AI-Assisted Programming and Labor Demand

Lindsey Raymond Eleanor Dillon

MIT | Microsoft Research

July 2026

What does AI progress mean for workers?

- AI can improve task-level productivity
  - $\rightarrow$ AI capabilities are advancing rapidly, programming is at the frontier

What does AI progress mean for workers?

- AI can improve task-level productivity
  - → AI capabilities are advancing rapidly, programming is at the frontier
- Open question: what do these improvements in productivity mean for workers?
  - → No near term effects?
  - → Substantial employment decline?
  - → More employment/work?

What does AI progress mean for workers?

- AI can improve task-level productivity
  - → AI capabilities are advancing rapidly, programming is at the frontier
- Open question: what do these improvements in productivity mean for workers?
  - → No near term effects?
  - → Substantial employment decline?
  - → More employment/work?
- The central question is not only whether labor demand changes, but how and why
  - → **Margins:** do firms adjust through hiring or separations?
  - → **Speed:** is adjustment gradual or immediate?
  - → **Incidence:** who is affected?

This paper: enterprise AI adoption linked to labor demand

- **Setting:** GitHub Copilot for Enterprise and GitHub Enterprise customers
  - → GitHub Enterprise: source code management platform for large, IT intensive firms with demanding compliance requirements

▸ Labor market size: BLS SOC codes

This paper: enterprise AI adoption linked to labor demand

- **Setting:** GitHub Copilot for Enterprise and GitHub Enterprise customers
  - → GitHub Enterprise: source code management platform for large, IT intensive firms with demanding compliance requirements
- **Identification:** Natural experiment among existing GitHub Enterprise customers
  - → Software sales cycles create quasi-exogenous variation in Copilot adoption

▸ Labor market size: BLS SOC codes

This paper: enterprise AI adoption linked to labor demand

- **Setting:** GitHub Copilot for Enterprise and GitHub Enterprise customers
  - → GitHub Enterprise: source code management platform for large, IT intensive firms with demanding compliance requirements
- **Identification:** Natural experiment among existing GitHub Enterprise customers
  - → Software sales cycles create quasi-exogenous variation in Copilot adoption
- **Sample:** 5,425 large, software-intensive firms
  - → GitHub Enterprise customers headquartered in the Americas/Europe
  - → Employ $\sim$45M workers globally
  - → $\approx$ 38% of U.S. tech-related employment

▸ Labor market size: BLS SOC codes

Research questions and findings

1. **Does Copilot adoption change firms' demand for technical labor?**

Research questions and findings

1. **Does Copilot adoption change firms' demand for technical labor?**
   - $\rightarrow$ Among contract reviewing timing compliers, tech hiring falls by 36%
   - $\rightarrow$ No detectable changes in separations

Research questions and findings

1. **Does Copilot adoption change firms' demand for technical labor?**
   - → Among contract reviewing timing compliers, tech hiring falls by 36%
   - → No detectable changes in separations
2. **How does AI adoption change the composition of tech hiring?**

Research questions and findings

1. **Does Copilot adoption change firms' demand for technical labor?**
   - → Among contract reviewing timing compliers, tech hiring falls by 36%
   - → No detectable changes in separations

2. **How does AI adoption change the composition of tech hiring?**
   - → Average tech hire is less code intensive
   - → Hiring composition is more senior in code intensive jobs
   - → 45% decline in junior SWE hiring
   - → No changes for Systems/Infrastructure juniors

Contribution

1. **Causal evidence of the impact of AI on work**
   → Building on work that asks how AI impacts work with either imputed AI adoption and observational comparisons (Acemoglu et al., 2022; Baird et al., 2024; Brynjolfsson et al., 2025; Eloundou et al., 2024; Hampole et al., 2025; Hosseini and Lichtinger, 2025; Humlum and Vestergaard, 2025; Kharazian et al., 2026; Westby et al., 2026; Yotzov et al., 2026)

Contribution

1. **Causal evidence of the impact of AI on work**
   → Building on work that asks how AI impacts work with either imputed AI adoption and observational comparisons (Acemoglu et al., 2022; Baird et al., 2024; Brynjolfsson et al., 2025; Eloundou et al., 2024; Hampole et al., 2025; Hosseini and Lichtinger, 2025; Humlum and Vestergaard, 2025; Kharazian et al., 2026; Westby et al., 2026; Yotzov et al., 2026)
   → We contribute a clean natural experiment with AI adoption data

