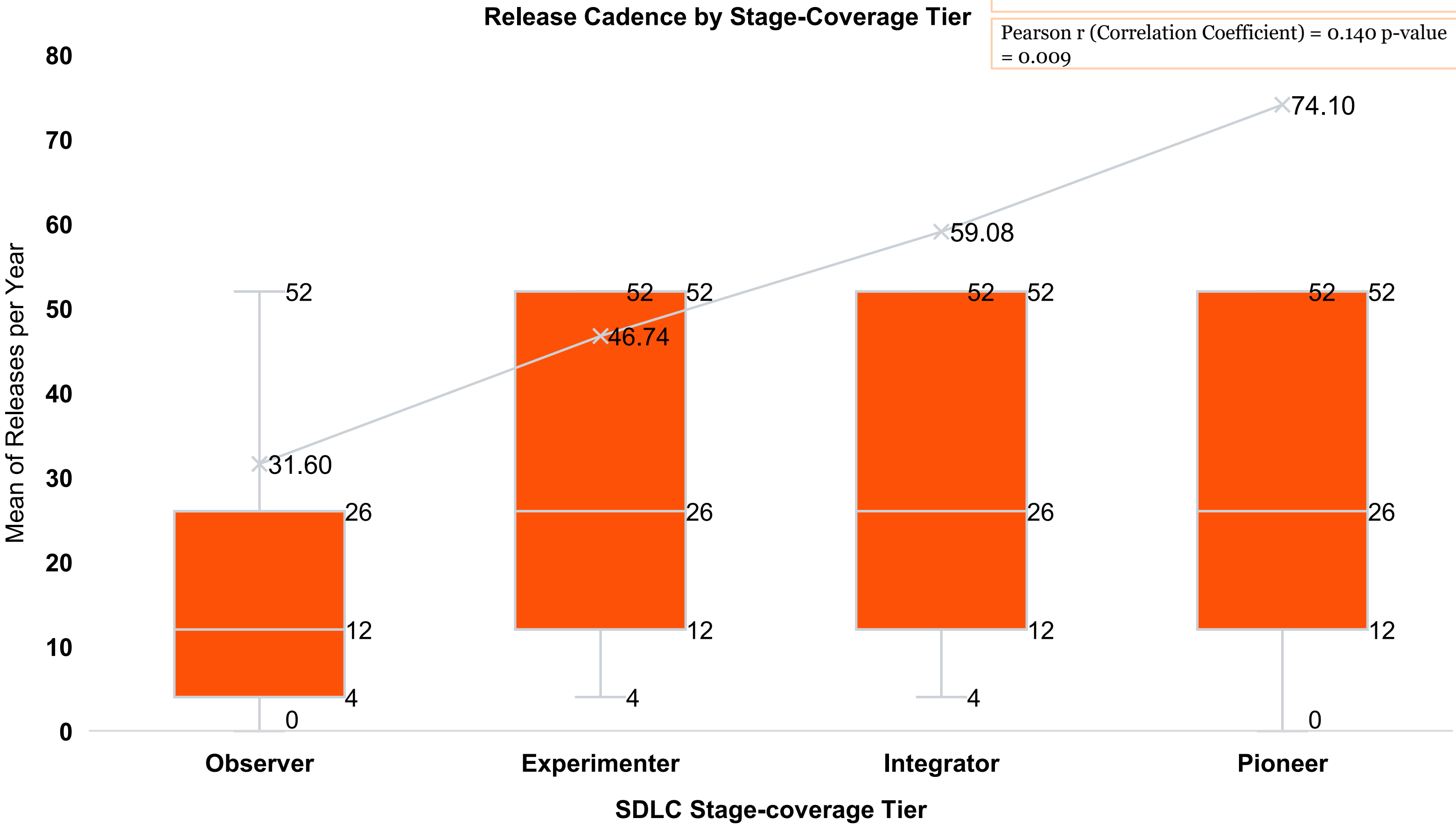
3.3 Business Impact | Tailored indicators and correlations

Automate more stages, ship more often: Pioneers average 75 releases per year vs 31 for Observers

Indicator 4 - Release Cadence vs Stage-Coverage Tier

Spearman ρ = 0.153, p = 0.004

Pearson r (Correlation Coefficient) = 0.140 p-value = 0.009



Key findings:

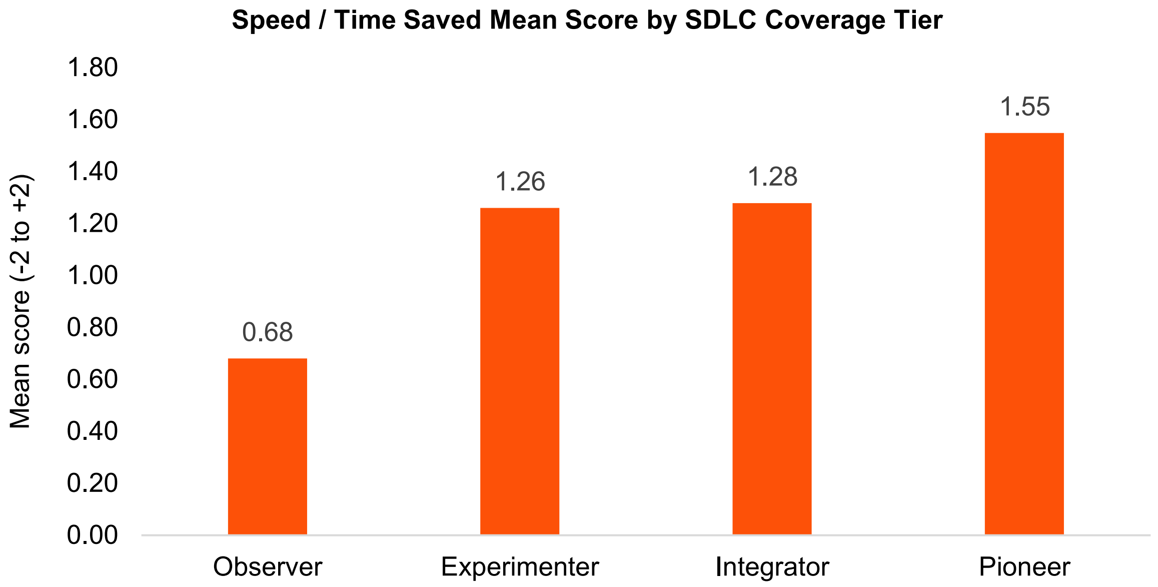
- **+44 releases/year** from lowest to highest tier
- **Spearman ρ = 0.15**, p = 0.004 — modest but significant monotonic lift
- **ANOVA** p = 0.067 (marginal), suggesting some variance

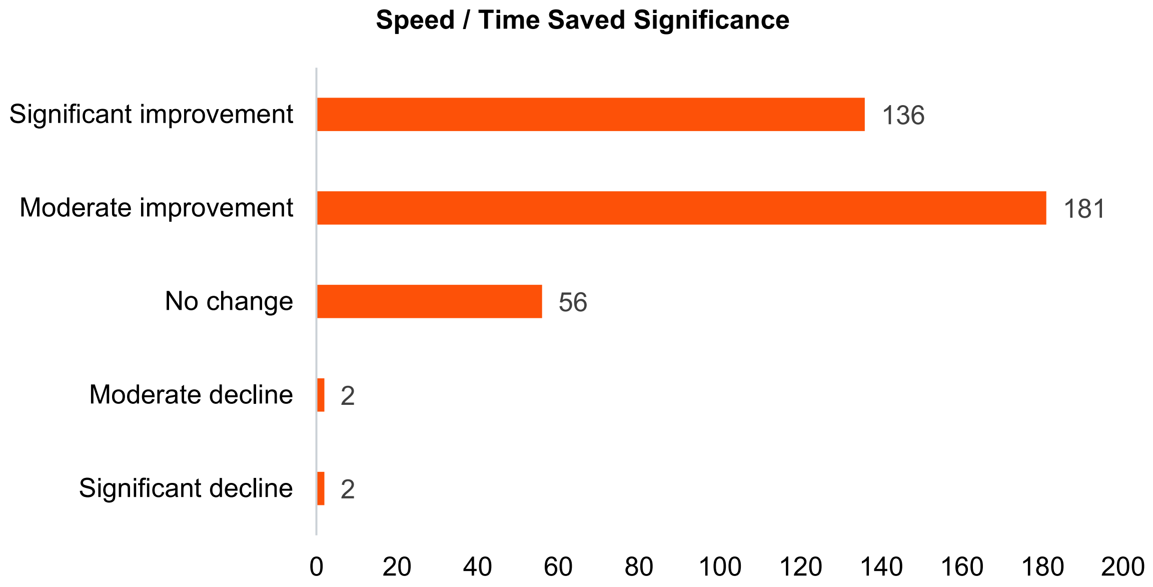
Notes:

- Whiskers capped at 1.5 × IQR; daily-release outliers omitted for visual clarity
- Frequency coded as Daily = 365, Weekly = 52, etc.; coarse buckets dampen nuance
- Median levels for Experimenter/Integrator converge at 26 (monthly), hinting at cadence ceiling until full Pioneer breadth is reached

84 % of respondents report that GenAI delivers a moderate-to-significant acceleration in software delivery, shortening time-to-market

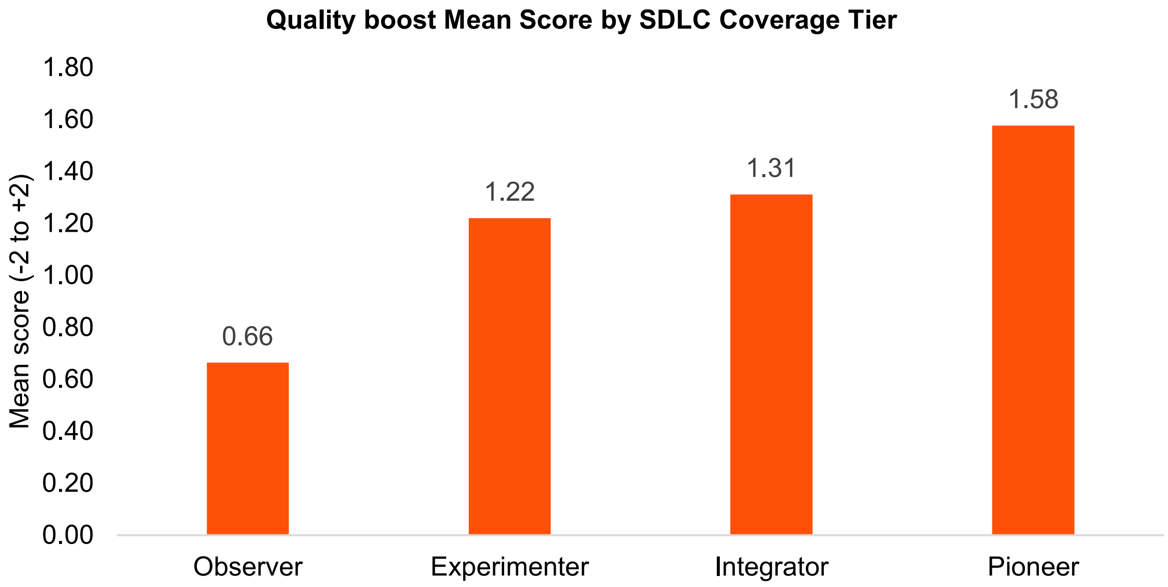
Indicator 5 - Time saved | Mean Score by SDLC Coverage Tier

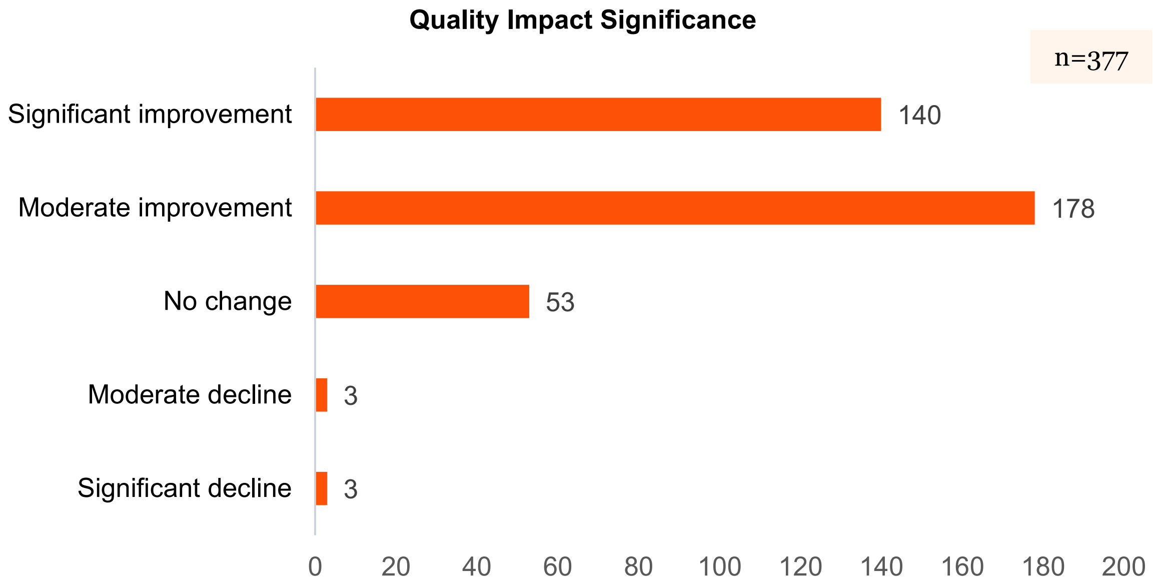




84 % of respondents see a moderate-to-significant uplift in code quality as GenAI is embedded into their SDLC

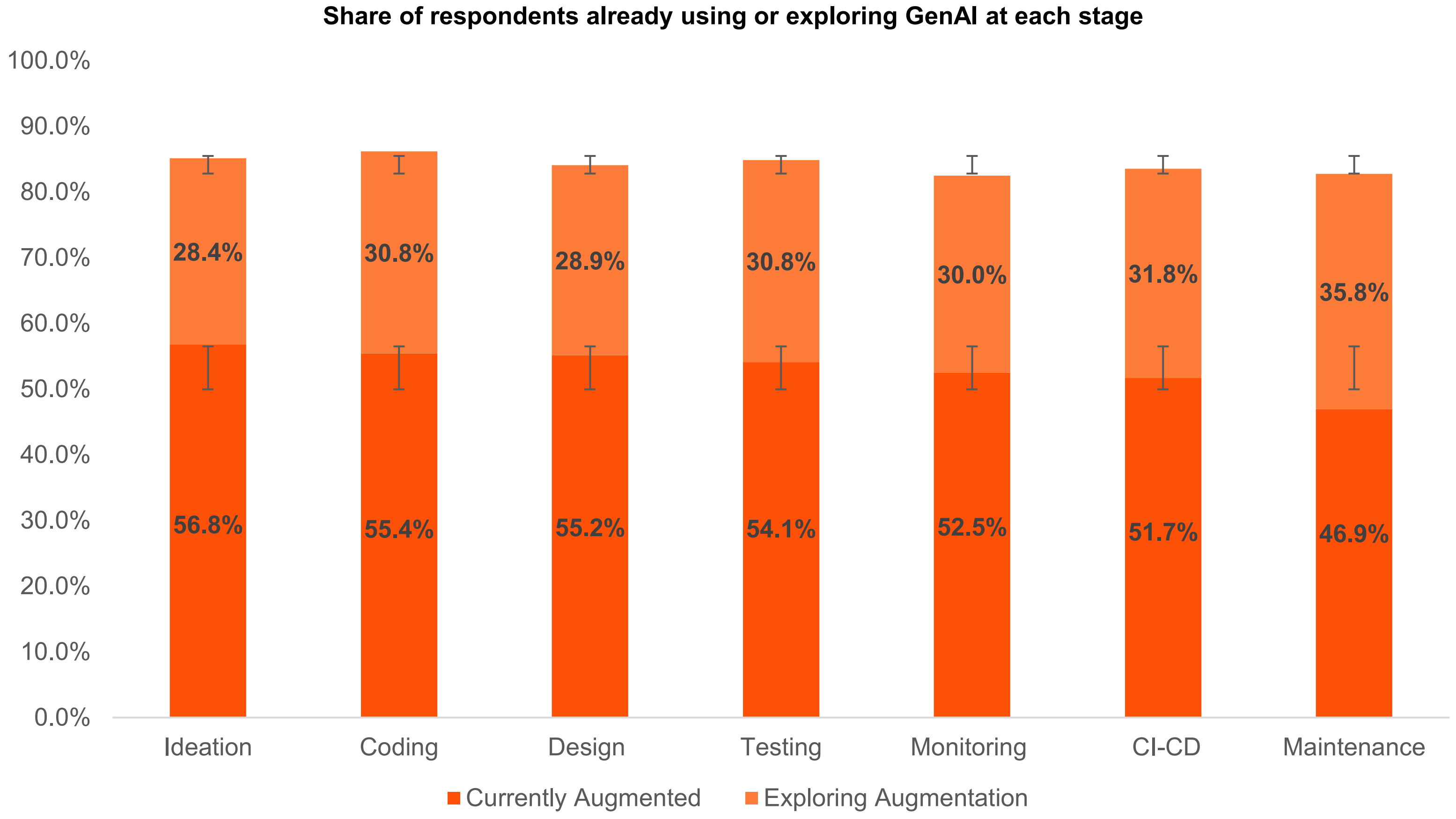
Indicator 6 - Quality Boost | Mean Score by SDLC Coverage Tier





GenAI is now a mainstream companion for upstream work—over half of all teams already use it in Requirements, Coding and Design—while post-release Maintenance still trails the field

Indicator 7 • Most Augmented SDLC Stages



Key findings:

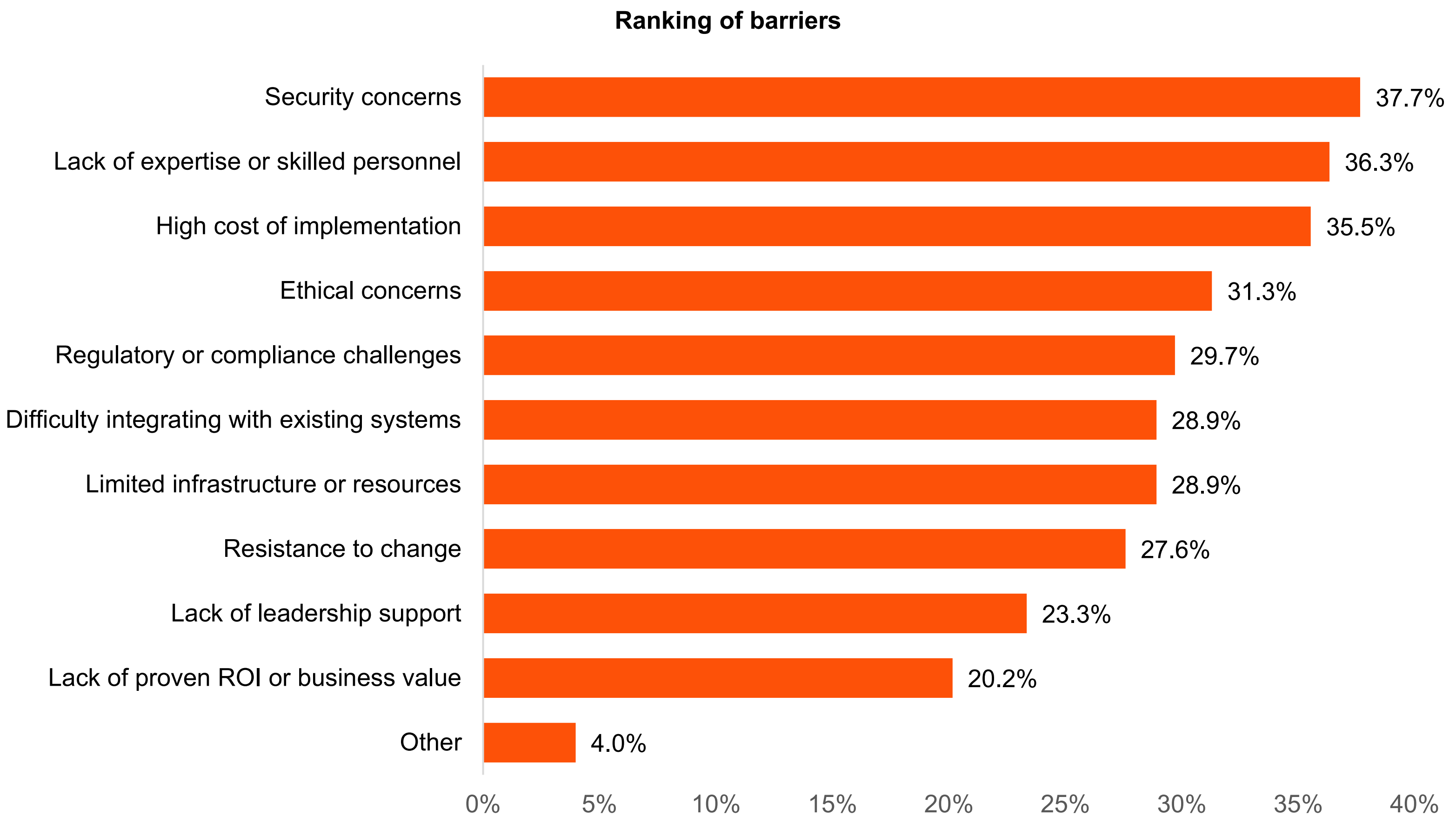
- **Upstream dominance –** Requirements/Ideation is now the most GenAI-augmented stage (~57 %), reflecting heavy prompt-engineering and user-story generation use cases
- **Code & Design neck-and-neck –** Both hover around 55 % adoption, confirming GenAI's strong foothold in core creation tasks
- **Downstream lag –** Only 47 % have automated Maintenance, even though we saw earlier it drives the biggest cadence uplift; a clear opportunity gap, Maintenance still trails at **47 %**, but boasts the **highest 'Exploring' share (36 %),** hinting at an imminent catch-up.
- **Healthy exploration –** (28-36) % are actively "exploring" in every stage, signaling a steady pipeline that could push most stages beyond 70 % adoption within a year

Notes:

- **Interpretation caution:** High "Yes" in early stages doesn't guarantee depth; some teams may be using GenAI in a limited, assistive manner
- Note that all stages now have **28–36 %** of teams in the exploring bucket

Security tops the regional worry-list, cited by nearly two in five respondents

Indicator 8.1 • Barrier Intensity: Overall Barrier Prevalence Cut (whole-sample %)



Key findings:

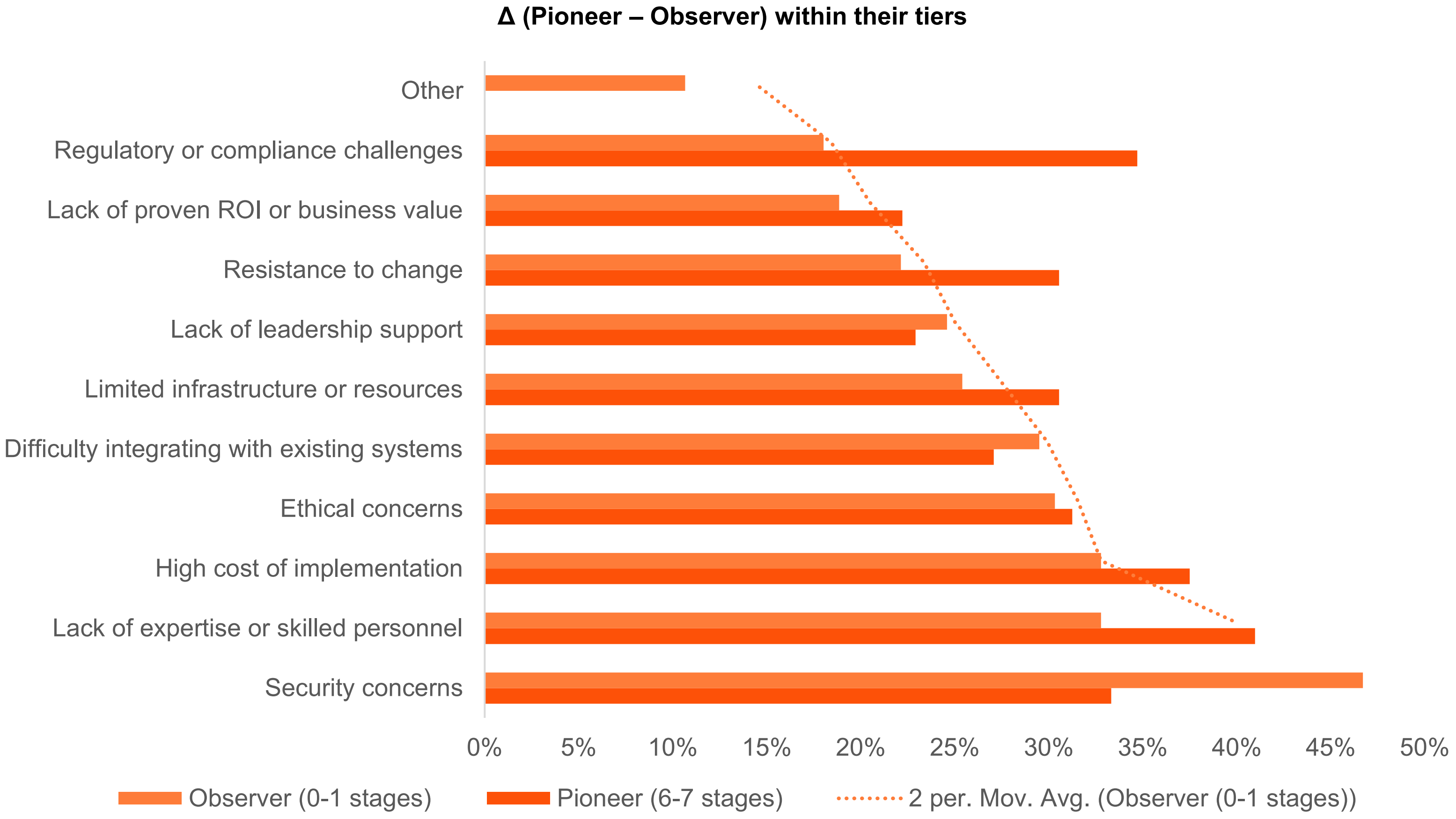
- **Security, skills, and cost** form a triad of universal barriers (≥ 35 % each) among the region
- Even lower-ranked barriers still appear **in ≥ 20 %** of answers, showing **GenAI friction is multifaceted**

Notes:

- Sample size n=377; number of teams is 377

Security paralyses observers; compliance and skills pinch the trail-blazers

Indicator 8.2 · Barrier Intensity: Barrier Gap by Tier Cut (within-tier %)



Key findings:

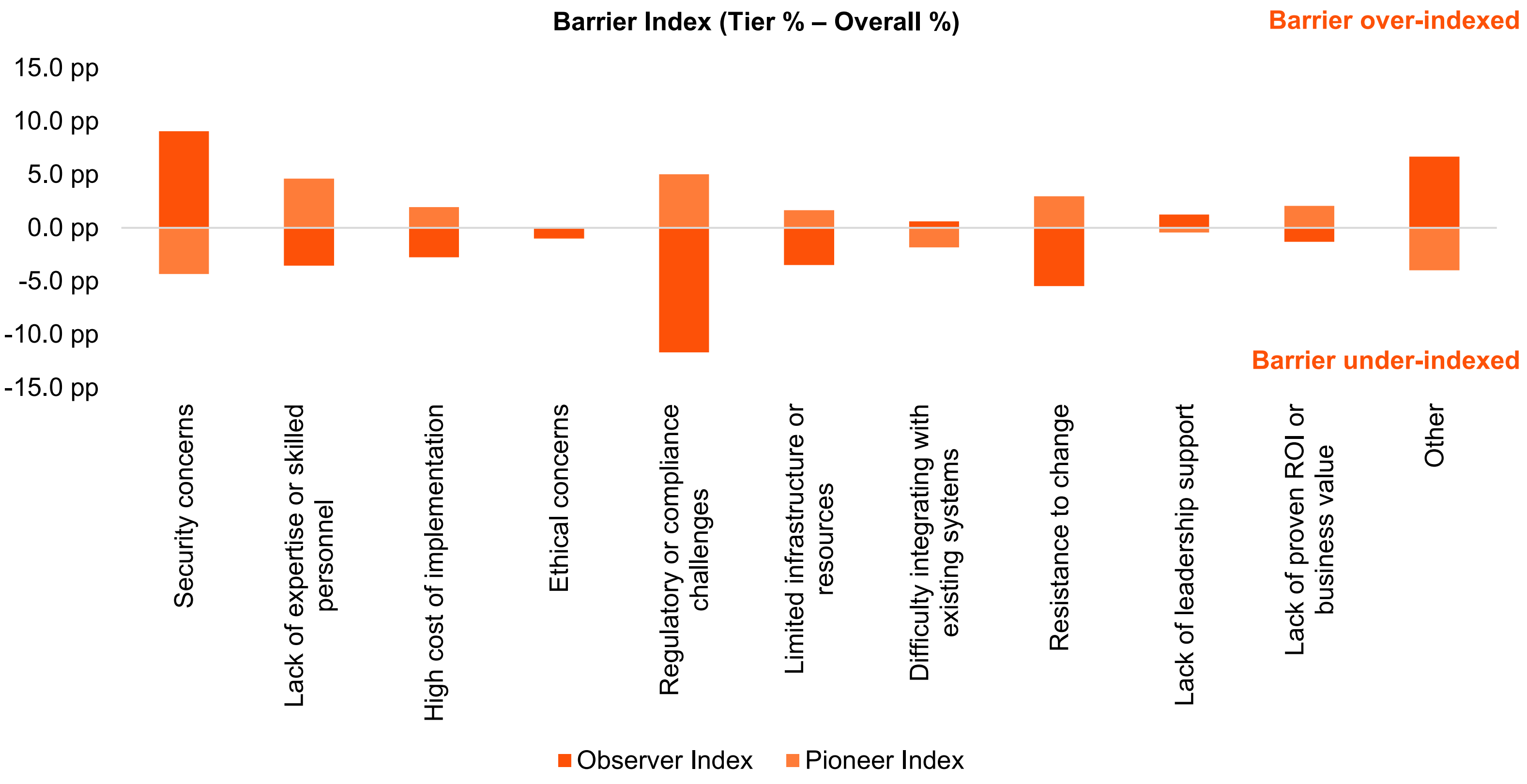
- **Security anxiety fades** as teams mature (47 % → 33 %)
- **Expertise & compliance rise** for Pioneers, revealing new pain once GenAI is scaled
- Cost is a **shared headache;** one-third of both tiers flag budget pressure

Notes:

- Percentages use each tier's own base (Observers = 122, Pioneers = 144). Values therefore cannot be compared directly to whole-sample bars

Observers over-index on security; veterans over-index on compliance and talent

Indicator 8.3 • Barrier Intensity: Barrier Index (Tier % – Overall %)



Positive values = barrier over-indexed in that tier; negatives = under-indexed in that tier

Key findings:

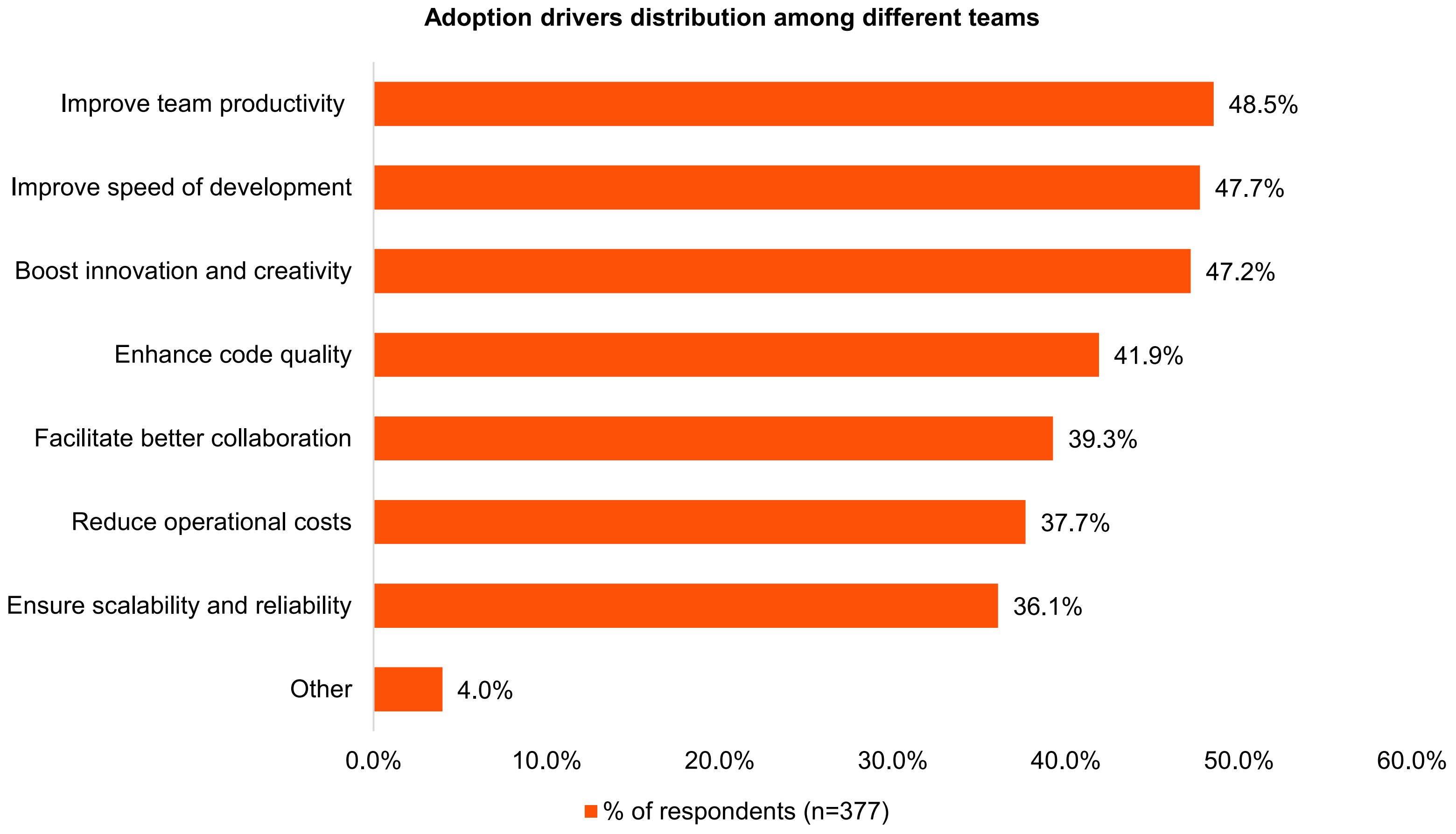
- Security's +9 pp index confirms it's **disproportionately an early-stage blocker**
- Compliance swings +5 pp for Pioneers—issues that barely register at the start become acute at scale
- Expertise gap flips from –3 pp to +5 pp, reinforcing the skills narrative

Notes:

- Index bars emphasize relative pain points; a small absolute barrier can still show a big positive index if it clusters in one tier

Improve team productivity, speed, and innovation form the top-3 motivators that pushes development teams to adopt GenAI in their SDLC

Indicator 9.1-Why teams adopt GenAI? Overall, Drivers Prevalence Cut (whole-sample)



Key findings:

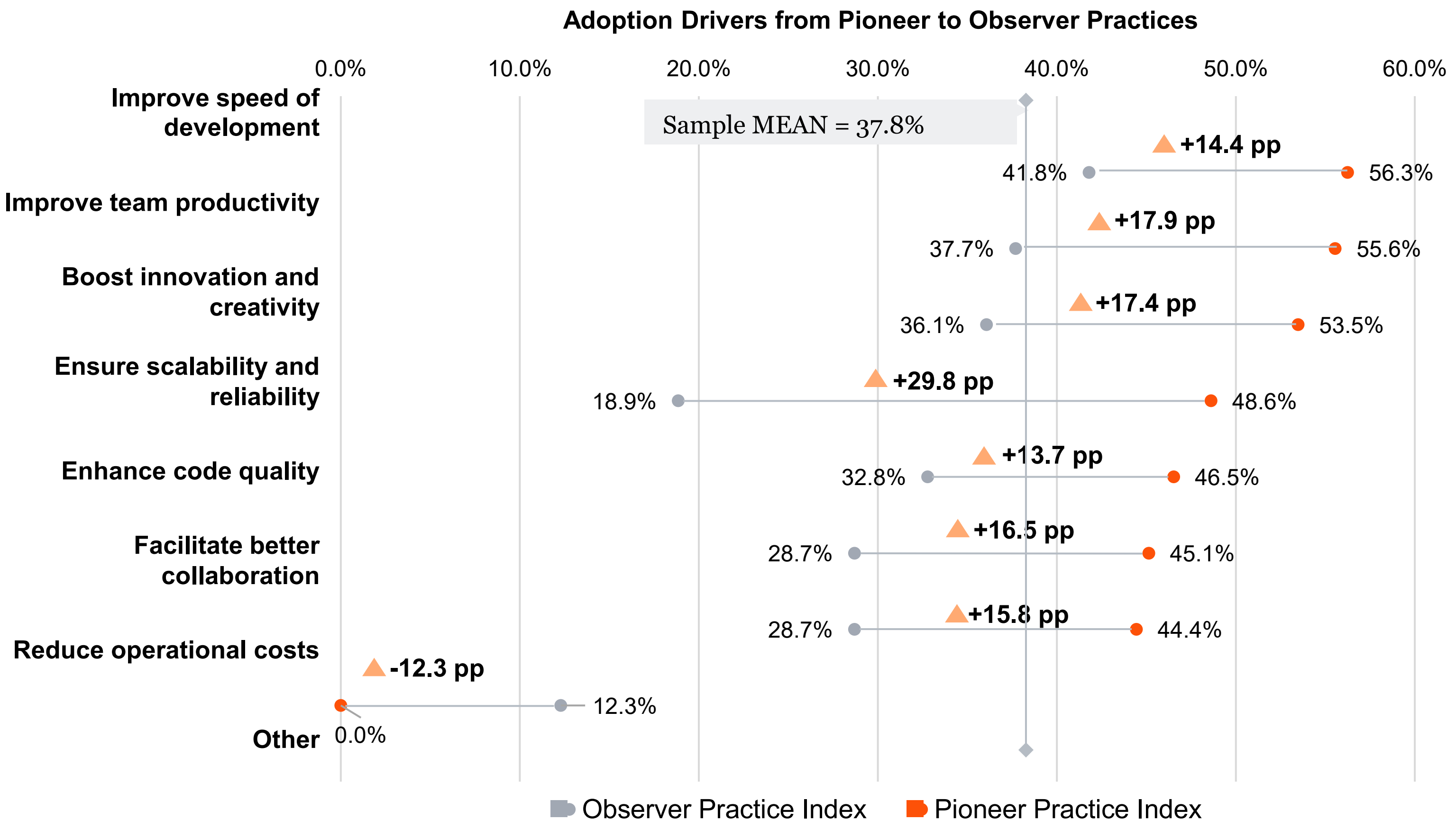
- Productivity, speed and innovation form the **top-3 motivators** (~48 % each)
- Cost-reduction is already on the radar for nearly 4 in 10 teams

Notes:

- **Multi-select question:** percentages don't sum to 100 %
- **Denominator:** Overall chart uses n = 377

Pioneer practice is over-index on every driver, while observer under-index on almost each driver

Indicator 9. 2 - Why teams adopt GenAI? Drivers By Tier (Pioneer vs Observer, (within-tier %)



Key findings:

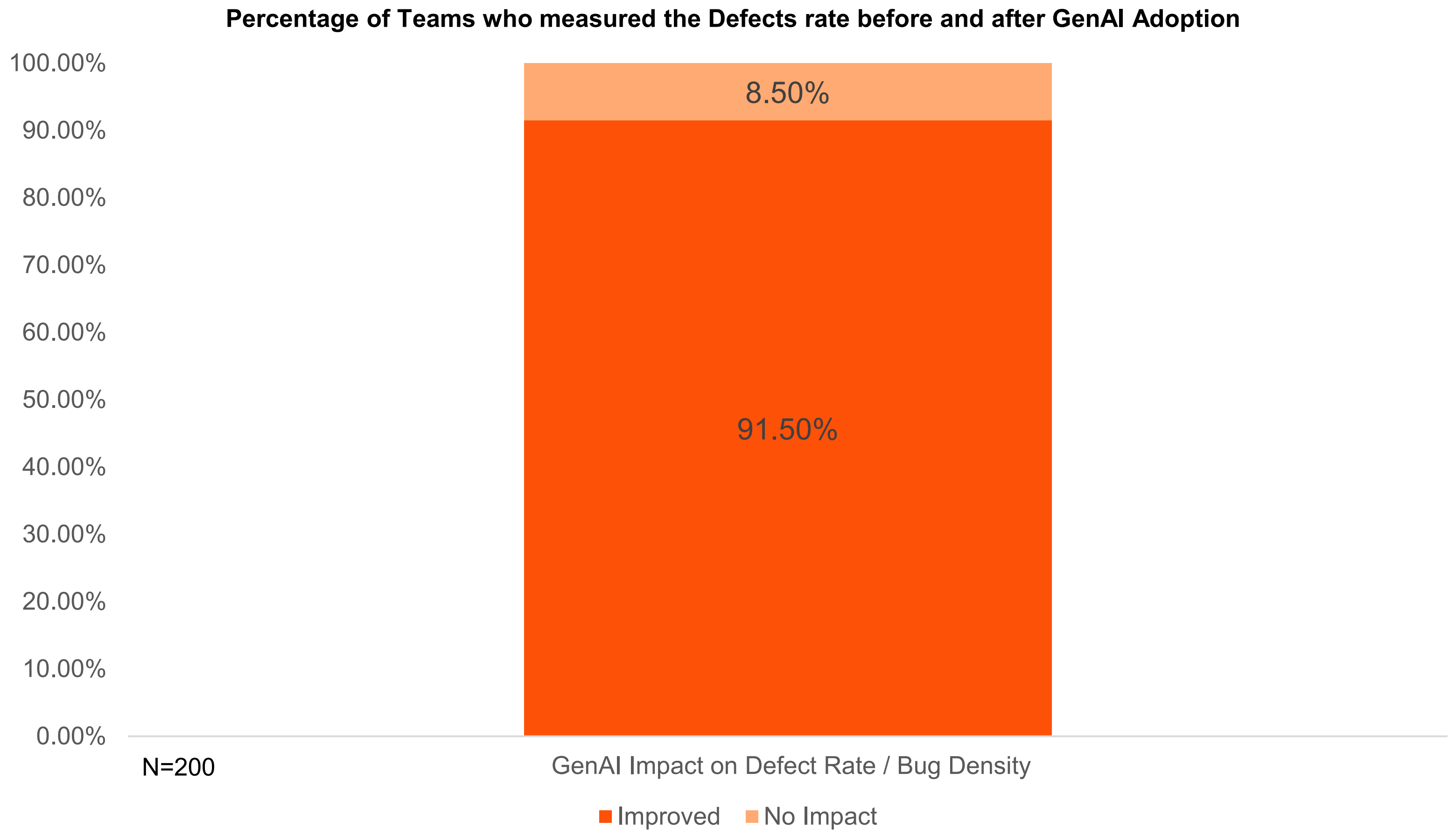
- **Pioneers over-index on every driver**, especially scalability (+30 pp) and productivity (+18 pp)
- Observers care least about scalability (19 %), suggesting they haven't yet felt scale pain
- **Pioneers over-index on every driver**; All Pioneer-Δ values ≥ +4 pp, Mature teams see GenAI as a multi-benefit platform, not a single-goal tool
- Scalability is the defining gap, +12.5 pp (Pioneer) vs –17.2 pp (Observer) Scale & reliability pain emerges only when GenAI touches 6-7 stages
- Observers are still chasing quick wins; Early adopters haven't yet translated vision into concrete benefits
- Cost-reduction rises with maturity; Once productivity/speed are proven, budget owners push for hard savings
- Innovation & collaboration lag among Observers; Limited breadth constrains users' ability to tap GenAI for creative or cross-team work
- **"Observers under-index across the board."** Their lower scores on every driver show that lip-service without real rollout returns little value.
- **Scalability is the watershed."** A 30-point swing between tiers signals where enablement efforts should focus next.

Notes:

- Chart uses each tier's own n (Observers = 122, Pioneers = 144)
- "Mature breadth unlocks new priorities." Pioneers pivot from speed to scalability, cost and enterprise-grade reliability.