Contribution

1. **Causal evidence of the impact of AI on work**
   - → Building on work that asks how AI impacts work with either imputed AI adoption and observational comparisons (Acemoglu et al., 2022; Baird et al., 2024; Brynjolfsson et al., 2025; Eloundou et al., 2024; Hampole et al., 2025; Hosseini and Lichtinger, 2025; Humlum and Vestergaard, 2025; Kharazian et al., 2026; Westby et al., 2026; Yotzov et al., 2026)
   - → We contribute a clean natural experiment with AI adoption data
2. **Evidence on firm-level effects of technological change**

Contribution

1. **Causal evidence of the impact of AI on work**
   → Building on work that asks how AI impacts work with either imputed AI adoption and observational comparisons (Acemoglu et al., 2022; Baird et al., 2024; Brynjolfsson et al., 2025; Eloundou et al., 2024; Hampole et al., 2025; Hosseini and Lichtinger, 2025; Humlum and Vestergaard, 2025; Kharazian et al., 2026; Westby et al., 2026; Yotzov et al., 2026)
   → We contribute a clean natural experiment with AI adoption data

2. **Evidence on firm-level effects of technological change**
   → Existing work studying IT, robots, telephones etc. (Acemoglu and D. Autor, 2011; Acemoglu and Restrepo, 2020; Akerman et al., 2015; D. H. Autor et al., 2003a,b; Bartel et al., 2007; Bessen et al., 2020; Bresnahan et al., 2002; David et al., 1998; Feigenbaum and Gross, 2024; Forman et al., 2012; Goldin and Katz, 2008; Goos et al., 2014; Hirvonen et al., 2026)

Contribution

1. **Causal evidence of the impact of AI on work**
   → Building on work that asks how AI impacts work with either imputed AI adoption and observational comparisons (Acemoglu et al., 2022; Baird et al., 2024; Brynjolfsson et al., 2025; Eloundou et al., 2024; Hampole et al., 2025; Hosseini and Lichtinger, 2025; Humlum and Vestergaard, 2025; Kharazian et al., 2026; Westby et al., 2026; Yotzov et al., 2026)
   → We contribute a clean natural experiment with AI adoption data

2. **Evidence on firm-level effects of technological change**
   → Existing work studying IT, robots, telephones etc. (Acemoglu and D. Autor, 2011; Acemoglu and Restrepo, 2020; Akerman et al., 2015; D. H. Autor et al., 2003a,b; Bartel et al., 2007; Bessen et al., 2020; Bresnahan et al., 2002; David et al., 1998; Feigenbaum and Gross, 2024; Forman et al., 2012; Goldin and Katz, 2008; Goos et al., 2014; Hirvonen et al., 2026)
   → We contribute evidence on AI coding assistants on technical workers

Outline

Setting and Treatment: Copilot for Enterprise

Sample and Data

Identification

Results

Setting: Enterprise software development

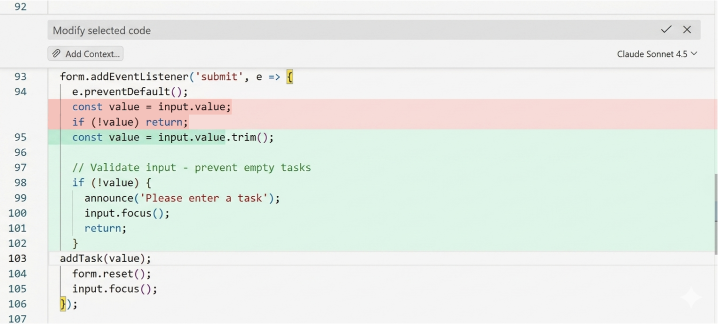
- **Enterprise software development**
  - → Multiple teams contribute to shared, interdependent codebases
  - → Business-critical systems must be reliable, secure, scalable, and with demanding compliance requirements

Setting: Enterprise software development

- **Enterprise software development**
  → Multiple teams contribute to shared, interdependent codebases
  → Business-critical systems must be reliable, secure, scalable, and with demanding compliance requirements

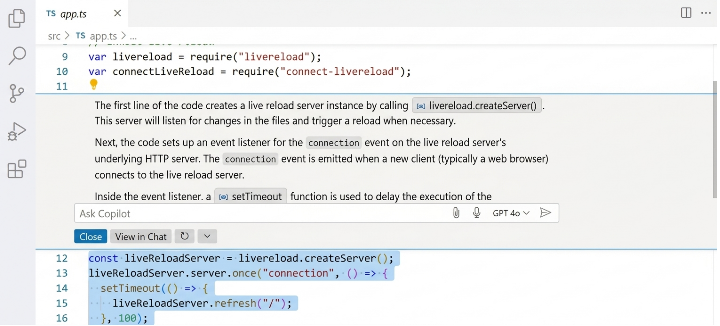
- **GitHub Enterprise**
  → A GitHub product for coordinating development across large organizations
  → **Workflow:** code changes, review, testing, security, and deployment
  → **Governance:** access controls, policies, and audit logs
  → **Not an AI tool**

Copilot brought AI coding assistance into enterprise workflows



1. Custom generated coding suggestions
2. A natural-language interface to the codebase

Copilot brought AI coding assistance into enterprise workflows



1. Custom generated coding suggestions
2. A natural-language interface to the codebase

Outline

Setting and Treatment: Copilot for Enterprise

Sample and Data

Identification

Results

Sample and Data

Sources

- GitHub: Enterprise customers, contract cycle, and Copilot purchases
- Revelio Labs: LinkedIn worker histories, job titles, and employment flows

Sample and Data

Sources

- GitHub: Enterprise customers, contract cycle, and Copilot purchases
- Revelio Labs: LinkedIn worker histories, job titles, and employment flows

Analysis dataset

- Balanced firm-month panel, January 2021–June 2025
- AI adoption: first month with > 0 Copilot seats
- Outcomes: headcount, hiring, and separations by firm $\times$ role $\times$ seniority

Sample and Data

Sources

- GitHub: Enterprise customers, contract cycle, and Copilot purchases
- Revelio Labs: LinkedIn worker histories, job titles, and employment flows

Analysis dataset

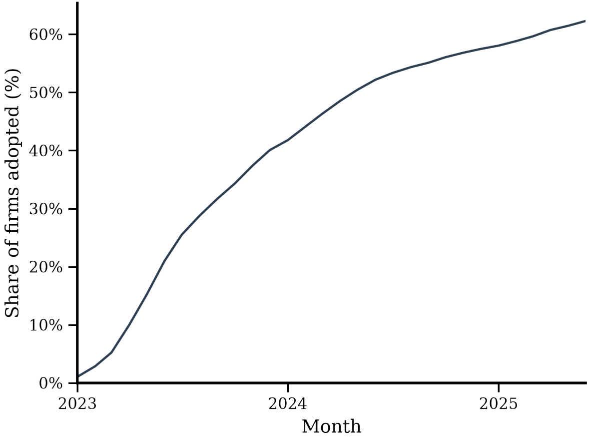
- Balanced firm-month panel, January 2021–June 2025
- AI adoption: first month with > 0 Copilot seats
- Outcomes: headcount, hiring, and separations by firm $\times$ role $\times$ seniority

Sample

- 5,425 GitHub Enterprise customers headquartered in the Americas and Europe
- 62% headquartered in the U.S.; multinationals
- Mean firm size: 8,292 workers globally; mean firm age: 47 years
- $\approx$ 12% technical workers, where 90% with a BA+, 30% an academic masters

`Summary stats` `Data validation`

62% of sample adopts Copilot by June 2025



- Copilot for Enterprise launched in January 2023
- Enterprise coding assistant market: Copilot, Amazon CodeWhisperer