When teams track defect rates, nine in ten teams who track bugs see fewer defects when leveraging GenAI in their SDLC stages

Indicator 10.1 - GenAI Impact on Defect Rate / Bug Density



Key findings:

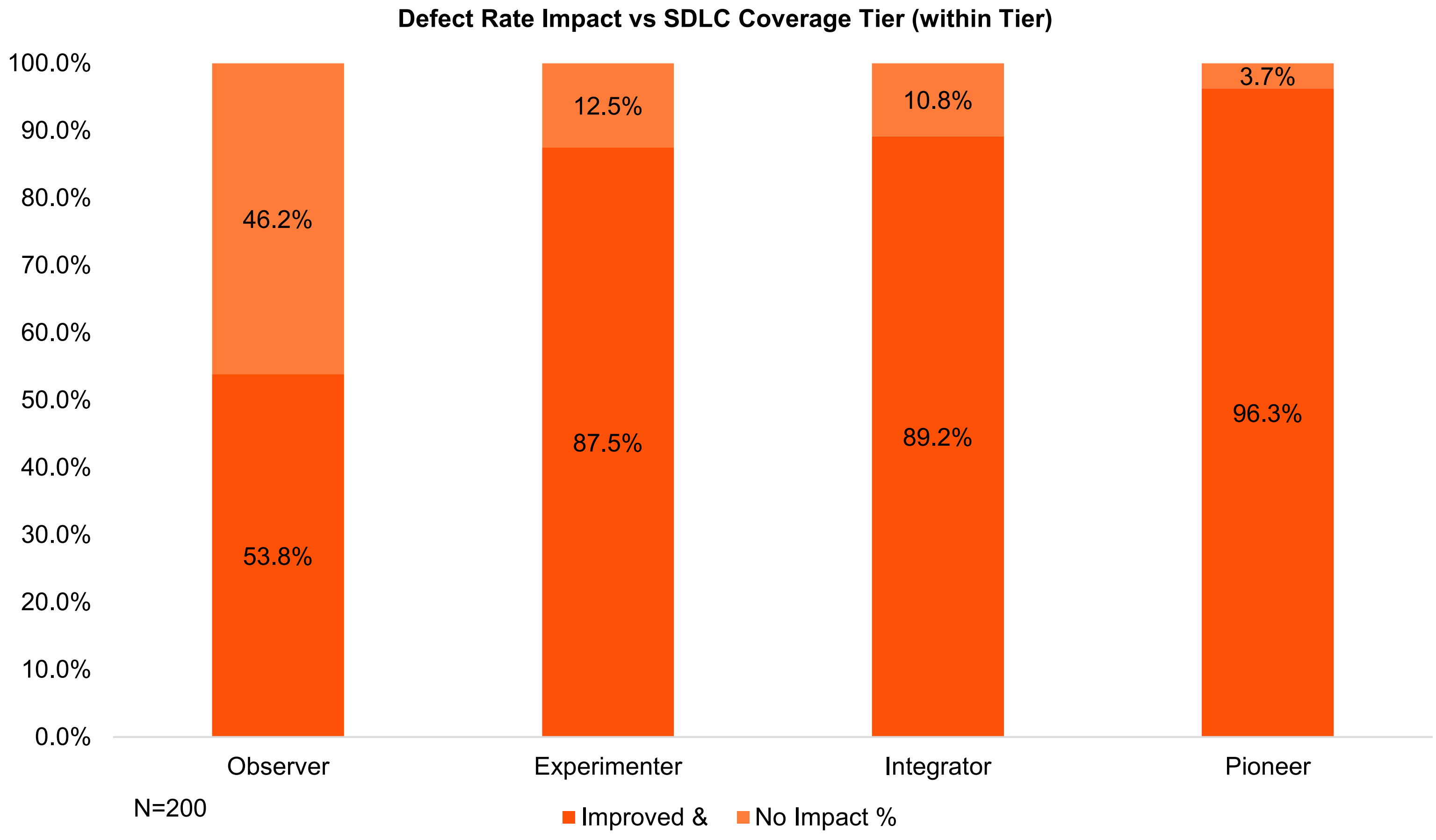
- **Positive skew –** 92 % overall see fewer bugs; GenAI rarely makes quality worse once tracked
- **"Tracking unlocks insight."** Only 55 % of the entire sample measure defects
- **"Goal-agnostic dividend."** Whether a team adopted GenAI for productivity or other reasons, once breadth and measurement are in place, quality tends to rise automatically

Notes:

- We included in this indicator only teams that **measure bugs before and after adopting GenAI within SDLC processes;** only 200 of 377 respondents measure bugs; and give usable impact data **(n=200)**

When teams actually track bugs, nine in ten report that leveraging GenAI in SDLC cuts defect rates and the effect rises to 96 % among Pioneers

Indicator 10.2 - GenAI Impact on Defect Rate / Bug Density (within-tier %)



Key findings:

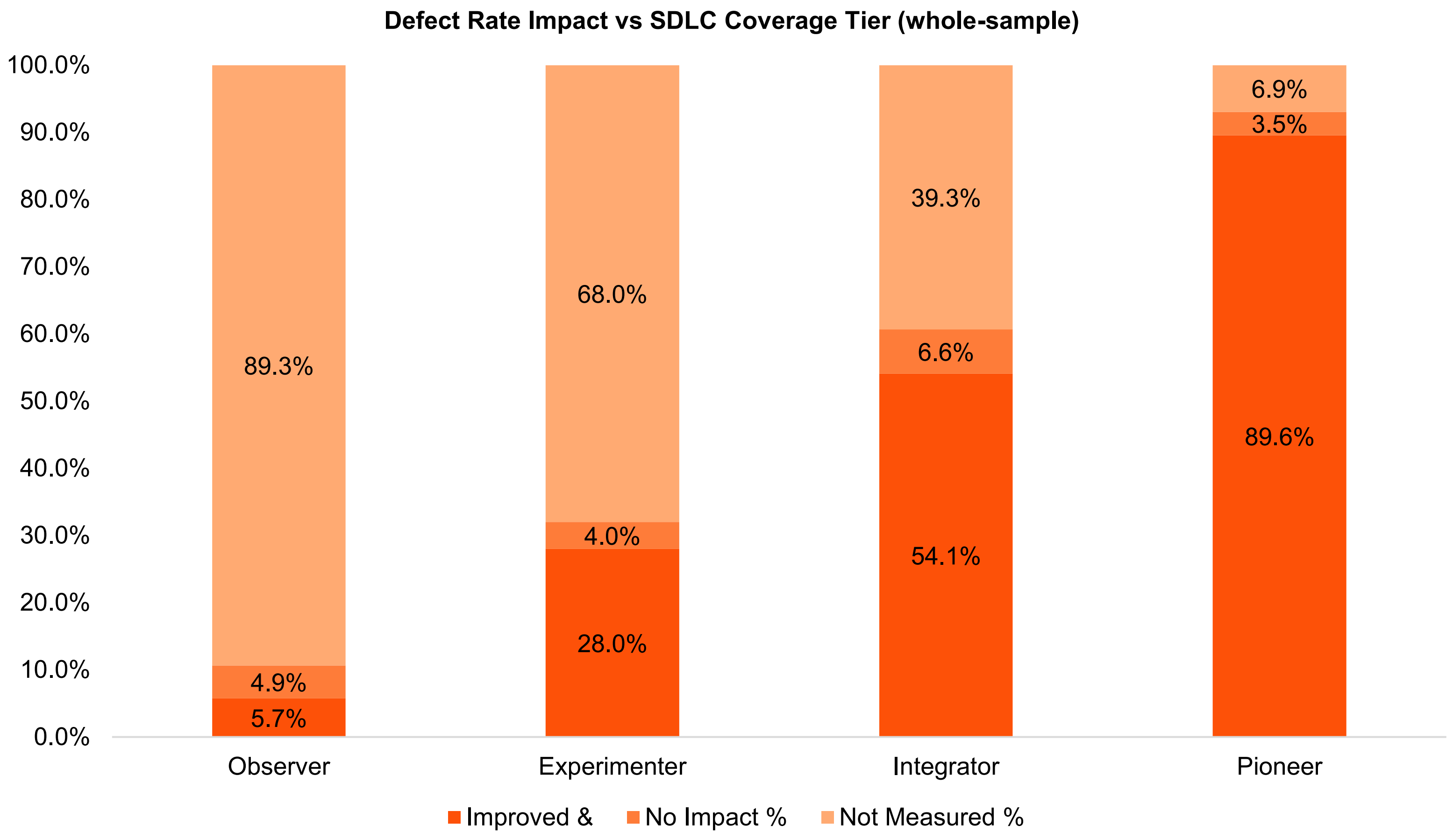
- **Maturity gradient –** Improvement climbs from 54 % (Observers) → 96 % (Pioneers), mirroring breadth correlations in earlier indicators
- **Quality lag for Observers –** Nearly half of low-breadth teams (Observers) register no quality benefit, underscoring the need for broader automation and better analytics

Notes:

- We included in this indicator only teams that **measure bugs before and after adopting GenAI within SDLC processes;** only 200 of 377 respondents measure bugs; and give usable impact data **(n=200)**

When teams track defect rates, nine in ten teams report that GenAI cuts defect rates and the effect rises to ~90 % among Pioneers practice

Indicator 10.3 - GenAI Impact on Defect Rate / Bug Density (whole-sample %)



Key findings:

- **Positive skew –** 92 % overall see fewer bugs; GenAI rarely makes quality worse once tracked
- **Maturity gradient –** Improvement climbs from 54 % (Observers) → 96 % (Pioneers), mirroring breadth correlations in earlier indicators
- **Quality lag for Observers –** Nearly half of low-breadth teams register no quality benefit, underscoring the need for broader automation and better analytics

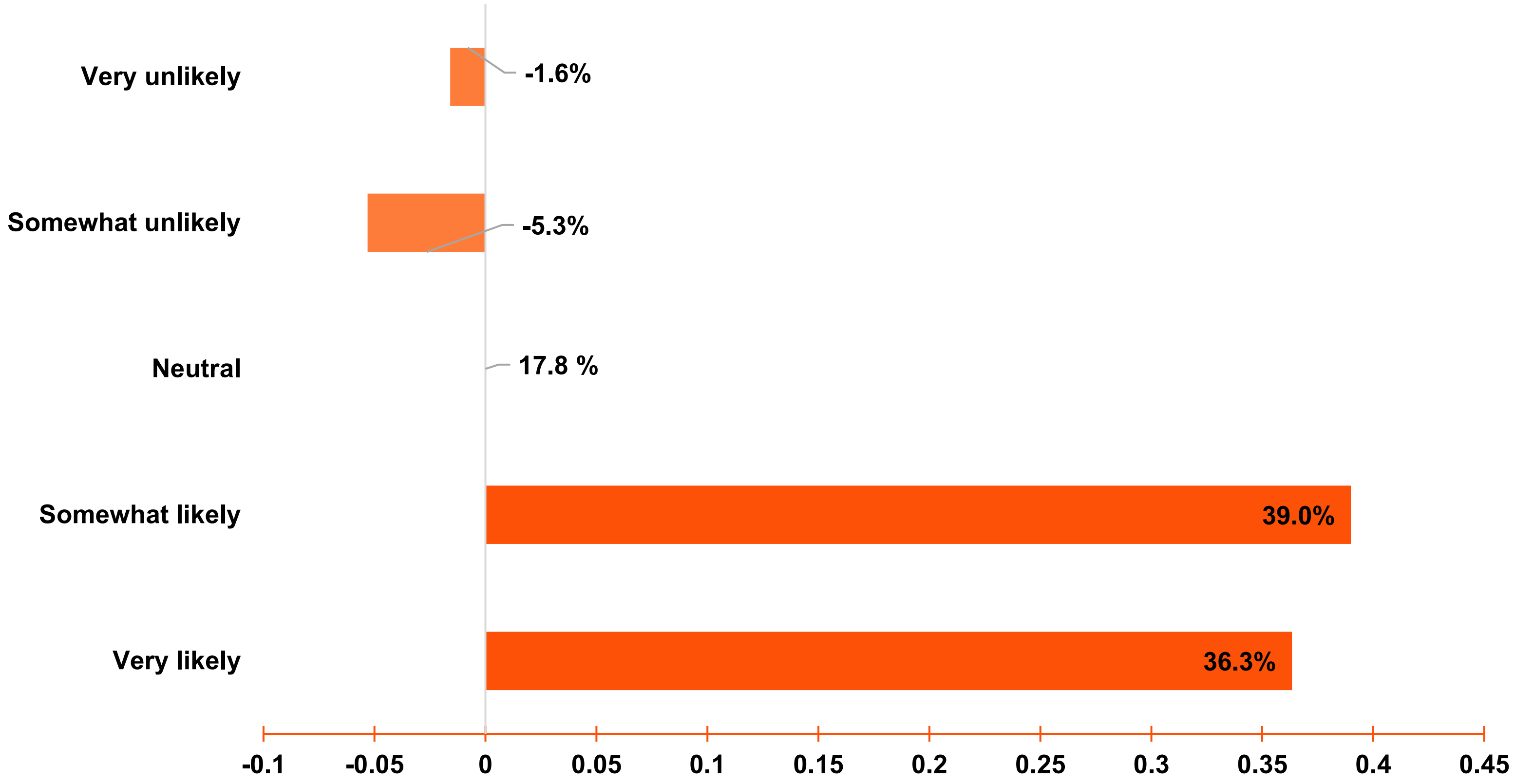
Notes:

- We included in this indicator only teams that **measure bugs before and after adopting GenAI within SDLC processes** (n=200) and **overall sample teams that don't measure** (n for not measured = 177), total is 377

Nearly 3 in 4 teams (75 %) signal they are likely to boost GenAI-tool spend incorporating it with SDLC lifecycle

Indicator 11.1 - Investment Sentiment

Spearman $\rho = +0.52$, $p < 10^{-20}$ – strong positive link between maturity and investment optimism



Key findings:

- **Clear majority ready to spend:** 3 in 4 teams are Likely investors; outright skepticism (Unlikely) sits at just 7 %
- **Maturity drives enthusiasm:** Likely share rises from 55 % (Observers) to 90 % (Pioneers); mean score jumps a full point (3.55 → 4.47)
- **Integrators already committed:** 84 % Likely, suggesting budget moves ahead of full Pioneer breadth
- **Very-Likely surge:** only 16 % of Observers tick “Very Likely” vs 58 % of Pioneers

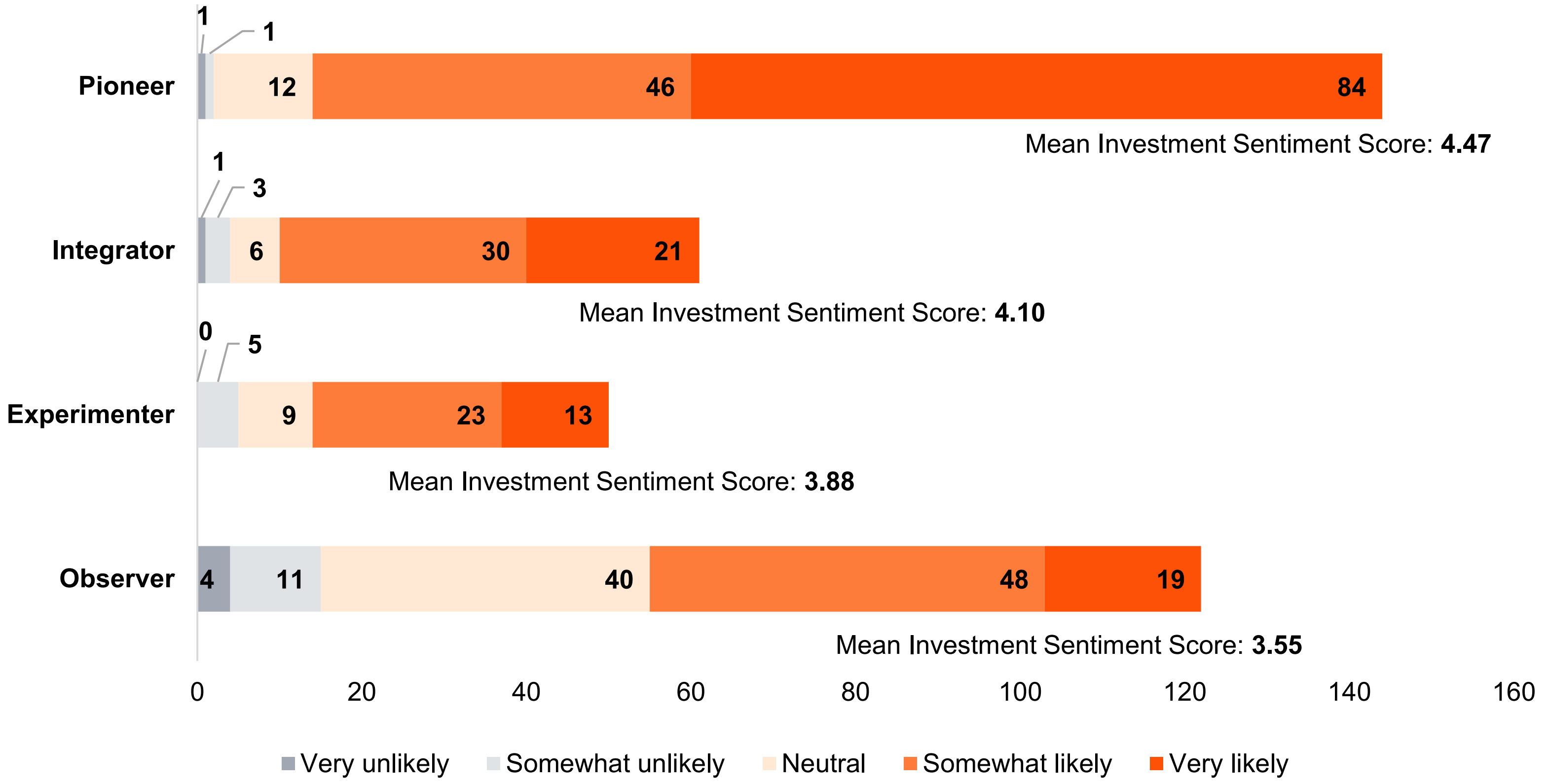
Notes:

- **Mapping used:** Very Unlikely = 1, Somewhat Unlikely = 2, Neutral = 3, Somewhat Likely = 4, Very Likely = 5

Nearly 3 in 4 teams (75 %) signal they are likely to boost GenAI-tool spend incorporating it with SDLC lifecycle

Indicator 11.2 - Investment Sentiment by coverage tier

Spearman ρ = +0.52, $p < 10^{-20}$ – strong positive link between maturity and investment optimism



Key findings:

- **Clear majority ready to spend:** 3 in 4 teams are Likely investors; outright skepticism (Unlikely) sits at just 7 %
- **Maturity drives enthusiasm:** Likely share rises from **55 %** (Observers) to **90 %** (Pioneers); mean score jumps a full point (3.55 → 4.47)
- **Integrators already committed:** 84 % Likely, suggesting budget moves ahead of full Pioneer breadth
- **Very-Likely surge:** only 16 % of Observers tick "Very Likely" vs 58 % of Pioneers

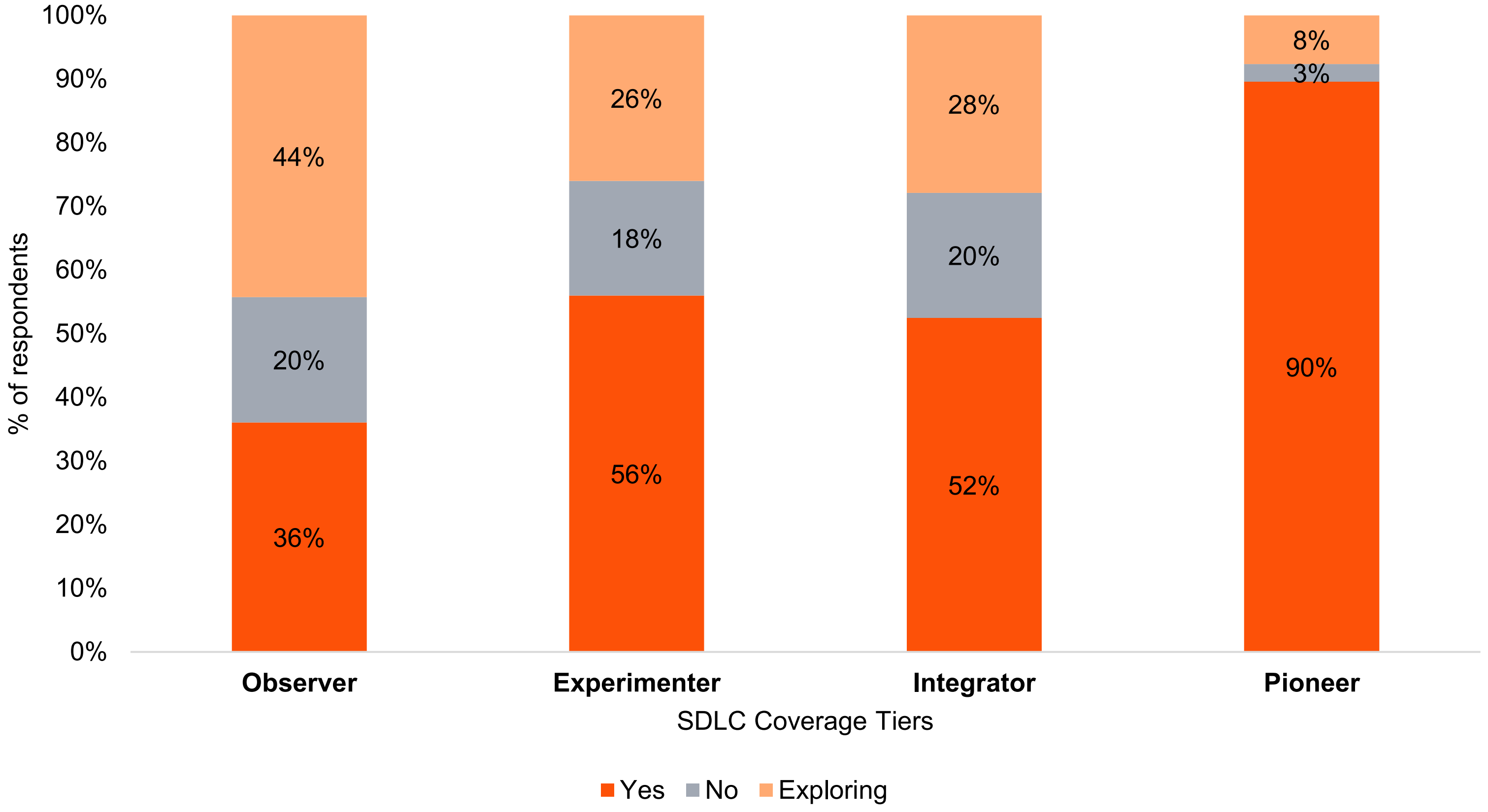
Notes:

- **Mapping used:** Very Unlikely = 1, Somewhat Unlikely = 2, Neutral = 3, Somewhat Likely = 4, Very Likely = 5
- **Mean Investment Sentiment Score** is calculated per tier n-sample, observer=122, pioneer=144, experimenter=50, integrator=61

Nearly two-thirds of organizations are already planning or exploring autonomous GenAI agents, into what is so called Agentic AI apps

Indicator 12 - Interest in Developing Agentic-AI Applications

Developing or using agentic AI apps by SDLC coverage tier



Key findings:

- **Majority momentum:** 62 % overall express interest; agentic AI is no longer niche
- **Maturity multiplier:** Pioneers are **2.5**× more likely than Observers to pursue agentic applications (90% vs 36%)
- **Inflection at mid-tiers:** Interest crosses the 50 % line at the Experimenter ➔ Integrator transition
- **Residual skepticism among early adopters:** two-thirds of Observers still hold back, pointing to opportunity for education and pilot proofs
- χ^2 (3 df) = **83.6**, $p \approx 4 \times 10^{-17}$ → maturity and agentic interest are very strongly associated.

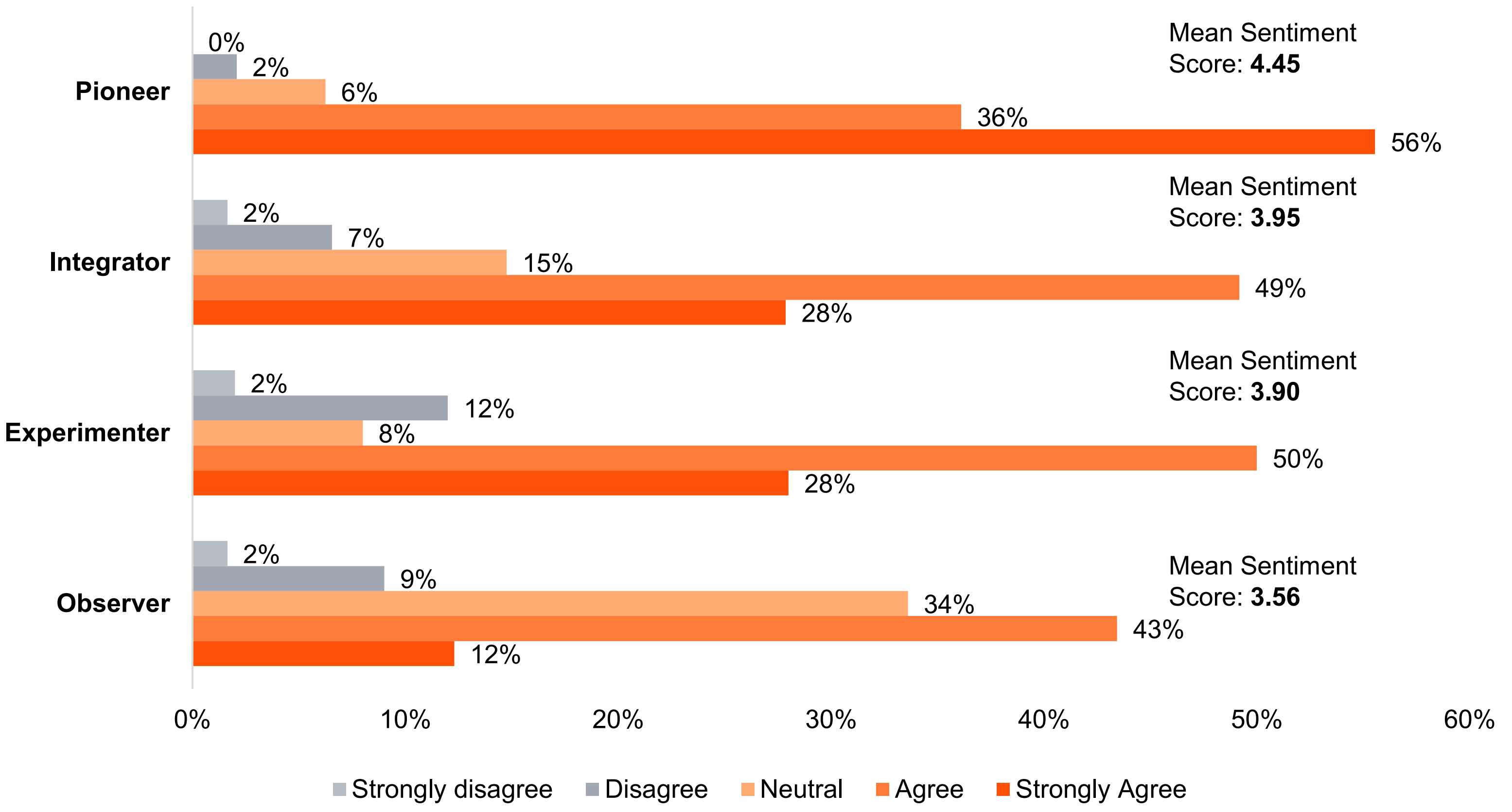
Notes:

- Chi-square shows the association between maturity tier and interest is highly significant

~75% of respondents believe agentic workflows will redefine the SDLC, teams with higher breadth coverage show similar positive sentiment

Indicator 13 - Confidence in an Agentic SDLC Future

GenAI-driven autonomous development will eventually replace traditional SDLC in the future



Key findings:

- **Widespread optimism:** Three-quarters of all participants already believe agentic workflows will reshape the SDLC
- **Maturity multiplier:** Positive share climbs from **56 %** in Observers to **92 %** in Pioneers, and the mean sentiment jumps nearly a full point
- **Early tipping:** Even Experimenters register 78 % positivity, showing conviction grows soon after initial breadths
- **Residual neutrality:** ~17 % remain on the fence — an opportunity for targeted showcases and ROI proof-points

Notes:

- Numeric mapping: Strongly Disagree = 1 … Strongly Agree = 5
- Spearman $\rho = +0.43$, $p \approx 2 \times 10^{-18}$ — confidence rises strongly with maturity

Seven emerging beliefs about GenAI's impact on SDLC, where GenAI is seen a power horse for development with concerns in following standards & creativity

Indicator 14.1 - General sentiment on GenAI adoption within SDLC

Where the SDLC thinks GenAI is heading?

	Observer	Experimenter	Integrator	Pioneer
GenAI will automate routine coding tasks	61%	80%	72%	81%
The role of software developers will shift towards more strategic activities	60%	70%	70%	81%
GenAI will reduce the need for manual testing	51%	66%	74%	77%
GenAI will cause a lot of unexplained issues	41%	40%	33%	40%
GenAI will reduce the creativity in software delivery	33%	36%	34%	51%
GenAI will lead to loosely coupled standards	34%	50%	43%	63%
GenAI will promote less tools or tech stack components	35%	54%	69%	69%

Key findings:

- **Uniform optimism; automation & role-shift are near-consensus:** Every tier tops 60 % agreement on "automate routine coding tasks" and "accelerate software delivery." Pioneers climb into the high-80s
- **Testing quality optimism emerges mid-journey:** Agreement that "GenAI will reduce the need for manual testing" rises from **51 % (Observer)** → 66 % (Experimenter) → **74 % (Integrator)** → 77 % (Pioneer)
- **Unexplained-issues fear is tier-agnostic:** "Cause a lot of unexplained issues" holds roughly **41 % agree in Observers**, drifting down to **32 % in Integrators** and **40 % in Pioneers** – concern persists at every level, it's a shared worry.
- **Creativity fears increases with breadth:** Observers split (33 % agree / 35 % disagree). Pioneers flip to **51 % disagree vs 34 % agree**—hands-on adopters see GenAI as creativity enhancer, not limiter

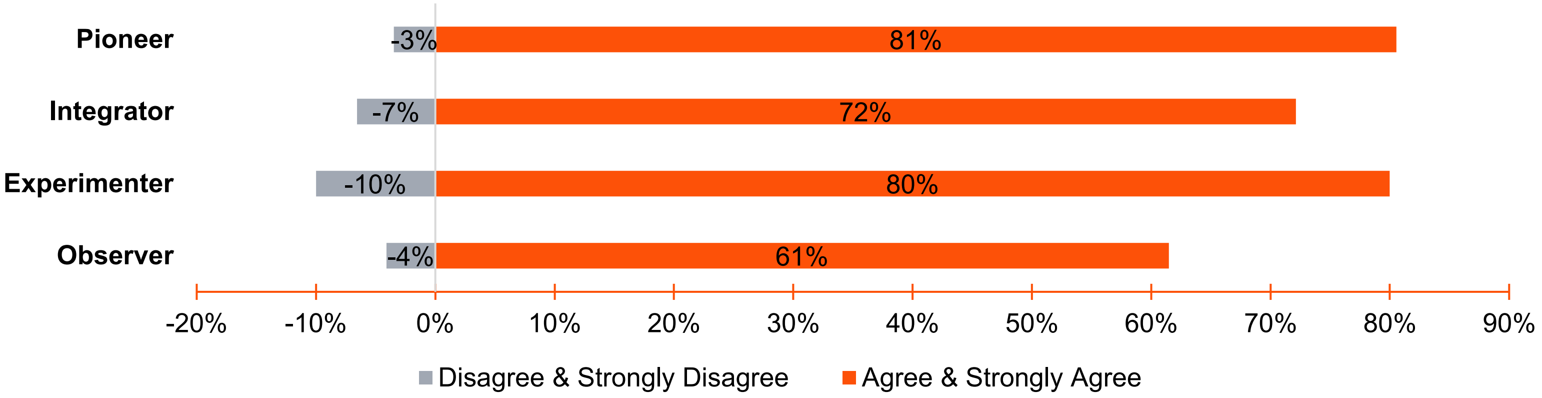
Notes:

- **Heat-map**: each cell shows the **share of that tier's respondents who selected "Agree" or "Strongly Agree."** Darker orange ⇒ stronger consensus

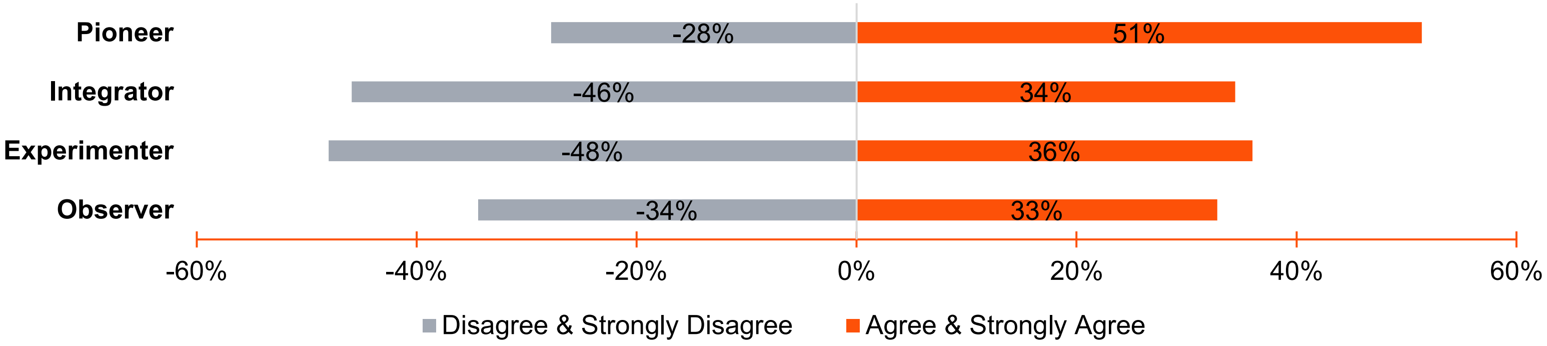
“GenAI will automate routine coding tasks” (highest agreement)
“GenAI will reduce the creativity in software development” (most polarized)

Indicator 14.2 - General sentiment on GenAI adoption within SDLC – Spotlights

GenAI will automate routine coding tasks



GenAI will reduce the creativity in software delivery



Key findings:

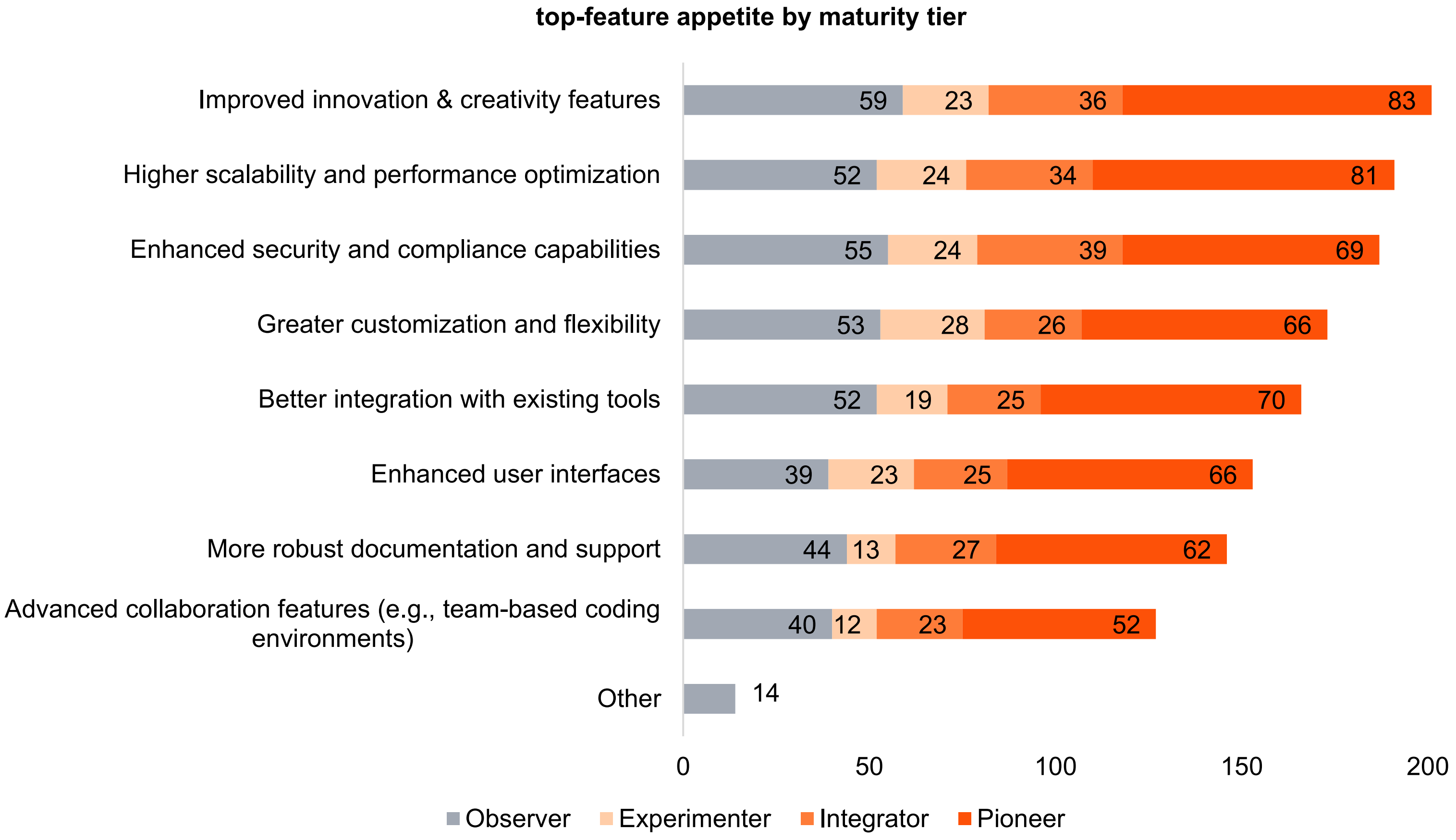
- **“GenAI will automate routine coding tasks.”** (Most positive): Agreement climbs **62 %** → **81 %** across tiers, while disagreement never rises above 10 %. Shows near-universal belief in productivity upside.
- **“GenAI will reduce the creativity in software development.”** (Most polarized): Observers split (33 % agree / 34 % disagree). Pioneers flip to **28 % disagree vs 51 % agree**—hands-on adopters see GenAI as creativity enhancer, not limiter.

Notes:

- **Diverging bars**: orange (right) = Agree, salmon (left) = Disagree, center = 0 %. Gap labels highlight how sentiment shifts with maturity

Next-gen wish-list moves beyond code gen: mature teams want built-in security and compliance guardrails, while early adopters still focus on basic productivity boosts

Indicator 15 - Wishlist for Future GenAI SDLC Features



Key findings:

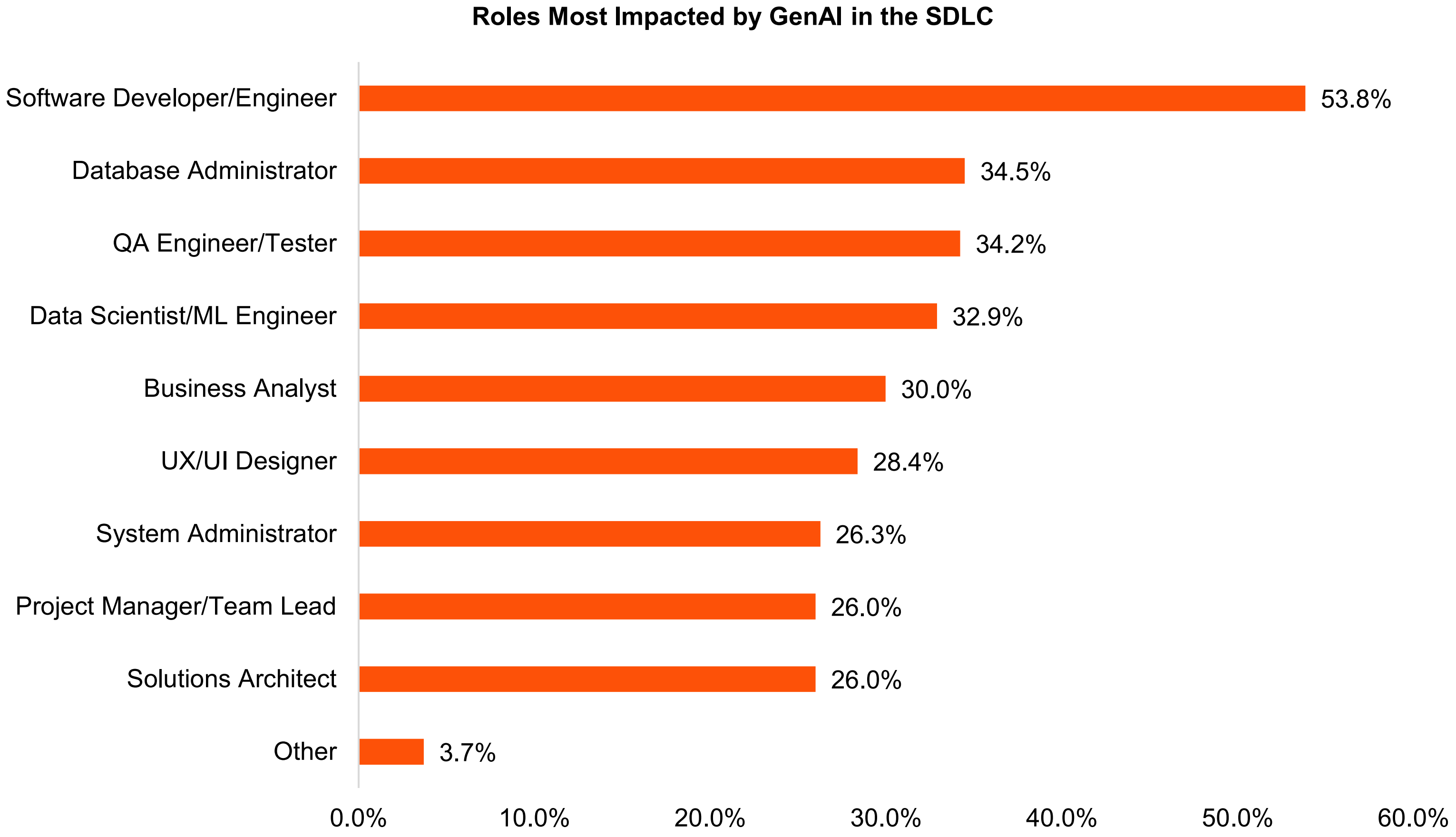
- **Creativity enhancers top the list** (53 %) suggesting teams want GenAI beyond rote code-gen
- **Performance & compliance close second/third,** highlighting scale + trust as immediate gaps
- **Integrator spike on security (64 %)** – once 4-5 stages are automated, governance pain becomes acute
- **Experimenters crave flexibility (56 % customization, 46 % UI),** reflecting early tinkering needs
- **Collaboration features lag** (<34 % overall) — multi-dev agent orchestration is still nascent

Notes:

- n-=377, representing the whole sample

Software developers / engineers will be the most affected role by Agentic SDLC open the door for shifting those roles to different responsibilities

Indicator 16.1 - Roles Most Impacted by Agentic SDLC



Key findings:

- 53.8% of 377 teams participated in the survey selects software developers as the most impacted SDLC role due to GenAI based tools and technologies
- (~34%) of respondents' teams selected DBA and QA engineers are second most impacted roles highlighting the power of GenAI in following structured pre-defined steps for database creation/administration and test cases automation
- More experienced roles like project management and solutions architecture are the least impacted roles as seen by only 26% of respondent's teams agree it will be impacted by Agentic SDLC

Notes:

- Sample size n=377; number of teams is 377

Impact cascades outward: developers feel GenAI first, but by Pioneer maturity nearly every SDLC role—from database admin to UX designer—faces substantial change

Indicator 16.2 - Roles Most Impacted by Agentic SDLC – By SDLC Tier Cut

	Observer	Experimenter	Integrator	Pioneer
Software Developer/Engineer	52.5%	44.0%	65.6%	53.5%
Database Administrator	31.1%	28.0%	31.1%	41.0%
QA Engineer/Tester	32.0%	30.0%	32.8%	38.2%
Business Analyst	26.2%	28.0%	23.0%	36.8%
UX/UI Designer	23.0%	26.0%	29.5%	33.3%
Project Manager/Team Lead	18.9%	24.0%	24.6%	33.3%
Data Scientist/ML Engineer	36.9%	28.0%	32.8%	31.3%
System Administrator	21.3%	24.0%	32.8%	28.5%
Solutions Architect	23.8%	28.0%	26.2%	27.1%

Key findings:

- **Developers top the impact list** (>50 % overall), reflecting both code-gen adoption and role shifts toward review/oversight
- **Ops roles rise with maturity:** DB admins and sys admins jump ~11 pp from Observers to Pioneers, mirroring increased GenAI use in deployment & monitoring stages
- **Analyst & design roles catch up late:** Business Analysts (+11 pp) and UX Designers (+10 pp) see the biggest Pioneer surge, suggesting GenAI expands from build to requirements & UI phases only after core automation is in place
- **Project Managers & solution architects remain cautious,** but Pioneer interest climbs to 33.3%, 27.1% respectively, hinting at emerging agent-orchestration workflows

Optimism grows with experience: mature teams are 10-20 pp more likely to foresee deeper integration, better training, and heavier automation from GenAI

Indicator 17 - Future Outlook & Sentiment Toward Agentic SDLC

	Observer	Experimenter	Integrator	Pioneer
Improved integration of GenAI into development workflows	40%	50%	62%	51%
Enhanced training opportunities for GenAI skills	39%	46%	43%	56%
Higher reliance on GenAI for automation and decision-making	38%	42%	56%	47%
Greater availability of open-source GenAI solutions	37%	54%	41%	46%
Increased adoption of GenAI tools	34%	34%	48%	50%
Emergence of ethical and regulatory frameworks for GenAI	31%	36%	44%	40%
Development of industry-specific GenAI solutions	29%	26%	43%	39%
Other	12%	0%	2%	1%

Key findings:

- Workflow integration tops the wish-list overall (49 %) and peaks at 62 % among Integrators
- Training optimism jumps 17 pp from Observers to Pioneers (39 → 56 %)
- Open-source momentum is strongest in Experimenters (54 %), hinting at cost-effective experimentation before full scale
- The smallest deltas are around ethical / regulatory frameworks (±9 pp), indicating uniform uncertainty.