Outline

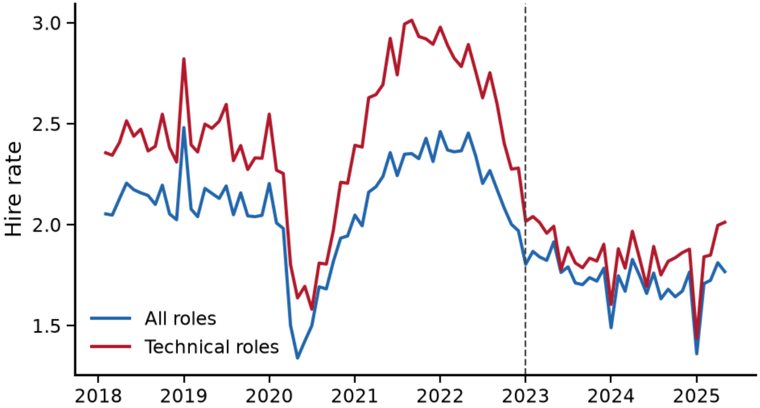
Setting and Treatment: Copilot for Enterprise

Sample and Data

Identification

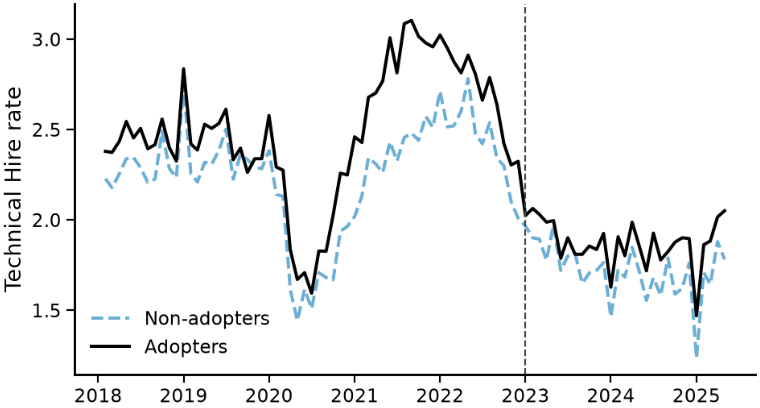
Results

Observational analysis is challenging



- Hire rate $= \frac{\text{hires}_{t,i}}{\text{headcount}_{t-1,i}}$ for firm i in month t

Observational analysis is challenging



- Hire rate $= \frac{\text{hires}_{t,i}}{\text{headcount}_{t-1,i}}$ for firm i in month t

SWE headcount was already growing before firms adopted Copilot

- OLS estimates of the impact of Copilot adoption on software engineering employment

$$\rightarrow \ln(L_{it}^{\text{SWE}}) = \alpha_i + \lambda_t + \beta\, \mathbb{1}\{t \geq A_i\} + \varepsilon_{it}$$

SWE headcount was already growing before firms adopted Copilot

- OLS estimates of the impact of Copilot adoption on software engineering employment

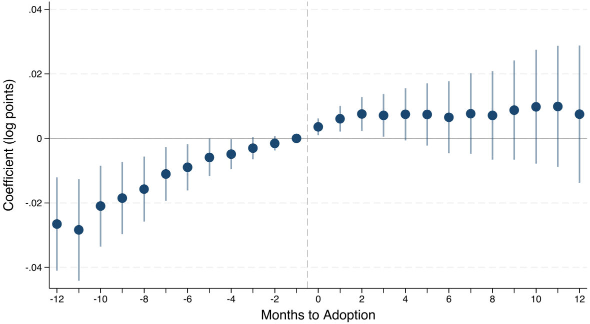
  $\rightarrow \ln(L_{it}^{\mathrm{SWE}}) = \alpha_i + \lambda_t + \beta\, \mathbb{1}\{t \geq A_i\} + \varepsilon_{it}$

- Copilot adoption associated with a 1.8% increase in SWE employment

  $\rightarrow \widehat{\beta^{\mathrm{SWE}}} = 0.018$ (SE $= 0.007$; $p = 0.010$)

SWE headcount was already growing before firms adopted Copilot

- OLS estimates of the impact of Copilot adoption on software engineering employment
  $\rightarrow$ $\ln(L_{it}^{\mathrm{SWE}}) = \alpha_i + \lambda_t + \beta\, \mathbb{1}\{t \geq A_i\} + \varepsilon_{it}$
- Copilot adoption associated with a 1.8% increase in SWE employment
  $\rightarrow$ $\widehat{\beta^{\mathrm{SWE}}} = 0.018$ (SE $= 0.007$; $p = 0.010$)



GitHub Enterprise contract reviews created Copilot sales opportunities

- Enterprise procurement creates organizational frictions to software adoption
  - → Security and compliance restrict firms to approved "enterprise grade" software
  - → Decision maker: IT Procurement Team

GitHub Enterprise contract reviews created Copilot sales opportunities

- Enterprise procurement creates organizational frictions to software adoption
  - → Security and compliance restrict firms to approved "enterprise grade" software
  - → Decision maker: IT Procurement Team
- GitHub Enterprise contracts have predetermined review dates
  - → GitHub Enterprise review dates were set before Copilot launched

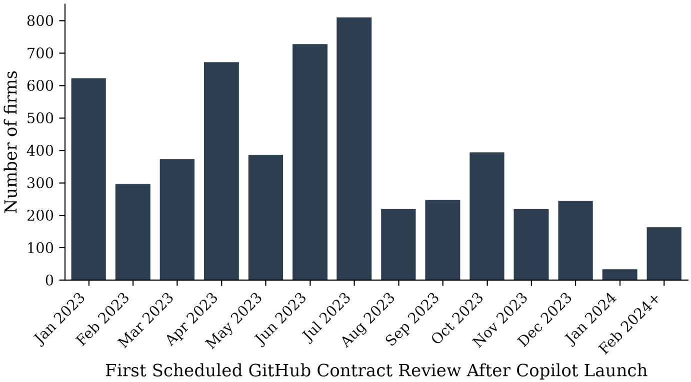
GitHub Enterprise contract reviews created Copilot sales opportunities

- Enterprise procurement creates organizational frictions to software adoption
  - $\rightarrow$ Security and compliance restrict firms to approved "enterprise grade" software
  - $\rightarrow$ Decision maker: IT Procurement Team
- GitHub Enterprise contracts have predetermined review dates
  - $\rightarrow$ GitHub Enterprise review dates were set before Copilot launched
- Scheduled contract reviews create Copilot sales opportunities

  Earlier GitHub contract review $\Rightarrow$ earlier exposure to Copilot

Distribution of predetermined GitHub review dates at Copilot launch

- R_i: a firm's first scheduled **GitHub Enterprise contract review** date after Copilot launch
- Source of variation: pre-determined timing of contract review



Empirical design: instrumented differences-in-differences

First stage:

$$\underbrace{A_{it}}_{\substack{\text{Copilot}\\\text{adoption}}} = \underbrace{\alpha_i}_{\text{firm FE}} + \underbrace{\lambda_t}_{\text{year-month FE}} + \sum_{\ell \neq -1} \underbrace{\pi_\ell}_{\substack{\text{effect at}\\\text{horizon } \ell}} \underbrace{\mathbf{1}\{t - R_i = \ell\}}_{\substack{\text{event time relative}\\\text{to contract review}}} + \varepsilon_{it}$$

Placebo occupations | Estimand: compliers vs. switchers

Empirical design: instrumented differences-in-differences

First stage:

$$A_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \pi_\ell \mathbf{1}\{t - R_i = \ell\} + \varepsilon_{it}$$

Empirical design: instrumented differences-in-differences

First stage:

$$A_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \pi_\ell \mathbf{1}\{t - R_i = \ell\} + \varepsilon_{it}$$

Reduced form:

$$\underbrace{Y_{it}}_{\text{employment outcome}} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \underbrace{\rho_\ell}_{\text{reduced-form effect at horizon } \ell} \mathbf{1}\{t - R_i = \ell\} + u_{it}$$

Empirical design: instrumented differences-in-differences

First stage:

$$A_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \pi_\ell \mathbf{1}\{t - R_i = \ell\} + \varepsilon_{it}$$

Reduced form:

$$Y_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \rho_\ell \mathbf{1}\{t - R_i = \ell\} + u_{it}$$

- Design: instrumented differences-in-differences (DID-IV)
- Regressions are unweighted; standard errors are clustered by firm
- Heterogeneity-robust estimator of Borusyak et al. (2024)

Placebo occupations
Estimand: compliers vs. switchers

Empirical design: instrumented differences-in-differences

First stage:

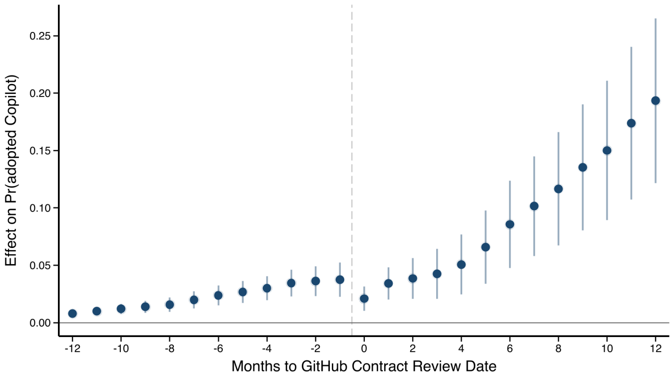
$$A_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \pi_\ell \mathbf{1}\{t - R_i = \ell\} + \varepsilon_{it}$$

Reduced form:

$$Y_{it} = \alpha_i + \lambda_t + \sum_{\ell \neq -1} \rho_\ell \mathbf{1}\{t - R_i = \ell\} + u_{it}$$

- Design: instrumented differences-in-differences (DID-IV)
- Regressions are unweighted; standard errors are clustered by firm
- Heterogeneity-robust estimator of Borusyak et al. (2024)
- DID-IV estimate: $\theta_{12} = \frac{\rho_{12}}{\pi_{12}}$: effect of Copilot, among contract review timing compliers Miyaji (2024)
  - $\rightarrow$ Firm block bootstrap confidence intervals
  - $\rightarrow$ Exclusion restriction: contract review timing did not impact employment outcomes except through Copilot
  - $\rightarrow$ Monotonicity; no anticipation; no carryover; parallel trends in adoption/employment outcomes; relevance

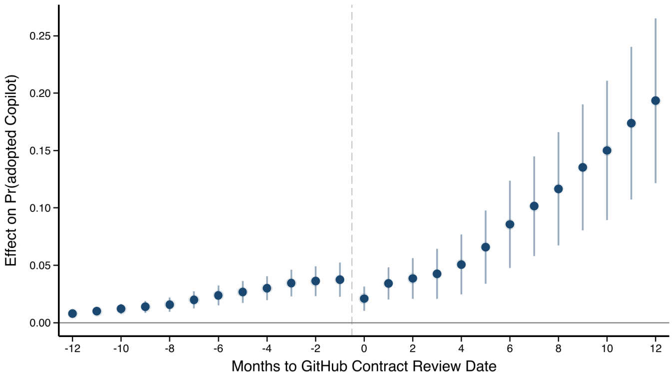
Copilot adoption increases after GitHub Enterprise contract review



- Contract review raises 12-month Copilot adoption by 19.3 p.p. $(F \approx 28)$

Anticipation test

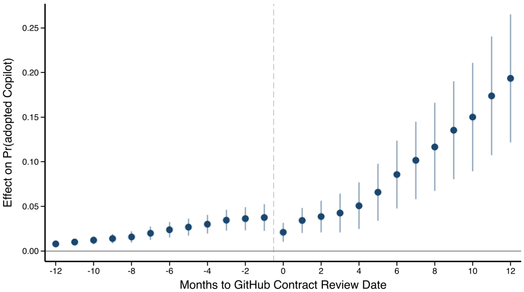
Copilot adoption increases after GitHub Enterprise contract review



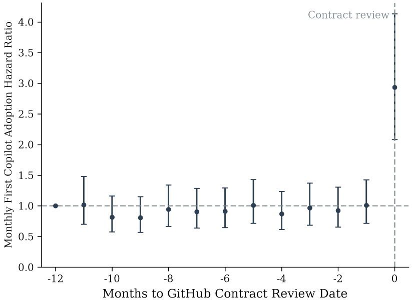
- Contract review raises 12-month Copilot adoption by 19.3 p.p. $(F \approx 28)$
- Leads do not cleanly test anticipation: Copilot launch-window composition effects