Notes:

- Tier sizes: Observer = 122, Experimenter = 50, Integrator = 61, Pioneer = 144

3.4 Deep-Dive Correlations

Our data shows that there is a positive correlation between level of adoption with the breadth coverage of SDLC stages using GenAI

Deep-dive 1: Stage-Coverage Breadth × Adoption Level

	Coverage tiers (stage level)			
Adoption bucket	Observer	Experimenter	Integrator	Pioneer
High	10.5	9.7	16.9	62.9
Moderate	28.8	15.8	23.7	31.7
Low	74.4	14	0	11.6
Not-yet	52.1	14.1	9.9	23.9

Key findings:

- **Pioneer hot-spot:** 63 % of High adopters automate 6-7 stages, versus just 12 % of Low adopters
- **Observer gravity:** Three-quarters of Low adopters remain Observers (0-1 stages)
- **Moderate drift:** Adoption shifts visibly from Observer → Experimenter → Integrator as we move Moderate → High, forming a clear maturity diagonal.
- **Low adopters** must first break out of Observer gravity (add 1-2 stages).
- **Moderates need to leap** from Integrator to Pioneer to unlock high status

Notes:

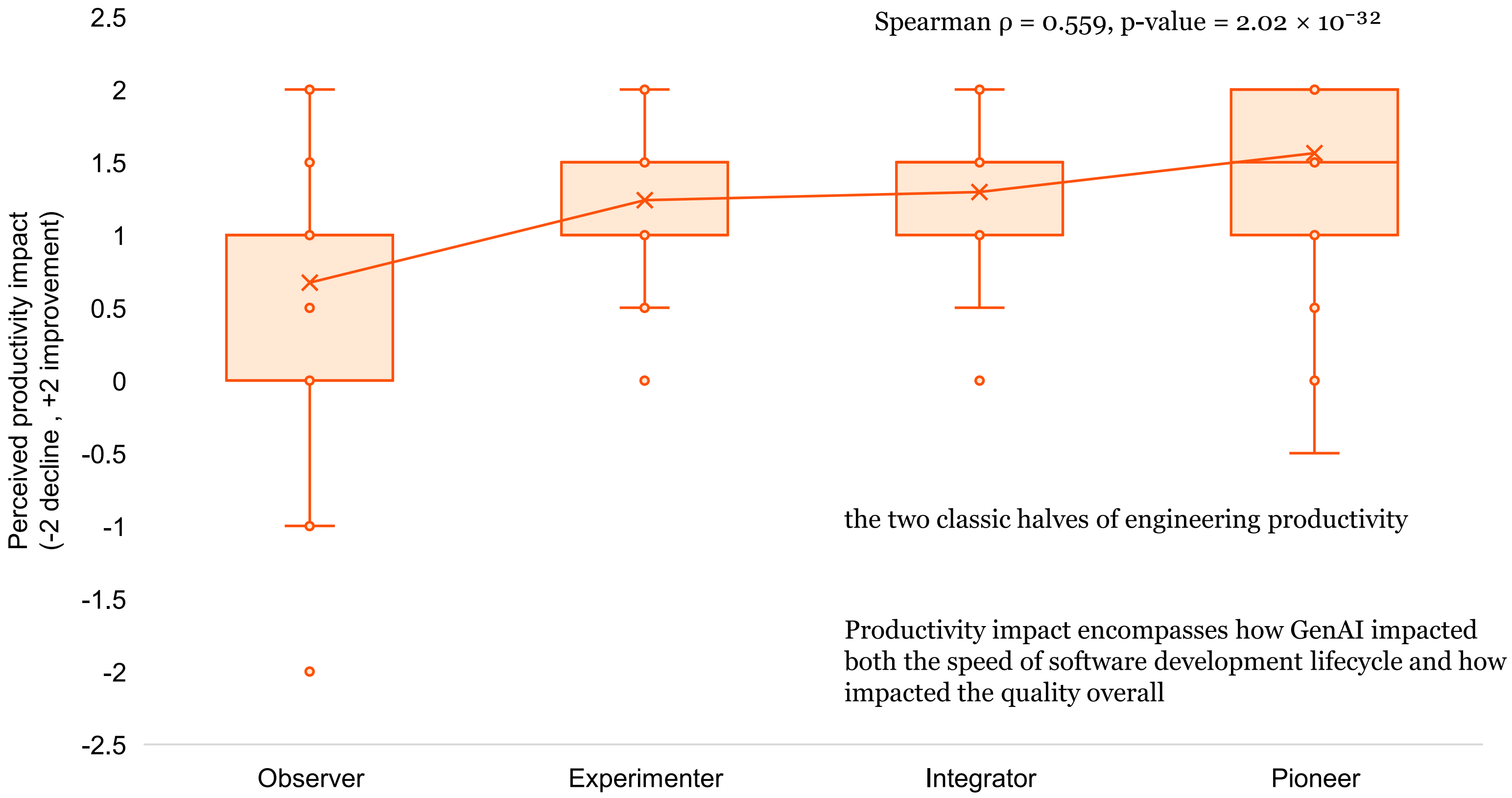
- **Tier thresholds:** Observer (0-1), Experimenter (2-3), Integrator (4-5), Pioneer (6-7).
- **Equal weight caveat:** Each stage counted equally; future work could weight critical phases.
- **Sample size sufficiency:** All adoption buckets > 40 respondents, so χ^2 assumptions met.
- **Stats:** $\chi^2 = 99.7$ (df = 9, $p < 0.000001$) → strong, non-random association

Breadth begets efficiency: each extra pair of SDLC stages automated nudges perceived productivity up by roughly a third of a point; Pioneers report the strongest lift

Deep-dive 2. productivity-impact score by SDLC Coverage tier (Observer → Pioneer)

Boxes = 25th–75th percentile, median line (inclusive)

Whiskers = 1.5 × IQR; outliers plotted as dots for transparency



Spearman ρ = 0.559, p-value = 2.02 × 10^{-32}

the two classic halves of engineering productivity

Productivity impact encompasses how GenAI impacted both the speed of software development lifecycle and how impacted the quality overall

Key findings:

- The association between Stage-Coverage tier and the direction of perceived productivity impact is highly significant; Pioneers are vastly more likely to report a positive lift, while Observers account for most neutral or negative cases
- **Steady ascent:** Median productivity climbs tier-by-tier (1.0 → 1.5), and the upper quartile reaches the maximum +2 for Integrators and Pioneers
- **Effect size: Spearman ρ = 0.56**, *p* ≈ 2 × 10^{-32} (strong, monotonic) | Pearson r = 0.55 (linear corroboration)
- **Observer drag:** A quarter of Observer teams see **no change or decline** (25th % = 0.0), underscoring the cost of minimal augmentation

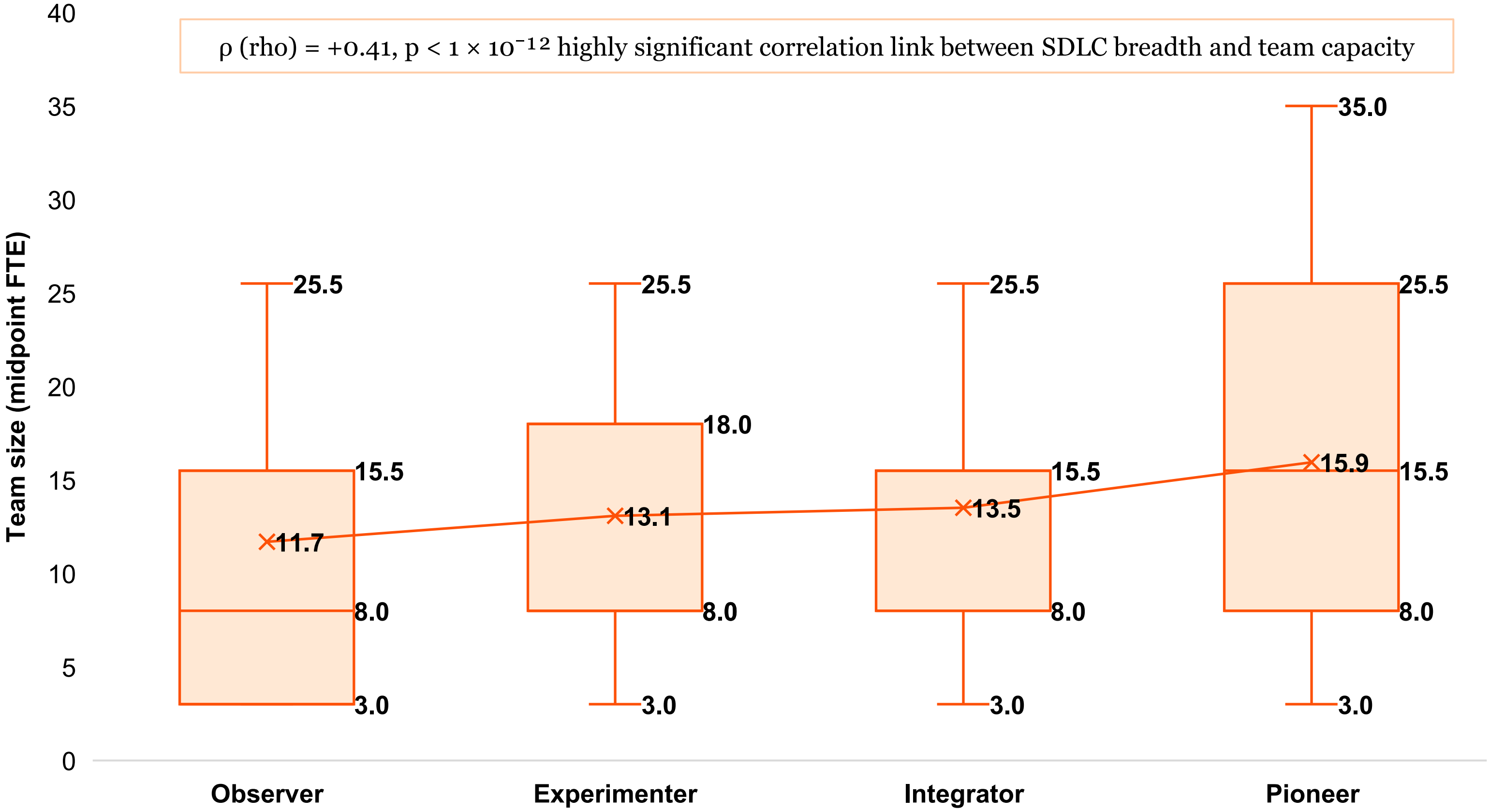
Notes:

- **Scale construction:** −2 = Significant decline, −1 = Moderate decline, 0 = No change, +1 = Moderate improvement, +2 = Significant improvement
- **Causality caution:** Productive teams may simply automate more stages *because* they can, not solely *because* GenAI caused the lift

Bigger benches, broader automation

Deep-dive 3. for team size capacity and SDLC maturity tier

Team Capacity vs Stage-Coverage Tier



ρ (rho) = +0.41, $p < 1 \times 10^{-12}$ highly significant correlation link between SDLC breadth and team capacity

Key findings:

- median team size roughly doubles from **Observer (8 FTE)** to **Pioneer (15.5 FTE)**
- **Observers cluster small:** 75 % of Observer teams sit in the 1-10 range
- **Pioneers broaden** half are ≥ 15.5 FTE reaching more than 30 FTEs in each team
- Across our region, each step-up in team capacity is matched by a richer GenAI footprint. Larger squads (median 15–16 FTE) are statistically more likely to automate six or seven SDLC stages, while micro-teams (median 8 FTE) remain concentrated in the Observer tier. The positive, highly significant Spearman coefficient (+0.41) confirms that scaling head-count and scaling GenAI adoption move hand-in-hand

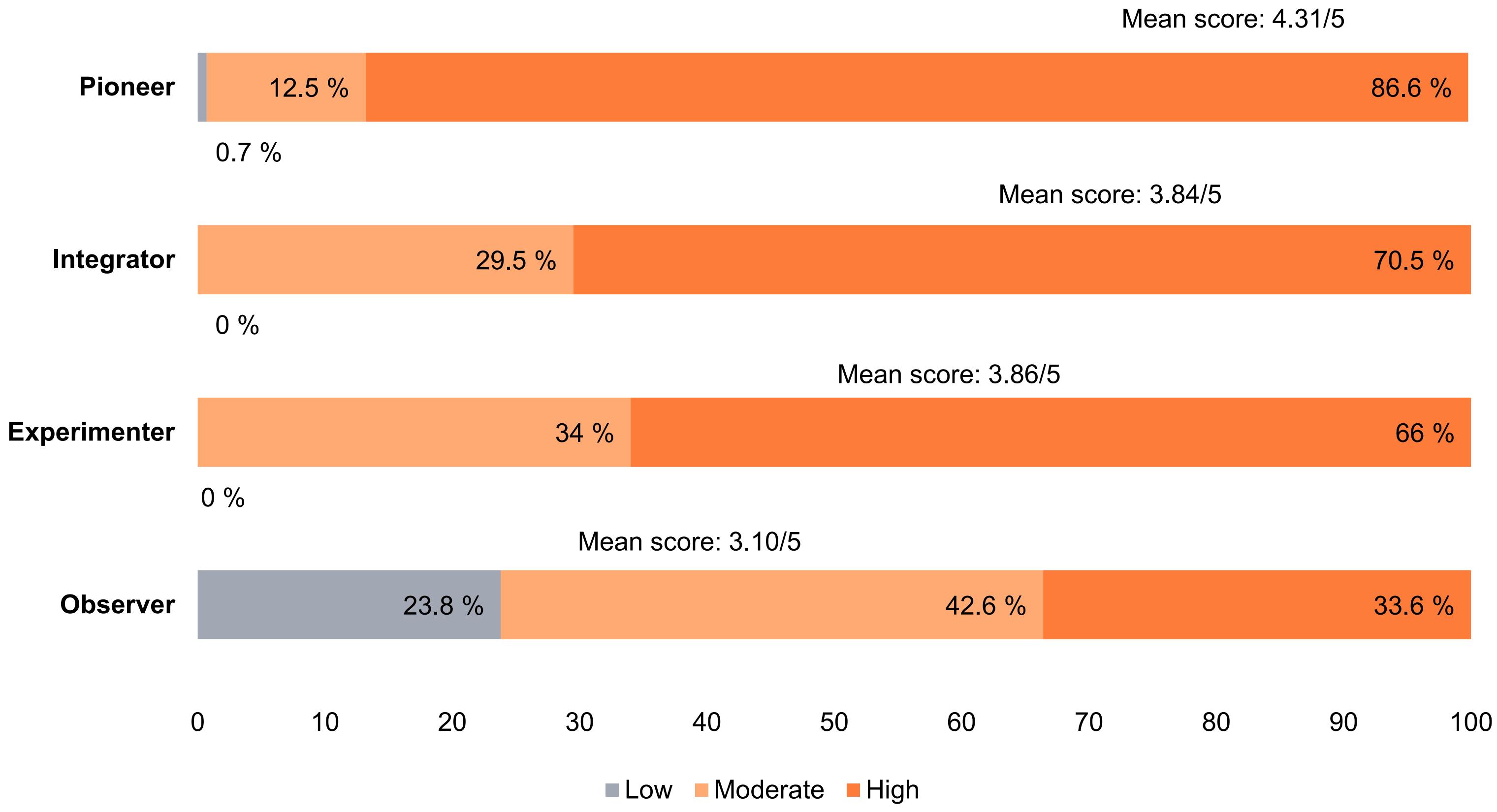
Notes:

- Team-size band → midpoint is a standard method for measuring the median # of FTE for each coverage tier
- Correlation is descriptive; causality could run both ways (bigger teams need automation; automation success drives team growth)

Skill scales with breadth: only one-third of Observers rate their GenAI know-how ‘High’, while four in five Pioneers do

Deep-dive 4. GenAI Talent Density - Skill Level Inside Teams

GenAI skills maturity level distribution among SDLC coverage tiers



Key findings:

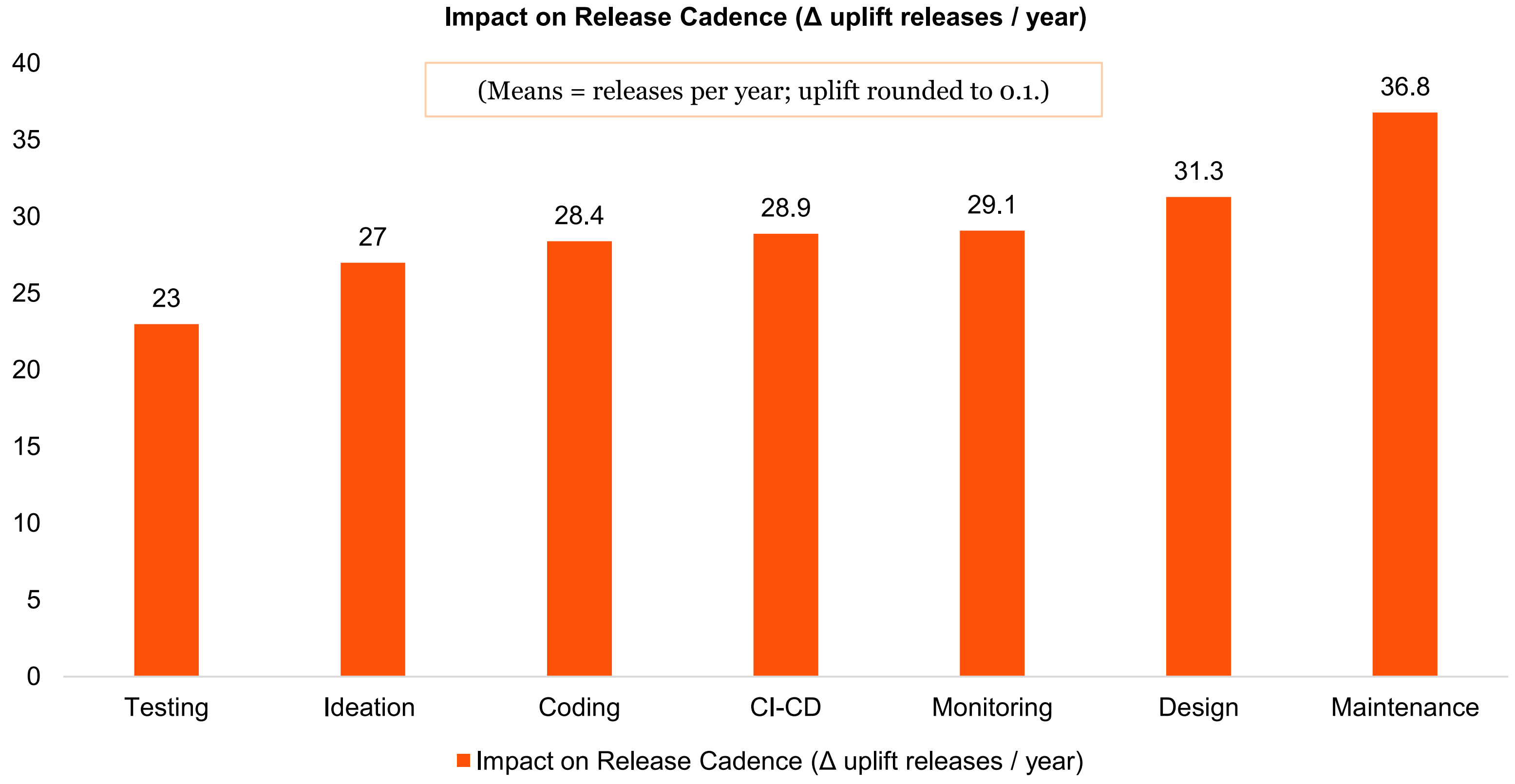
- **Skill ascends with maturity** – mean score climbs from 3.1 (low-moderate) to 4.3 (high)
- **86.8% of Pioneers** report “High” or “Very High” skills, more than double the Observer share
- **Mid-tiers already solid** – Integrators hover near 3.8 mean; capability building begins well before full Pioneer breadth

Notes:

- **Numeric mapping**: Very Low = 1 → Very High = 5
- **Mean score** per tier was computed directly on that mapping
- **High + Very High %** = count of “High” or “Very High” answers ÷ tier n

Maintenance automation is the hidden accelerator: teams that GenAI-enable post-release maintenance ship ~37 more versions per year

Deep-dive 5. Which SDLC Stages Turbo-charge (uplift) Release Cadence?



(Δ = difference in mean releases/year between teams that **do** vs **do not** automate each stage with GenAI)

Key findings:

- Among survey population the **Maintenance stage in SDLC** is the most one impacted positively on release cadence of software; post-release tasks often bottleneck minor version bumps; automating them frees up continuous delivery, in addition this suggested that already existing large code bases within surveyed organizations are being revamped, enhanced, and bug triaged leveraging GenAI based tools / IDEs etc.
- **Upstream design matters** – GenAI-assisted design reviews cut re-work, translating to ~25 extra releases per year
- **Testing lags in impact** – despite hype, automated testing alone yields the smallest release cadence gain, suggesting many teams already had mature CI test suites pre-GenAI era

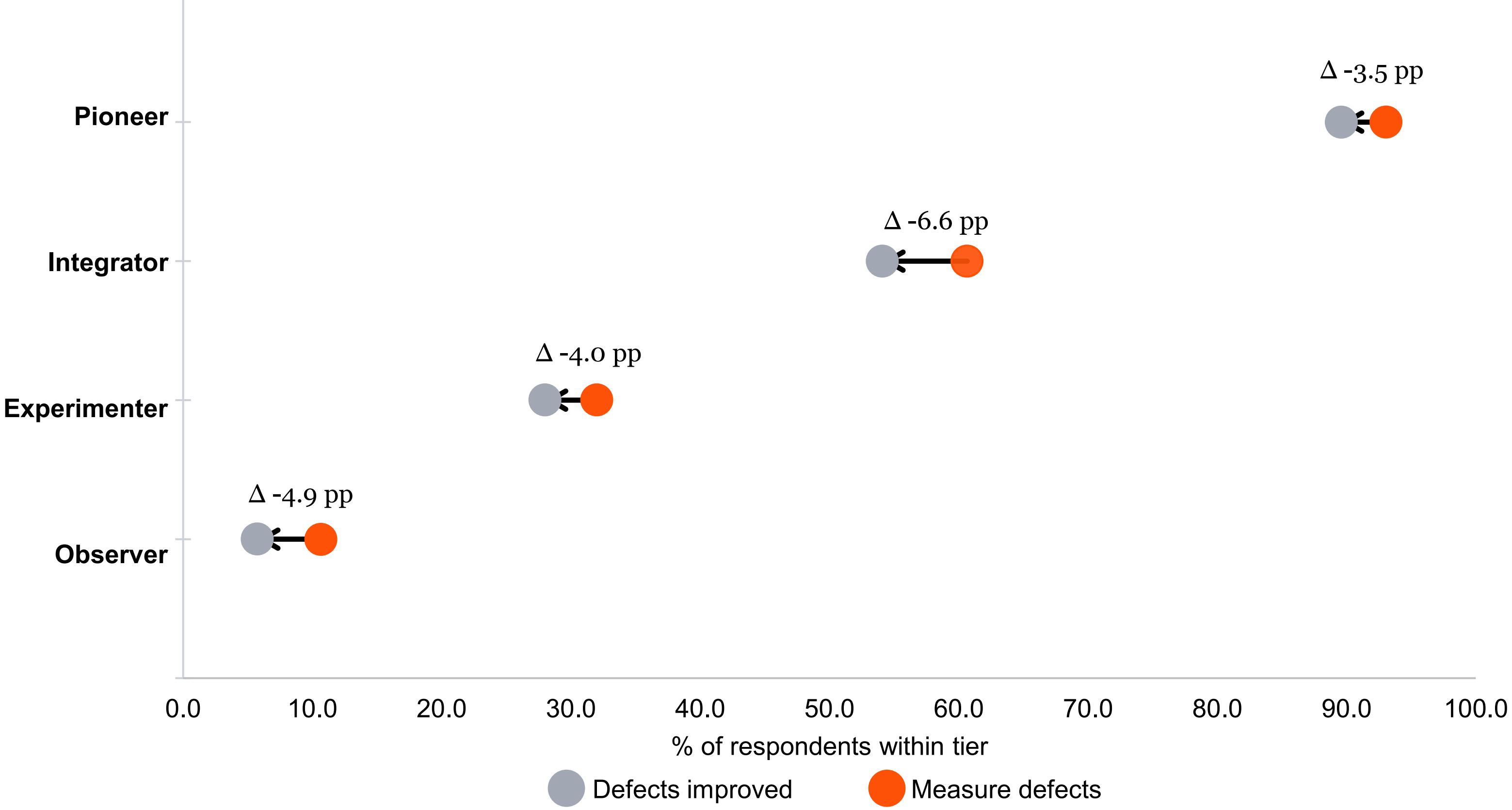
Notes:

- **Correlation, not causation.** High-velocity teams may choose to automate Maintenance precisely because they already release often
- **Enough sample on both sides.** Every stage has > 140 Yes-respondents and > 200 No/Explore, so means are stable
- **Means not medians** so that uplift reads directly as "extra releases/year."