Anticipation test

Copilot adoption increases after GitHub Enterprise contract review



(a) GitHub contract review on the Copilot adoption

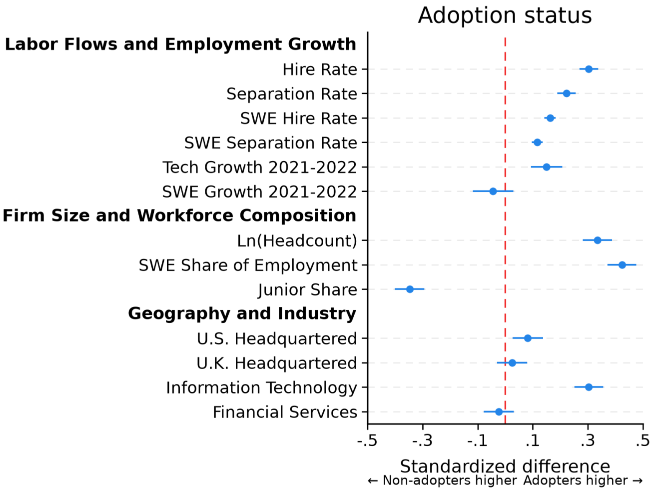


(b) Monthly first-Copilot-adoption hazard

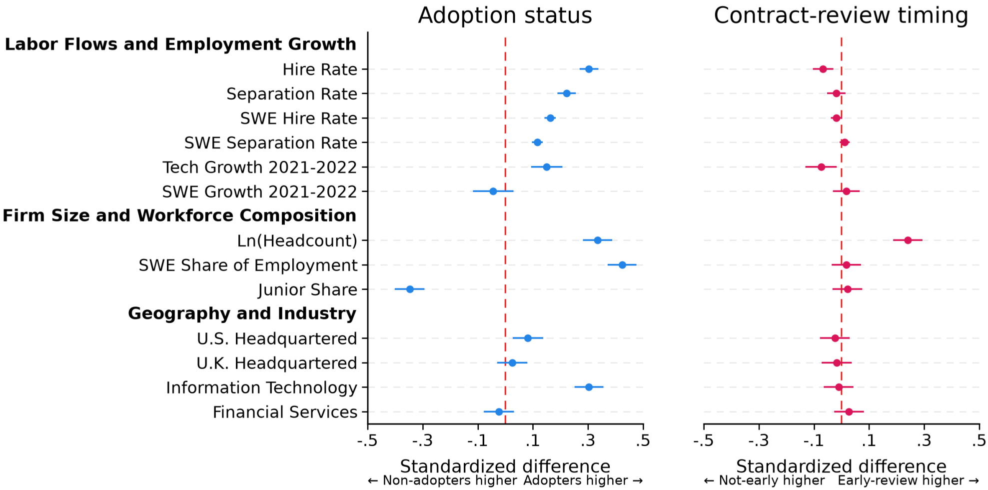
- Contract review raises 12-month Copilot adoption by 19.3 p.p. $(F \approx 28)$
- Leads do not cleanly test anticipation: Copilot launch-window composition effects
- Panel (b) directly tests anticipation: among not-yet-adopters after launch, the monthly first-adoption hazard is flat

Anticipation test

Instrument balance



Instrument balance



Outline

Setting and Treatment: Copilot for Enterprise

Sample and Data

Identification

Results

How Copilot can change demand for technical labor

- **Software production combines complementary tasks across roles**
  - → Technical roles: software, data, systems/infrastructure, cybersecurity, QA, design engineers; app developers

How Copilot can change demand for technical labor

- **Software production combines complementary tasks across roles**
  - → Technical roles: software, data, systems/infrastructure, cybersecurity, QA, design engineers; app developers
- **Copilot adoption is a software production productivity shock**
  - → Increases productivity of code-intensive, execution-heavy work (Cui et al., 2026; Peng et al., 2023)

▸ Framework

How Copilot can change demand for technical labor

- **Software production combines complementary tasks across roles**
  - → Technical roles: software, data, systems/infrastructure, cybersecurity, QA, design engineers; app developers
- **Copilot adoption is a software production productivity shock**
  - → Increases productivity of code-intensive, execution-heavy work (Cui et al., 2026; Peng et al., 2023)
- **The net effect on technical labor demand is ambiguous**
  - → *Productivity:* fewer worker-hours required per unit of software
  - → *Reallocation:* tasks shift across workers, roles, seniority levels
  - → *Scale:* lower production costs increase demand