Deep dive comparison among SDLC coverage tiers; when teams measure improvement is reported accounted for Augmented SDLC with GenAI

Deep-dive 6. GenAI Impact on Defect Rate / Bug Density | Measurement Gap Focus

Gap focus: contrast Observer to Pioneer on both "Measuring" and "Improved" N=200



Key findings:

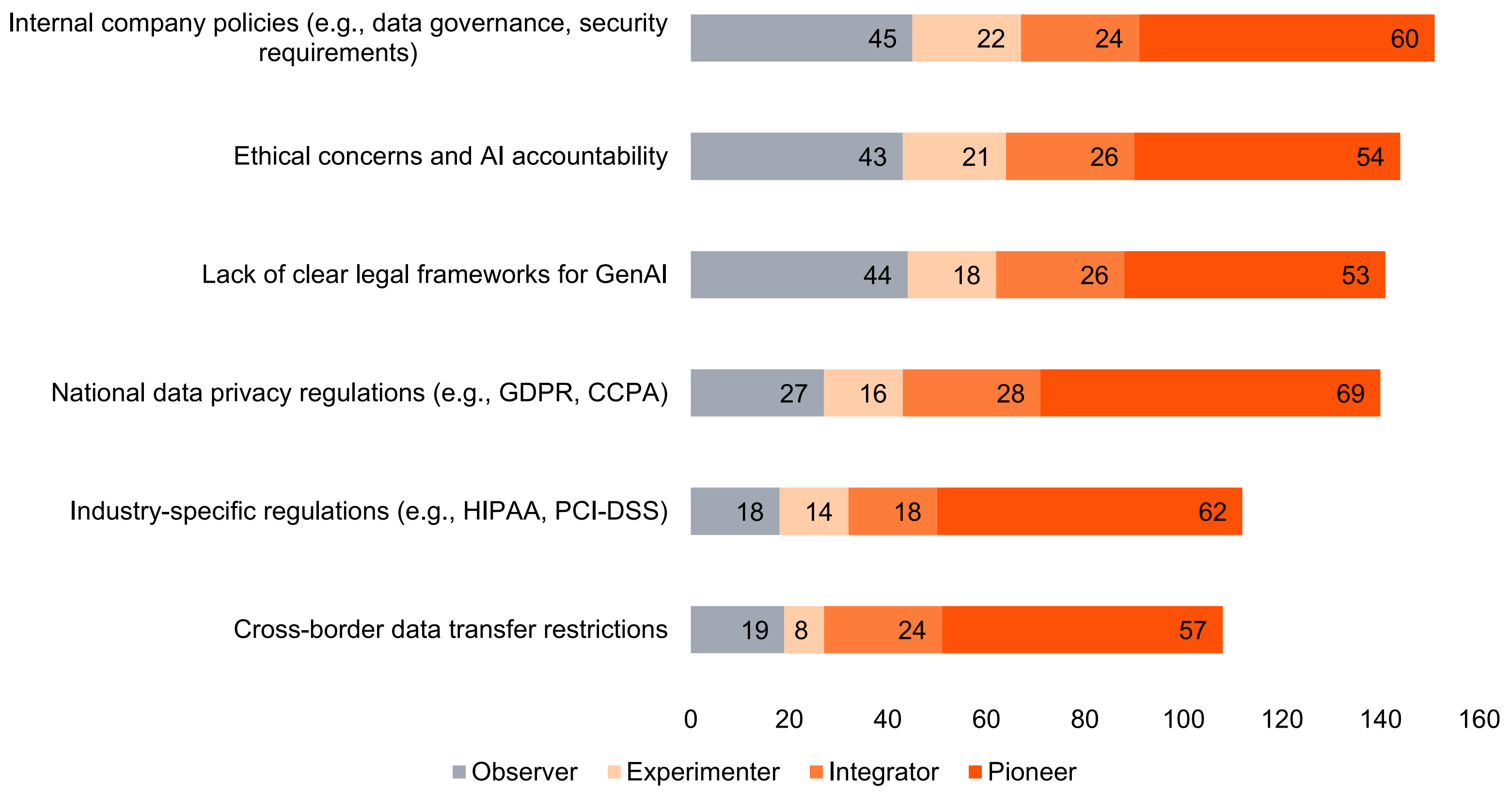
- **Pioneers institutionalize measurement** 93% track defect metrics vs 10 % of Observers
- Pioneers close 89.6 % of their tier with real quality gains; gap shrinks to –3.5 pp
- **Observers fly blind:** 9 out of 10 Observers can't even quantify quality impact; only ~6 % report improvement
- **Middle tiers still leave value on the table:** Integrators measure ~60%, but 6.6 pp of that group see no defect reduction; an opportunity for targeted QA automation

Notes:

- We included in this indicator only teams that **measure bugs before and after adopting GenAI within SDLC processes;** only 200 of 377 respondents measure bugs; and give usable impact data **(n=200)**
- Some team measures the bugs before and after, but didn't report improvement

Teams faces many challenges from regulations compliance perspective when it comes to implementing GenAI based applications

Deep-dive 7. Regulatory / Compliance Challenges



Key findings:

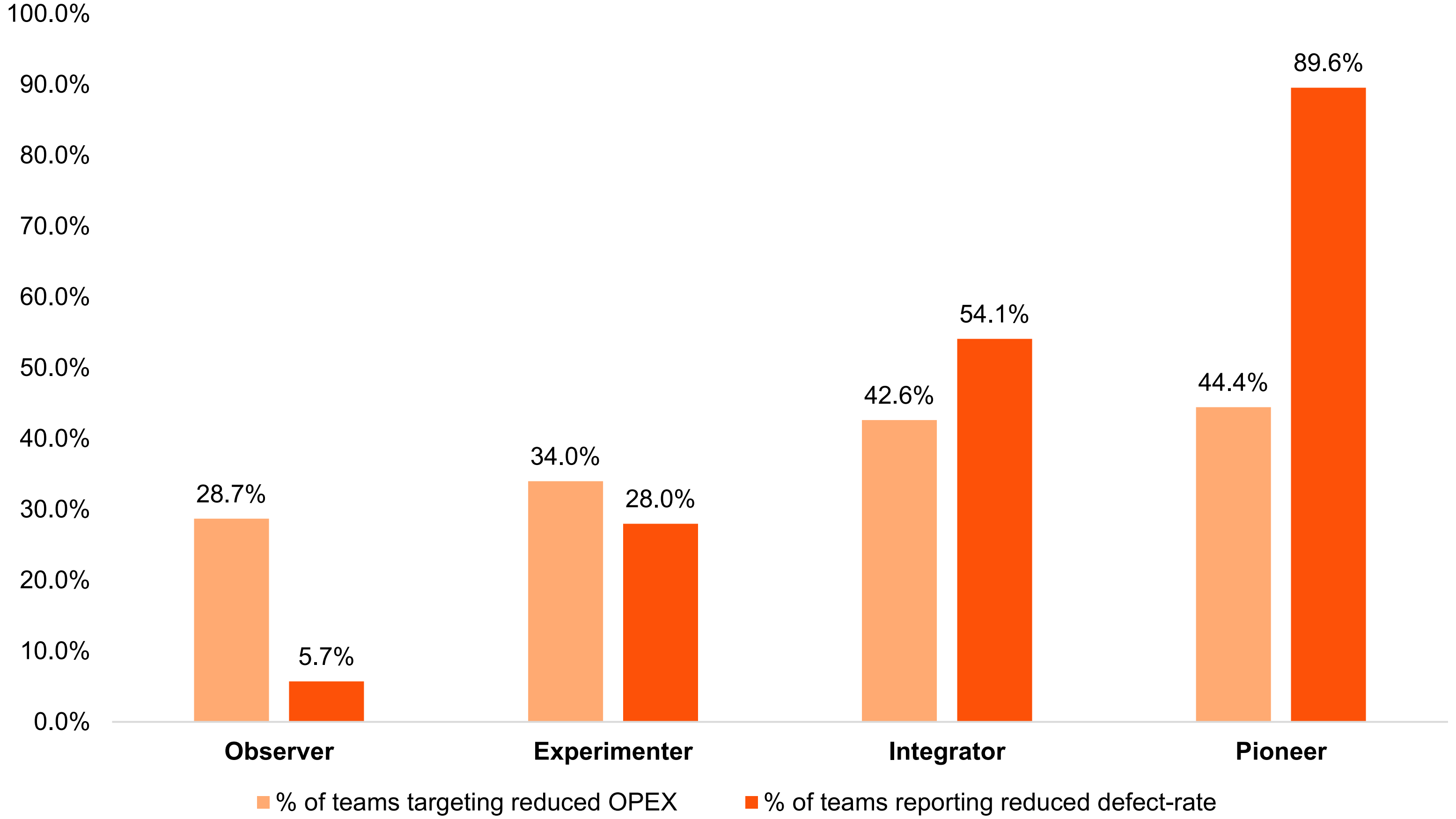
- Internal policies score the highest blocker in implementing GenAI solutions, where it seen more often when GenAI breadth across SDLC is higher (Pioneer teams see it more than observers by 15 pp)
- Pioneer teams highlights more challenges compared to lower coverage tiers like integrator, which shows higher maturity in Agentic SDLC the more challenges teams will face

Notes:

- Teams can select more than one challenge in the survey; thus, results total are greater than sample size (n=377)

Higher matured teams leverage GenAI to reduce OPEX, while we used Defect-rate reduction index to measure the realization of this effort vs goals

Deep dive 8. Cost-Savings Overlay & Defect-rate Indicators 9 & 10



Key findings:

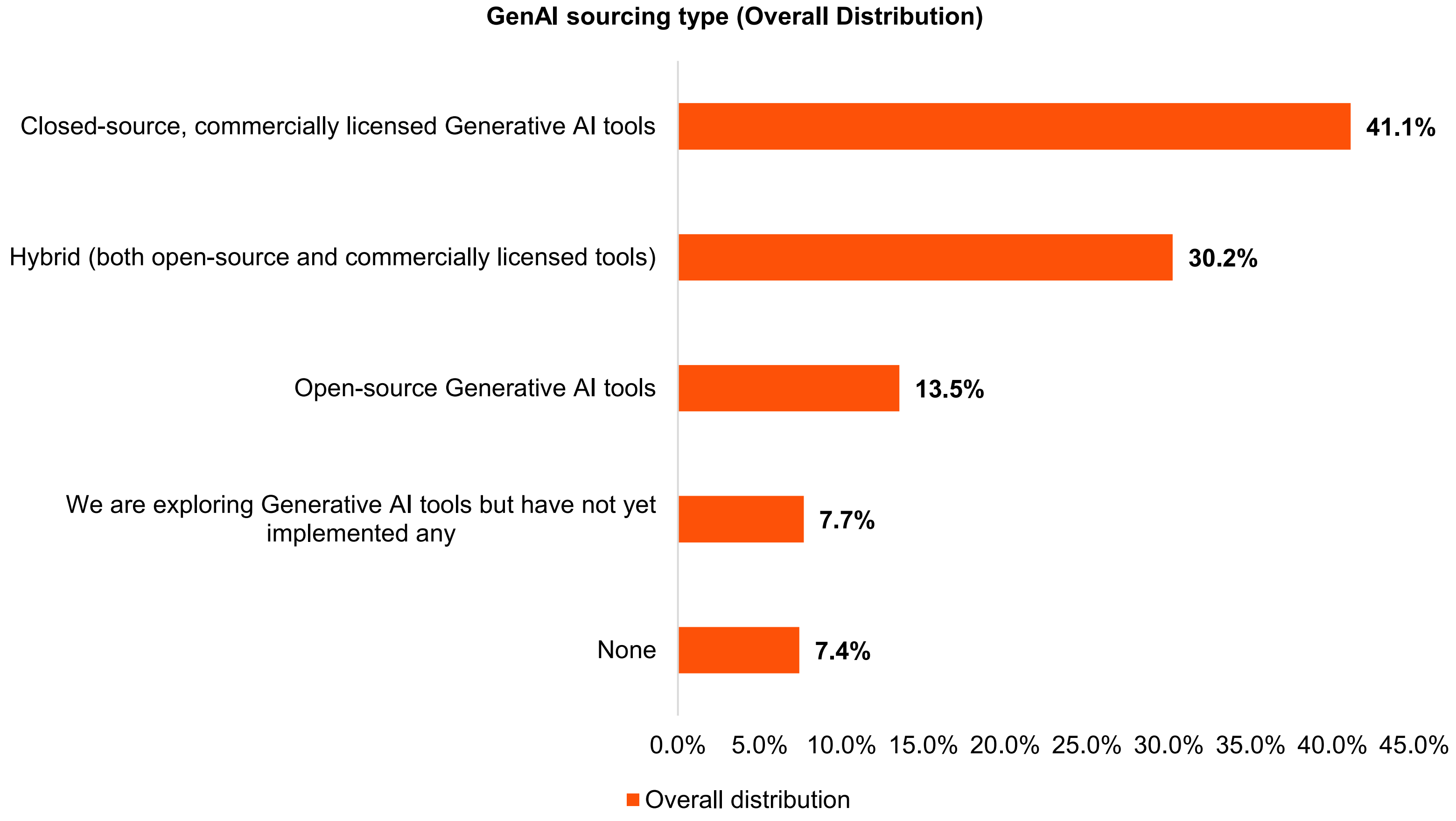
- **Cost-efficiency becomes a priority mid-journey:** share rises from 29 % (Observers) to 44 % (Pioneers).
- **Quality payoff lags cost intent:** only 6 % of Observers both chase savings and cut defects, versus 90 % of Pioneers.
- **Integrator inflection:** defect-rate reduction jumps 26 pp (28 → 54 %) while cost goal rises 9 pp, suggesting efficiency measures begin paying off once 4–5 stages are automated.
- **Money follows quality:** correlation shows teams that succeed in cutting defects are the same ones targeting OPEX—GenAI delivers twin benefits when fully embedded

Notes:

- This is considered one indicative relation from OPEX perspective which is the reported reduced defect-rate
- Spearman ρ = **+0.64**, $p < 10^{-28}$ — strong positive association

Closed-source models are the top used type of GenAI within Augmented SDLC, while opensource adoption considered the lowest

Deep-dive 9. GenAI Technology Types in Use



Key findings:

- A small but notable **open-source-only** cohort persists (14 % overall), indicating cost-sensitive or sovereignty-driven teams
- Closed source models are at top of consumption reflecting the direct access to accelerators through APIs for AI models inferences, suggesting a faster access with no upfront setup required compared with open-source models

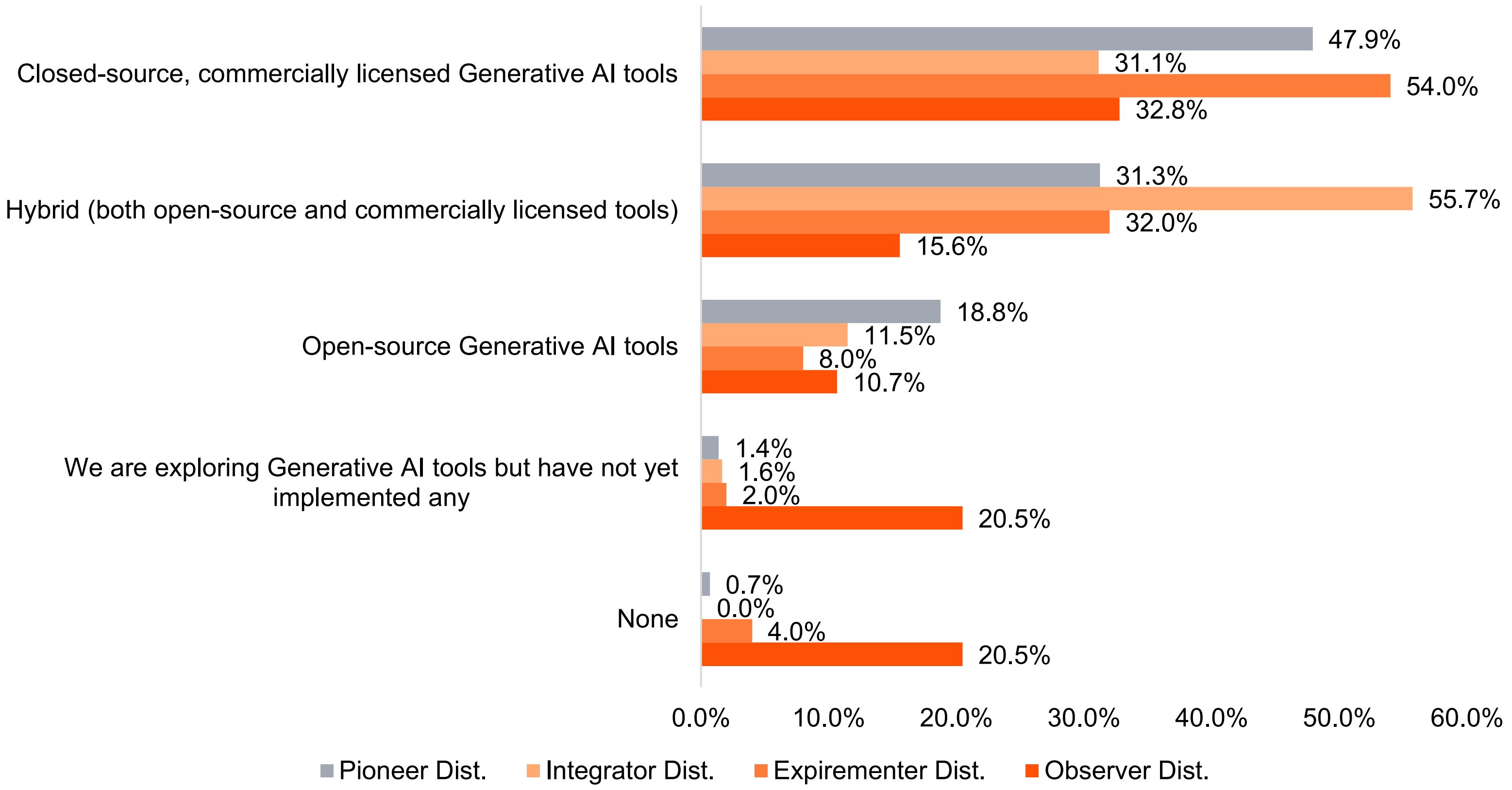
Notes:

- Closed-source type examples: OpenAI, Google, Microsoft, and others
- Open-source type examples: Meta, models deployed on Hugging face, and others

From curiosity to commitment: Pioneers double down on commercial or blended tool-stacks, while Observers remain stuck in the ‘exploring’ phase

Deep-dive 10. GenAI Technology Types in Use

GenAI sourcing type by SDLC coverage tier



Key findings:

- **Commercial LLM stacks dominate overall (41 %),** rising to 48 % in Pioneers
- **Hybrid usage** (both open & closed) jumps from 16 % to 31 %, showing mature teams diversify
- **Exploration gap:** one-fifth of Observers are still experimenting vs only 1 % of Pioneers—illustrating the execution gulf

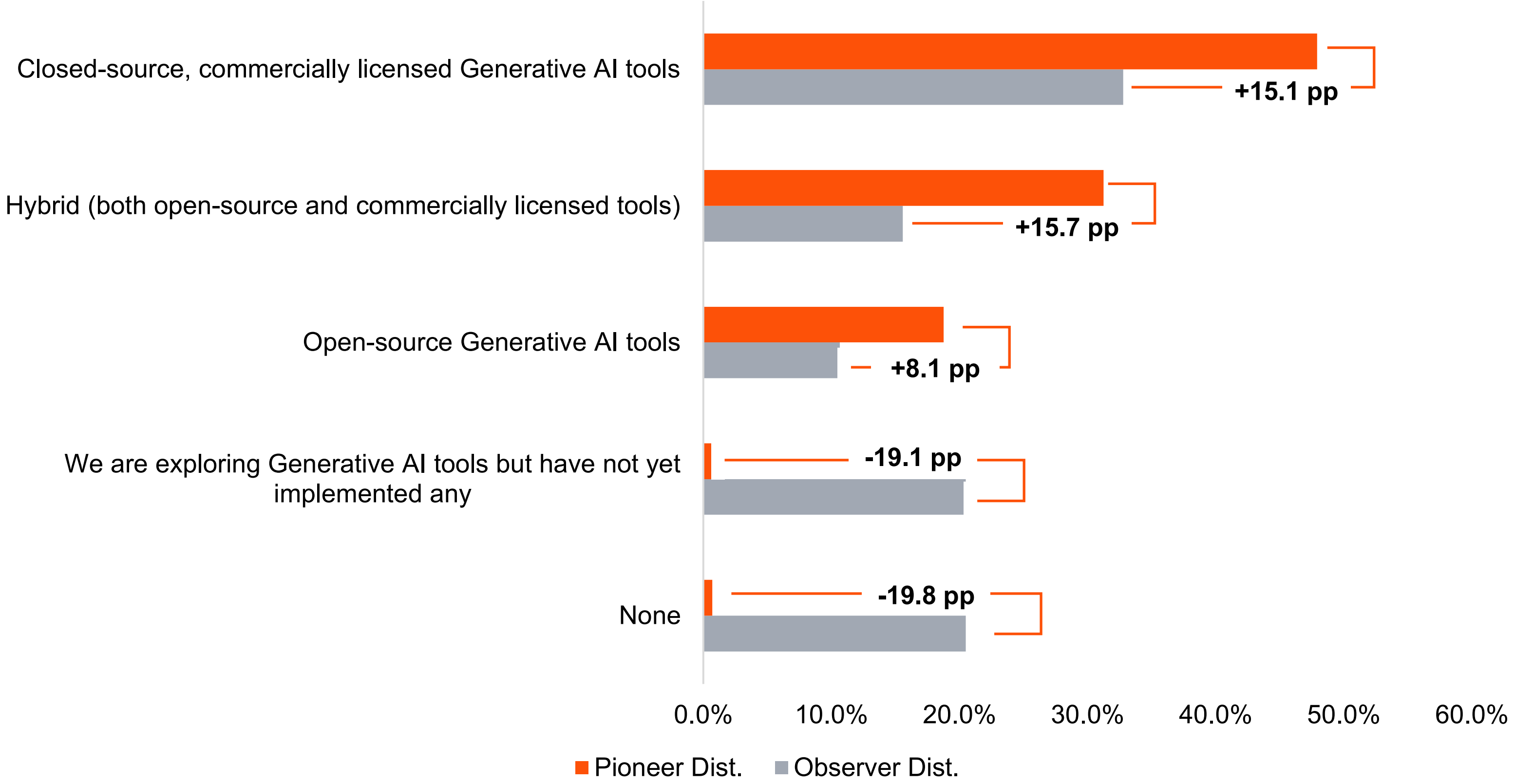
Notes:

- Closed-source type examples: OpenAI, Google, Microsoft, and others
- Open-source type examples: Meta, models deployed on Hugging face, and others

From curiosity to commitment: Pioneers double down on commercial or blended tool-stacks, while Observers remain stuck in the 'exploring' phase

Deep-dive 11. GenAI Technology Types in Use

GenAI sourcing type by SDLC coverage tier (Observer vs Pioneer)



Key findings:

Commercial LLM stacks dominate overall (41 %), rising to 48 % in Pioneers

Hybrid usage (both open & closed) jumps from 16 % to 31 %, showing mature teams diversify

Exploration gap: one-fifth of Observers are still experimenting vs only 1 % of Pioneers—illustrating the execution gulf

Notes:

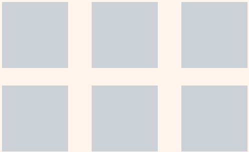
- Closed-source type examples: OpenAI, Google, Microsoft, and others
- Open-source type examples: Meta, models deployed on Hugging face, and others

3.5 GenAI Tools Landscape in Software Development

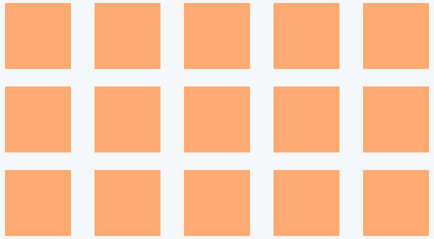
46.9% of teams are using GenAI based IDE tools while 79.1% of them are using more than one tool to augment their SDLC stages end to end

Indicator 18.1 - GenAI based IDE distribution among respondents teams

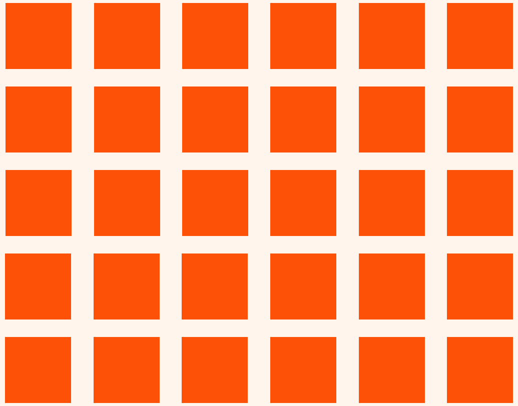
17.2% of teams are not using GenAI based IDEs

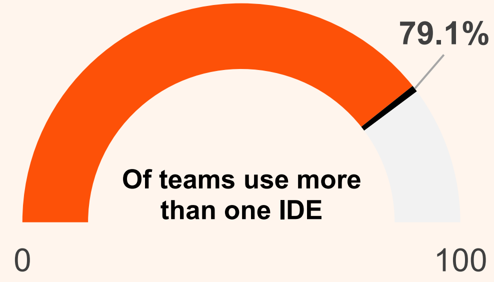


35.8% of teams are exploring and planning to adopt GenAI based IDEs



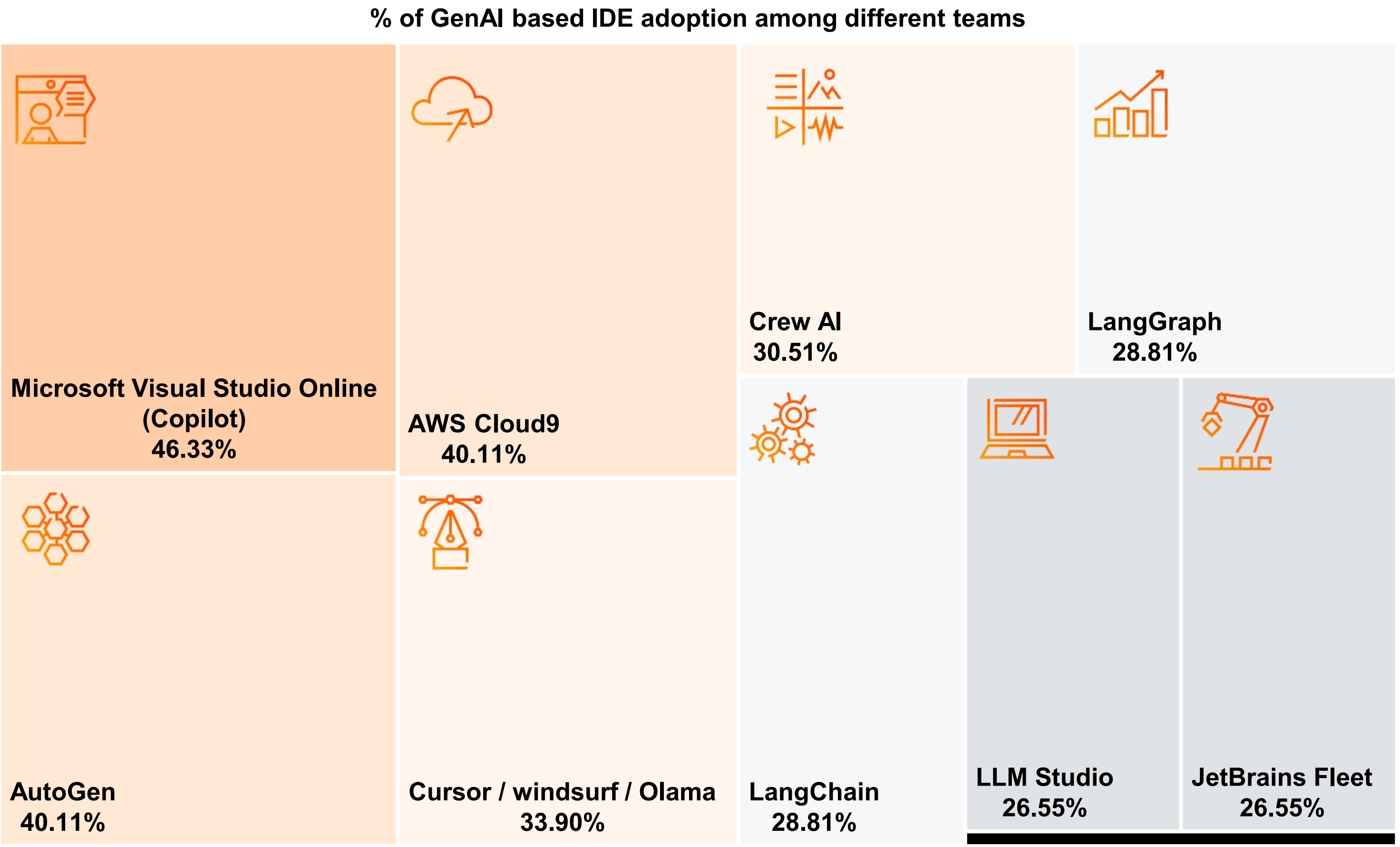
46.9% of teams are using GenAI based IDEs in their Agentic SDLC





Among teams that already use GenAI-powered IDEs (46.9%), cloud-hosted environments dominate (led by Visual Studio Online)

Indicator 18.2 - GenAI based IDE distribution Tree Map by all



Key findings:

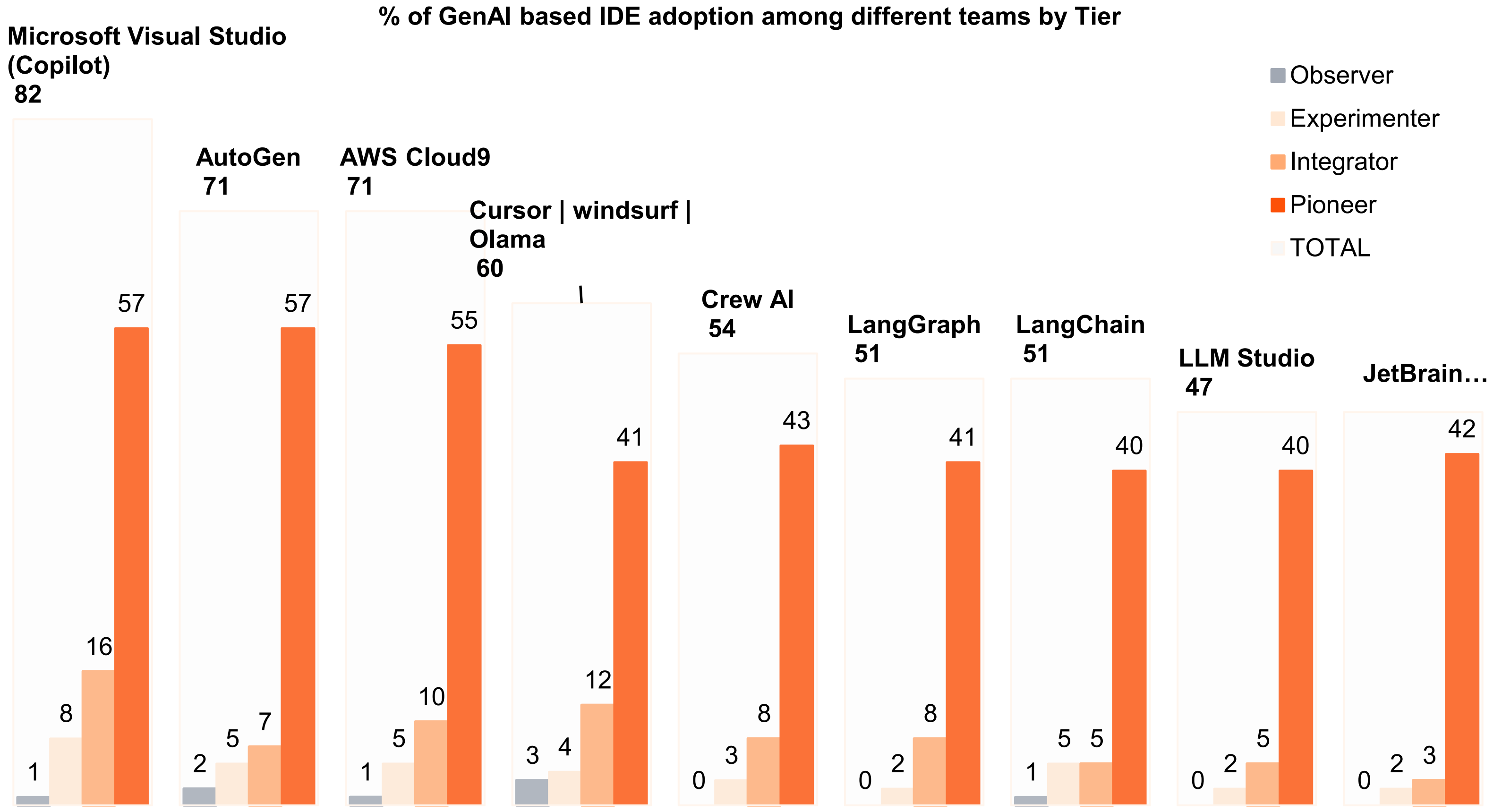
- **46.9% of teams** are using GenAI based IDE in their SDLC lifecycle with **35.8% exploring** adopting the same
- 79.1% of overall teams using GenAI based IDE have reported that they use more than one IDE in their SDLC processes
- Visual Studio Online is #1: 46.3 % of adopters tick it, edging out Auto Gen and AWS Cloud9 (40 % each)
- Pioneer teams ranked 71.8% in adopting GenAI based IDEs, while integrator teams stalls at 16.4% only, followed by experimenter teams at 8.5% and observer teams at 3.4% only; leaving an indication the higher the breadth of GenAI coverage on SDLC stage, the higher % of using GenAI based IDEs

Notes:

- Only **6 Observer** teams currently use GenAI IDEs, so Observer percentages carry high uncertainty
- n=177 only, as the rest of the overall population of this survey (377) haven't responded to this question
- IDE: Integrated Development Environment

Pioneers diversify into VS Online, Auto Gen, AWS Cloud9 and Crew AI, while **Observer** usage is patchy and highly concentrated

Indicator 18.3 - GenAI based IDE distribution Tree Map by SDLC Coverage Tier



Key findings:

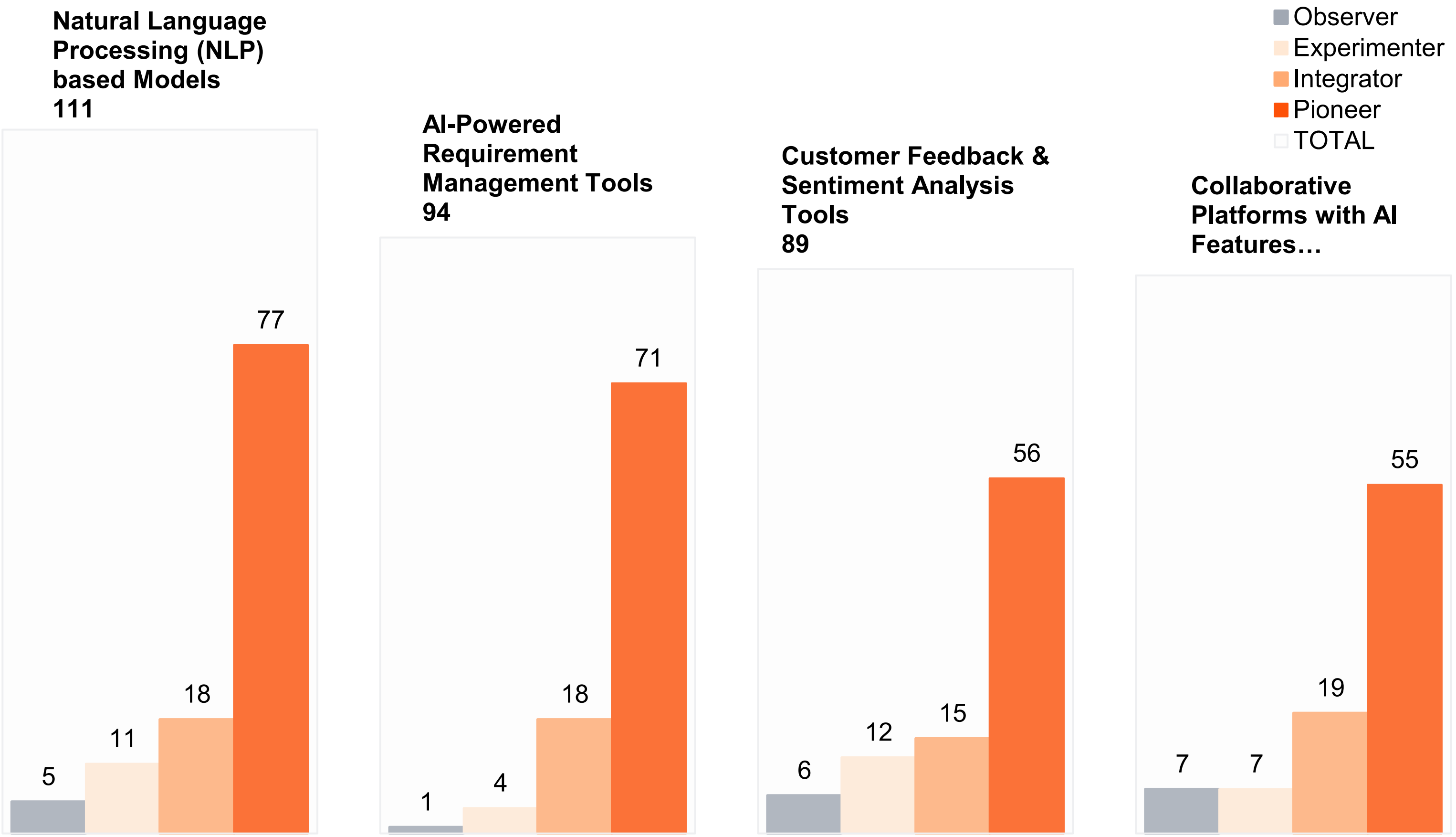
- **Tool diversity grows with maturity:** Pioneers exceed 30 % usage on nine named tools; Observers top out at 50 % on a single tool
- Pioneers ≥ 40 % on three tools; VS Online 45 %, Auto Gen 45 %, AWS Cloud9 43 %
- Observers are tiny and skewed; only 6 Observer teams adopters for GenAI IDE
- Hybrid tool-stacks appear after mid-maturity: Integrators already clear 24–55 % on six different tools; suggests breadth precedes diversification

Notes:

- n=177 only, as the rest of the overall population of this survey (377) haven't responded to this question
- Teams can select more than one IDE
- IDE: Integrated Development Environment

Which GenAI tools are used in your organization to automate the process of requirements gathering in SDLC lifecycle

Requirements management stage augmentation landscape



Key findings:

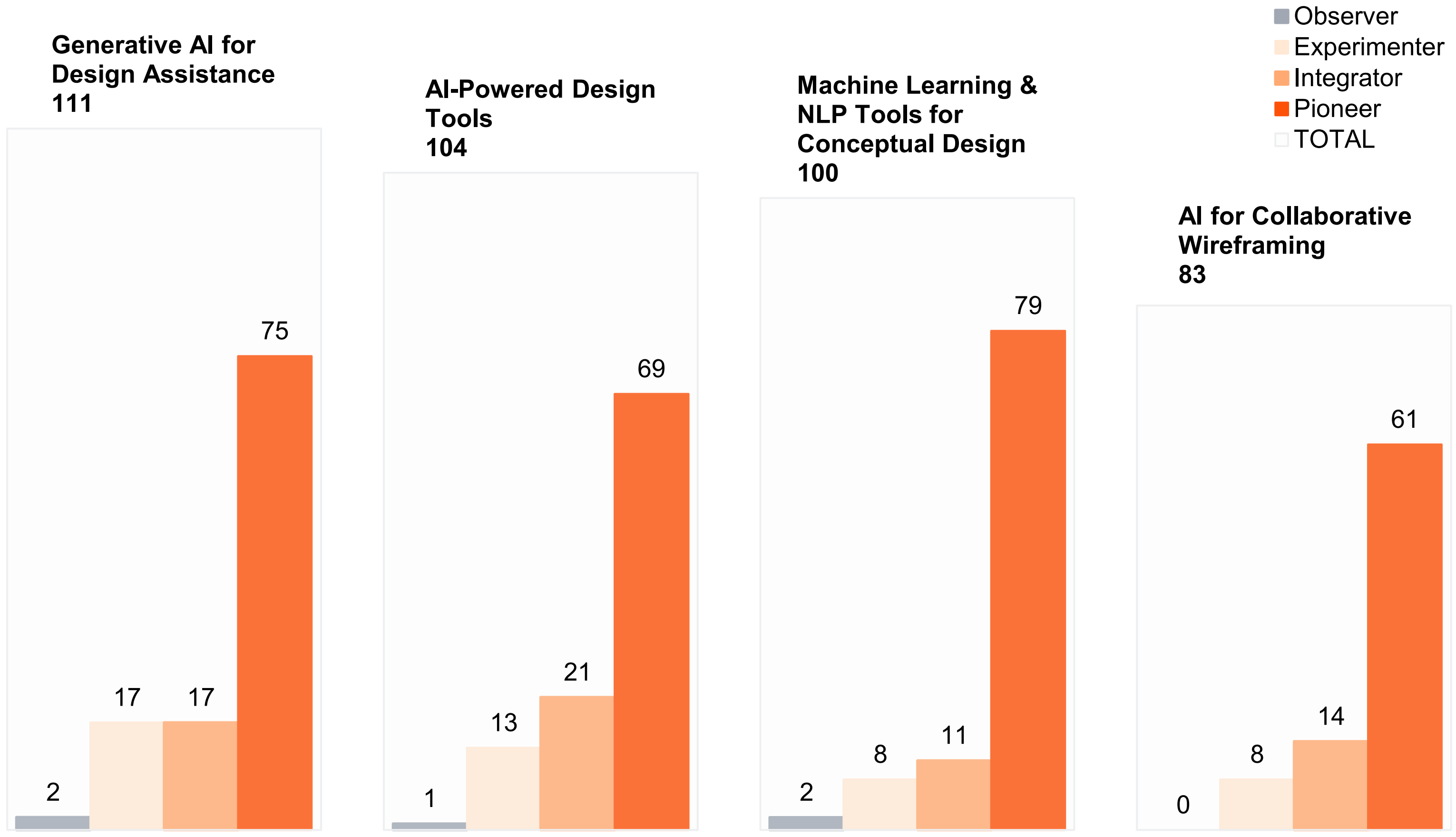
- over half of requirements-automation teams rely on an NLP parser to convert free-text into structured stories, while ~44 % augment that with specialized story generators. Custom prompt libraries are still niche (< 4 %), suggesting most teams lean on off-the-shelf models at this early stage

Notes:

- Total sample for teams who augment this SDLC stage is n=214 teams out of 377 survey respondents
- Teams can select more than one capability

Which GenAI tools are used in your organization to automate the design prototyping activities in SDLC lifecycle

Design Prototypes / Architecture stage augmentation landscape



Key findings:

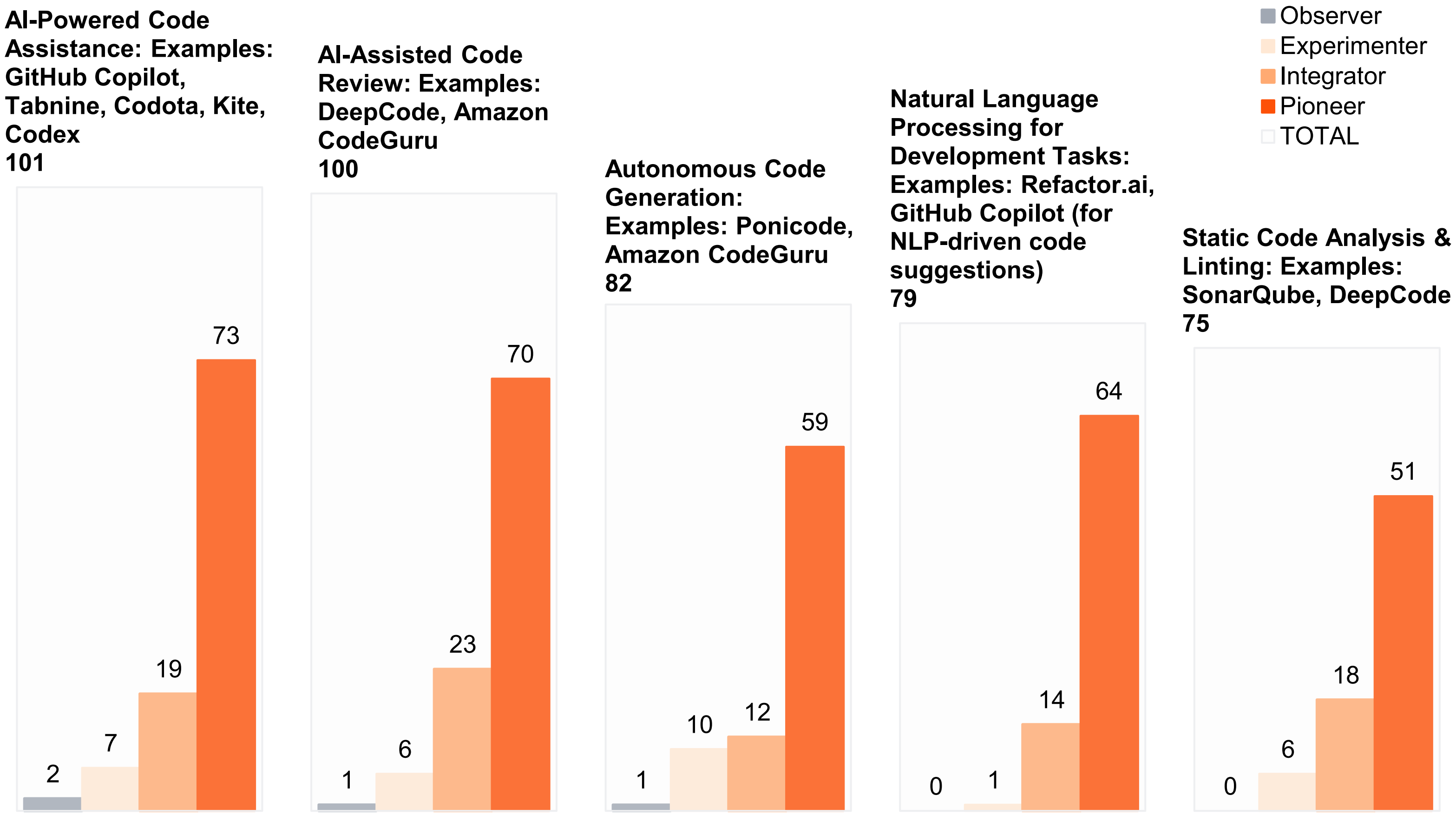
- Dual approach dominates: A slight majority (53 %) lean on code-assistant style tools (Copilot / Tabnine) while 50 % pair them with AI-augmented UI design suites (Figma, Sketch).
- **Concept-design ML tools** (AutoML, Pattern AI) hit 48 %, indicating early adoption of generative mock-ups and variant exploration
- **Collaborative wire-framing AI** (Miro / Whimsical) lags at 40 %, suggesting multi-stakeholder white-boarding is still mostly manual

Notes:

- Total sample for teams who augment this SDLC stage is n=208 teams out of 377 survey respondents
- Teams can select more than one capability

Which GenAI tools are used in your organization to augment the code generation activities in SDLC lifecycle

Coding stage augmentation landscape



Key findings:

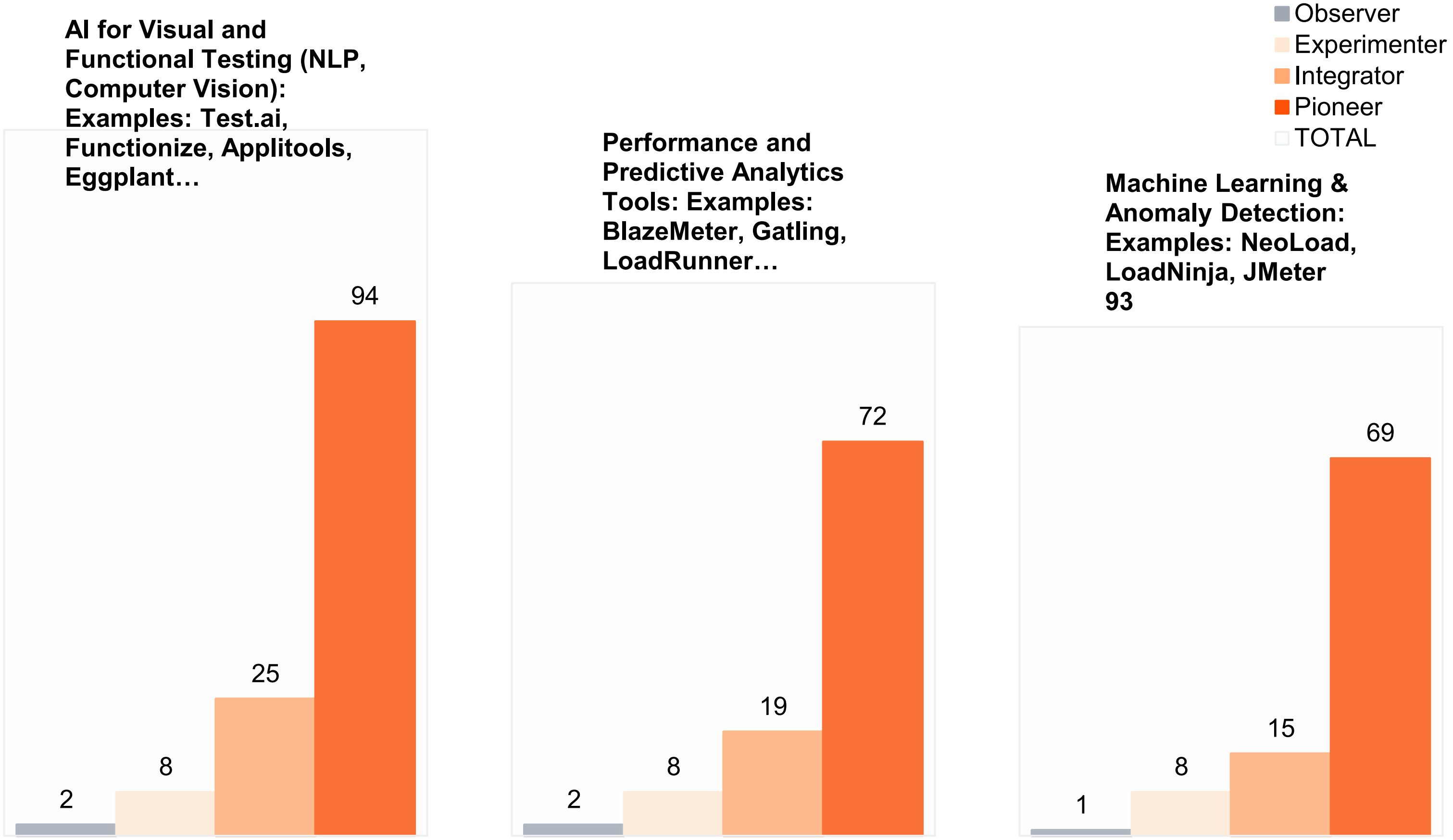
- **Dual leadership:** IDE completion and autonomous code agents run neck-and-neck (~48 % each), forming the core GenAI stack for coding
- **Shift left for quality:** nearly **30 %** already add AI code-review bots, signaling early governance adoption
- **Refactoring & API helpers** sit in the ~20 % range, while unit-test generators remain nascent (15 %)

Notes:

- Total sample for teams who augment this SDLC stage is n=208 teams out of 377 survey respondents
- Teams can select more than one capability

Which GenAI tools are used in your organization to augment the testing activities in SDLC lifecycle

Testing stage augmentation landscape



Key findings:

- **Visual & exploratory AI leads (63 %) — teams lean on GenAI to spot UI regressions and flow anomalies.**
- **Self-healing suites (Functionize / Test.ai) adopted by ~50 %, signalling move toward maintenance-free automation.**
- **Unit-test generation (38 %) and test-case design (30 %) still climbing, mirroring coding-stage adoption.**
- **Synthetic data (24 %) is emerging, crucial for privacy-compliant testing**

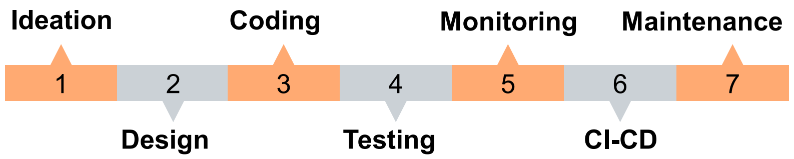
Notes:

- Total sample for teams who augment this SDLC stage is n=208 teams out of 377 survey respondents
- Teams can select more than one capability

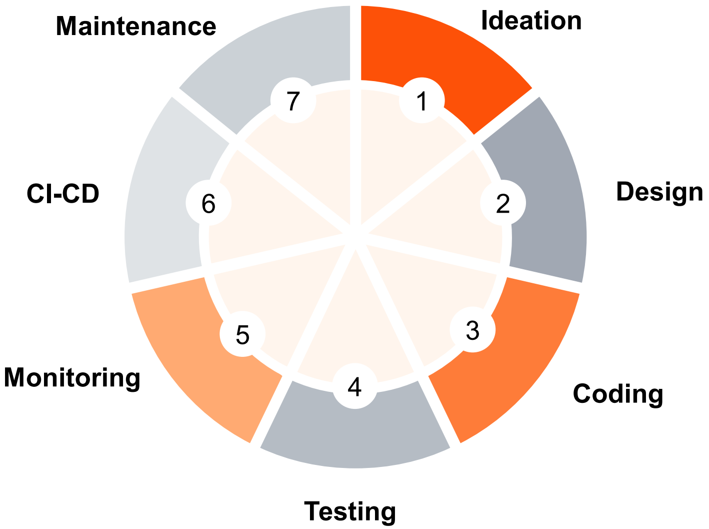
04 The forward future "Agentic SDLC" model

Introducing the future Agentic SDLC concept

Current SDLC

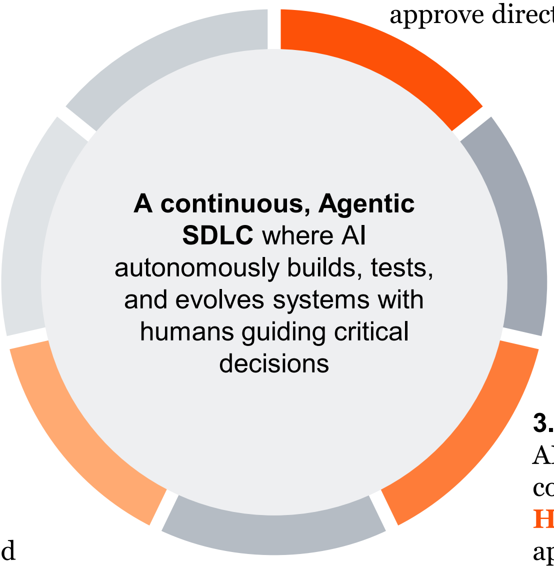


A linear, human-driven software development lifecycle focused on manual development, testing, and maintenance



An agile or iterative, human-driven software development lifecycle

Agentic SDLC



A continuous, Agentic SDLC where AI autonomously builds, tests, and evolves systems with humans guiding critical decisions

1. Autonomous ideation and requirements detailing
AI identifies needs, generates requirements, analyzes opportunities.
Human role: validate opportunities, refine scope, approve direction.

2. Autonomous architecture and design
AI generates system designs, model architectures, data flows.
Human role: review designs, enforce standards, ensure alignment & feasibility

3. Autonomous Development
AI writes code, builds backend and frontend components
Human role: perform code reviews, approve merges, check model behavior

4. Autonomous Testing (QA)
AI generates test cases, runs functional, performance, and security tests.
Human role: validate test coverage, assess risks, approve release readiness.

5. Autonomous Governed CI/CD
AI manages deployments, rollouts, risk scoring.
Human role: approve high-impact releases, override deploy decision if needed

6. Autonomous Observability
AI monitors software telemetry predicts failures, triggers self-healing.
Human role: investigate anomalies, validate actions, handle escalations.

7. Autonomous Evolution
AI refactors code, proposes enhancements and prioritized them.
Human role: approve plans, manage roadmap, perform periodic audits.

Top Enablers for Successful Agentic SDLC Adoption

(Derived from barrier gaps + Pioneer case practices)

Enabler	Evidence	Pioneer best-practice snapshot	Action for readers
1. Early compliance guard-rails – audit logging, privacy alignment, cross-border data controls	Barrier deep-dive: teams lacking ≥3 safeguards were 2× more likely to cite "Regulatory challenges"	Pioneer-UAE-5 deployed audit trails + regional data routing before scaling past 3 stages	Build a **"controls starter pack"** (logging, DPIA, geo-fencing) before the first production release
2. Full-stack observability with AI-driven anomaly detection	58% of Ops teams use AI observability; these teams show the **highest cadence & lowest defect-rates**	Pioneers layer Datadog Grok + canary AI to cut MTTR	Instrument across dev–test–prod; feed logs to LLM root-cause bots
3. Continuous refactoring & test-autonomy loops	Integrator/Pioneer tiers with self-healing tests saw **54% defect-rate reduction** vs 6% in Observers	Diffblue + Functionize keep tests green as code evolves	Pair code-assistants with auto-test regeneration; budget 5 % sprint capacity for refactor cycles
4. Domain-specific prompt & pattern libraries	62 % of Pioneers maintain curated prompt sets; correlates with **+22 pp productivity lift**	Custom LLM prompt libs accelerate requirements & coding stages	Stand up a **prompt guild**; version prompts like code; share best responses
5. Cross-functional "AI ops squad"	79 % of Pioneers report multi-role squads vs 34 % of Observers	Pioneer-UAE-5's squad blends CTO, QA lead, DBA, UX, FinOps	Form an **AI adoption pod** with reps from dev, QA, ops, compliance
6. Upskilling	41% of Pioneers cite lack of expertise or skilled personnel as the top barrier to entry, making upskilling the most critical enabler of sustained agentic SDLC adoption	GitHub hard-wires AI upskilling into delivery through Copilot-enabled, in-flow learning	Prepare for an agentic SDLC by investing **in governance, skills and operating models** that support higher levels of automation
7. Testing stage as the entry stage for Agentic SDLC adoption	Rising confidence in GenAI-driven testing (51% in observers, 77% in pioneers) indicates testing as a low-risk SDLC entry stage	Pioneers already leveraging AI augmented tools for visual and functional testing including computer vision and other NLP tasks for software delivery testing	Use testing as the controlled entry point to build confidence and readiness before **scaling GenAI across the SDLC**

The momentum of GenAI in the SDLC is clear, and closing the remaining gaps will unlock exponential value

GenAI is already creating step-change improvements across the SDLC

Early gains demonstrate the compounding effect GenAI can unlock when applied at scale

Isolated GenAI wins to an orchestrated, end-to-end agentic ecosystem

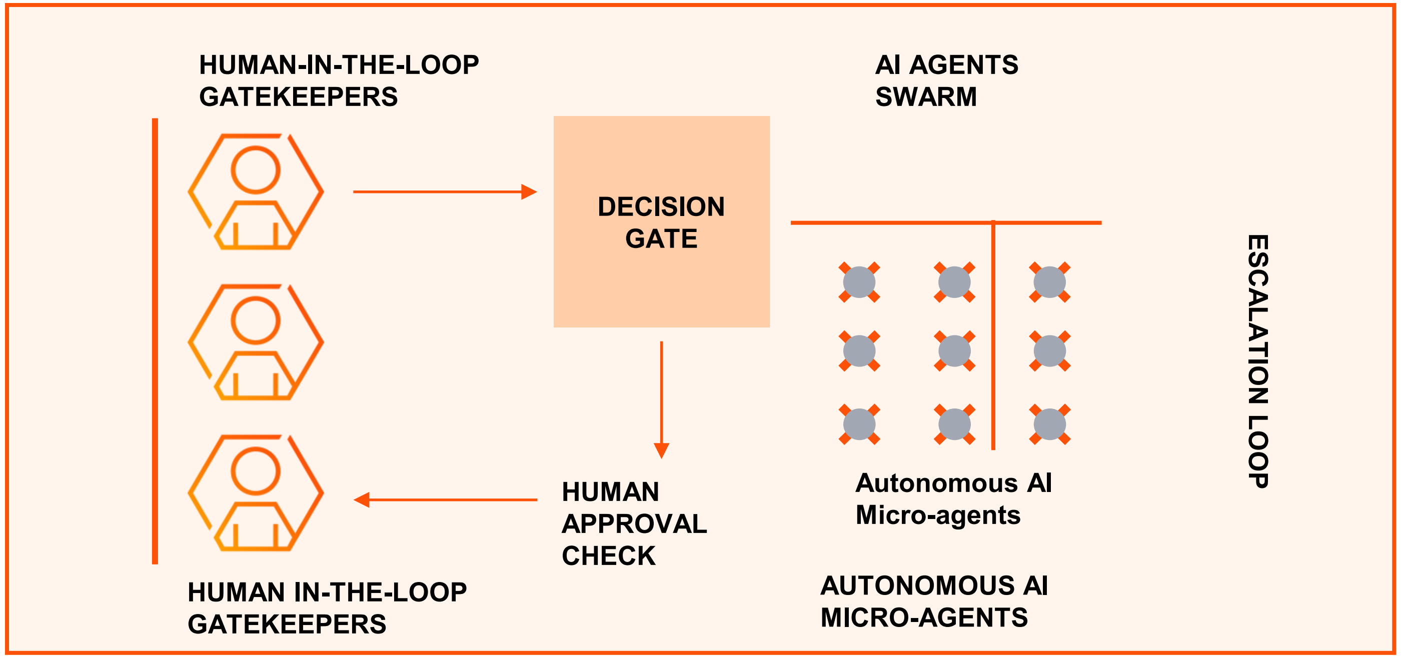
Shared vector memory lets breakthroughs in one phase ripple across the entire cycle instantly

Incremental improvements to a self-accelerating SDLC flywheel

Autonomous agents reinforcing every stage, delivery accelerates faster with each cycle.

Introduce the concept of Future operating model: the “Self-Building Application”

Following the accelerated improvements and breakthrough in AI towards reaching a **super intelligence**


HUMAN-IN-THE-LOOP GATEKEEPERS
AI AGENTS SWARM
DECISION GATE
ESCALATION LOOP
HUMAN APPROVAL CHECK
Autonomous AI Micro-agents
HUMAN IN-THE-LOOP GATEKEEPERS
AUTONOMOUS AI MICRO-AGENTS

Many ideas are becoming reality and services in less than a year building using and on top of AI agents where it is becoming the new status que for browsing the internet, guiding scientific discoveries, and running different business discipline

KPIs to manage the journey

- **Breadth index:** # SDLC stages augmented
- **Release cadence:** versions/year
- **Time/quality:** composite speed & quality scores
- **Defect density trend:** pre/post adoption; share of repos with tracked defects
- **Cost goals vs realized:** pair OPEX targets with defect-rate reductions

Quantitative forecast: When will a “Fully Augmented SDLC” become the norm?

“By 2027 more than half of Middle-East software teams will run a fully augmented SDLC; by 2029 **two-thirds will**.”

Survey signals

•**38 %** of all respondents already sit in the **Pioneer** tier (≥ 6 stages automated).

•**62 %** are **interested in agentic AI apps** (Q38).

•**76 %** are **likely to raise GenAI investment** within 24 months (Q32).

External adoption curve

- Gartner & Deloitte both expect *autonomous* GenAI agents to reach **“early-majority” adoption by 2029-2030** Deloitte Italia Gartner.
- Forbes notes GenAI’s adoption rate is “one of the fastest in tech history,” outpacing cloud by ~40 % Forbes.

Simple projection

Year	Assumed cumulative Pioneer share	Rationale
2025 (survey)	**38 %**	Baseline measurement
2026	46 %	8-point uplift driven by 76 % “Likely” investors
2027	54 %	Early-majority threshold crossed
2029 (median)	**65 %**	Syncs with Deloitte/Gartner agentic milestone
2030	~70 %	Plateau as late adopters close gap

Method: logistic growth anchored at 38 % (2025) and asymptoting near 75 % (typical enterprise-tech ceiling). The **median target year** for a fully augmented SDLC (≥ 50 % of orgs in Pioneer tier) is **2027**, while **two-thirds adoption arrives by ~2029**.

05 Impacted SDLC Talents & Skills: roles sunset, roles born

Roles most at risk of “sunset” vs roles most likely to emerge

Likely to shrink / be heavily automated[1]	Evidence (Q36 % Impacted + industry commentary)	New / growth roles[2]	External signals
Software developer / engineer	53.8 % of respondents say GenAI will impact them; GitHub CEO predicts AI will automate “up to 90 % of code writing within a few years” (Business Insider)	**Prompt & LLM engineer** (designing, curating, versioning prompts) **Context Engineer** Builds and manages the contextual foundations that AI agents rely on	62 % of Pioneers already maintain prompt libraries; job-boards show triple-digit growth in “Prompt Engineer” postings (McKinsey & Company)
QA engineer / tester	34.5 % impact; 63 % of GenAI test teams now use visual/exploratory AI (Indicator 16)	**AI test-orchestration lead** builds self-healing test suites **Vibe-coder** Define requirements, plan, build, test, and review results through natural language either via keyboard or voice commands	Many startups exist today that are leveraging the computer vision and GenAI models to perform generative testing and code fixes for software components (ex. Replit, Loveable, Factroy.AI)
Database / system administrator	34.2% & 26.3 % impact; Ops stage shows 58 % adoption of AIOps platforms → routine tuning & patching automated	**xOps AI Analyst** Full-stack observability **AIOps Engineer** orchestrates AI agents across data, systems architecture, and operations	IDC notes FinOps AI as fastest-growing sub-segment of AI platforms (IDC)
Project manager / team lead	Only 26 % impact today, but agentic orchestration threatens task-tracking duties	**AI governance & risk lead** sets policy, audit & compliance for GenAI pipelines	Regulatory-compliance barrier biggest delta in survey; governance hiring surging in 2025 (Gartner)

[1] “Sunset” here means the task mix shifts sharply toward oversight, not that the job disappears.

[2] Emergent roles drawn from Pioneer practice, job-board scans and analyst forecasts.

06 Appendix

We have analyzed the survey and created a list of indicators to measure the current impact of GenAI on software development lifecycle, this would be an anchor for future progress and impact re-assess type of work (1/3)

#	Indicator	Description
01	Indicator 1- **Overall adoption level (%) of GenAI in software development by Region**	Overall adoption level of GenAI in software development
02	Indicator 2 · **How integrated are GenAI tools into your organization's SDLC?**	How integrated are GenAI tools into your organization's SDLC?
03	Indicator 3 · **Stage-Coverage Tiers Index (Breadth of GenAI Automation)**	Based on 7 questions for integrating GenAI in each SDLC stage
04	Indicator 4 · **Release Cadence vs Stage-Coverage**	Release Cadence vs Stage-Coverage
05	Indicator 5 - **Time saved |** Mean Score by SDLC Coverage Tier	Speed / Time Saved Mean Score by SDLC Coverage Tier
06	Indicator 6 - **Quality Boost |** Mean score by SDLC Coverage Tier	Quality Boost | Mean score by SDLC Coverage Tier
07	Indicator 7 · Most Augmented SDLC Stages	Augmented SDLC, respondents using or exploring GenAI at each stage
08	Indicator 8.1- Barrier Intensity: Overall Barrier Prevalence Cut (whole-sample %)	Barrier Intensity: Overall Barrier Prevalence
09	Indicator 8.2- Barrier Intensity: Barrier Gap by Tier Cut (within-tier %)	Barrier Intensity: Barrier Gap by Tier Cut
10	Indicator 8.3- Barrier Intensity: Barrier Index (Tier % – Overall %)	Barrier Intensity: Barrier Index (Tier % – Overall %)
11	Indicator 9.1-Why teams adopt GenAI? Overall, Drivers Prevalence Cut (whole-sample)	Adoption drivers distribution among different teams (whole-sample)

We have analyzed the survey and created a list of indicators to measure the current impact of GenAI on software development lifecycle, this would be an anchor for future progress and impact re-assess type of work (2/3)

#	Indicator	Description
12	Indicator 9. 2 - Why teams adopt GenAI? Drivers By Tier (Pioneer vs Observer, (within-tier %)	Adoption drivers distribution among different teams (by tier)
13	Indicator 10.1 - GenAI Impact on Defect Rate / Bug Density	Teams who measured the Defects rate before and after GenAI Adoption
14	Indicator 10.2 - GenAI Impact on Defect Rate / Bug Density (within-tier %)	Teams who measured the Defects rate before and after GenAI Adoption (within-tier %)
15	Indicator 10.3 - GenAI Impact on Defect Rate / Bug Density (whole-sample %)	Teams who measured the Defects rate before and after GenAI Adoption (whole-sample %)
16	Indicator 11.1 · Investment Sentiment	likelihood to boost GenAI-tool spend incorporating it with SDLC lifecycle
17	Indicator 11.2 - Investment Sentiment by coverage tier	likelihood to boost GenAI-tool spend incorporating it with SDLC lifecycle by coverage tier
18	Indicator 12 - Interest in Developing Agentic-AI Applications	Developing or using agentic AI apps by SDLC coverage tier
19	Indicator 13 - Confidence in an Agentic SDLC Future	GenAI-driven autonomous development will eventually replace traditional SDLC in the future sentiment
20	Indicator 14.1 - General sentiment on GenAI adoption within SDLC	General sentiment on GenAI adoption within SDLC
21	Indicator 14.2 - General sentiment on GenAI adoption within SDLC – Spotlights	General sentiment on GenAI adoption within SDLC (Spotlights)

We have analyzed the survey and created a list of indicators to measure the current impact of GenAI on software development lifecycle, this would be an anchor for future progress and impact re-assess type of work (3/3)

#	Indicator	Description
22	Indicator 15 - Wishlist for Future GenAI SDLC Features	top-feature appetite by maturity tier
23	Indicator 16.1 - Roles Most Impacted by Agentic SDLC	Roles Most Impacted by GenAI in the SDLC
24	Indicator 16.2 - Roles Most Impacted by Agentic SDLC – By SDLC Tier Cut	Roles Most Impacted by GenAI in the SDLC (by tier cut)
25	Indicator 17 - Future Outlook & Sentiment Toward Agentic SDLC	Future Outlook & Sentiment Toward GenAI in the SDLC
26	Indicator 18.1 - GenAI based IDE distribution among respondents teams	% of GenAI based IDE adoption among different teams
27	Indicator 18.2 - GenAI based IDE distribution Tree Map by all	% of GenAI based IDE distribution Tree Map by all
28	Indicator 18.3 - GenAI based IDE distribution Tree Map by SDLC Coverage Tier	% of GenAI based IDE distribution Tree Map by all by SDLC coverage tier

Thank you

About PwC

At PwC, we help clients build trust and reinvent so they can turn complexity into competitive advantage. We're a tech-forward, people-empowered network with more than 370,000 people in 149 countries. Across audit and assurance, tax and legal, deals and consulting we help build, accelerate and sustain momentum. Find out more at www.pwc.com.

With over 12,000 people across 12 countries in 30 offices, PwC Middle East combines deep regional insight with global expertise to help clients solve complex problems, drive transformation and achieve sustained outcomes. Learn more at www.pwc.com/me.

PwC refers to the PwC network and/or one or more of its member firms, each of which is a separate legal entity. Please see www.pwc.com/structure for further details.