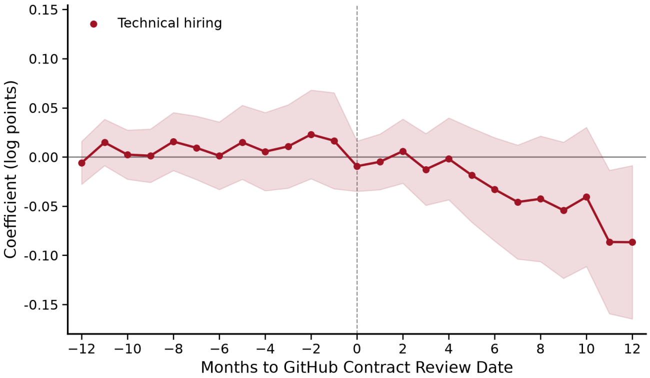
▸ Framework

How Copilot can change demand for technical labor

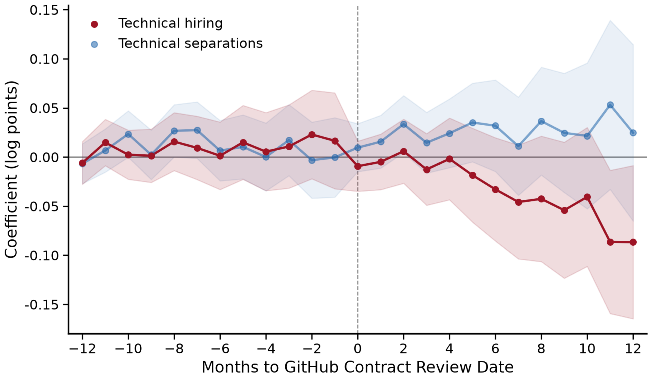
- **Software production combines complementary tasks across roles**
  - → Technical roles: software, data, systems/infrastructure, cybersecurity, QA, design engineers; app developers
- **Copilot adoption is a software production productivity shock**
  - → Increases productivity of code-intensive, execution-heavy work (Cui et al., 2026; Peng et al., 2023)
- **The net effect on technical labor demand is ambiguous**
  - → *Productivity:* fewer worker-hours required per unit of software
  - → *Reallocation:* tasks shift across workers, roles, seniority levels
  - → *Scale:* lower production costs increase demand
- **We measure the net effect**

▸ Framework

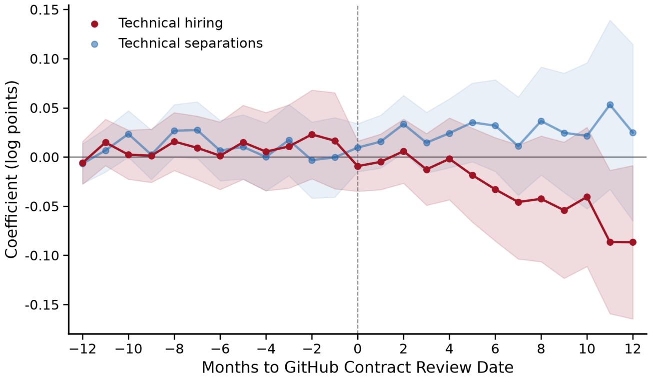
Contract review slows technical hiring



Contract review slows technical hiring

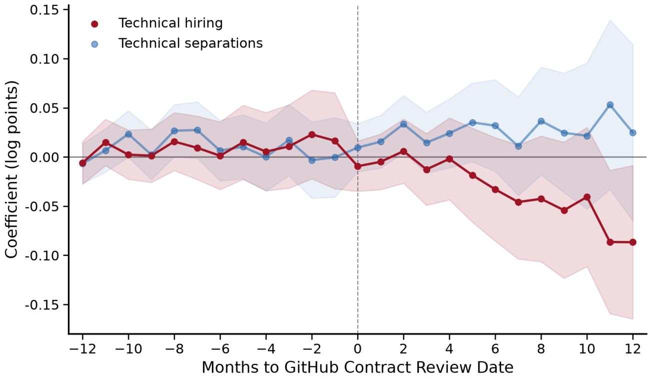


Contract review slows technical hiring



- Foregone hiring, not increased departures
  - $\rightarrow$ Hiring reduced form: $\widehat{\rho}_{12}^{\text{hire}} = -0.087$ (SE $= 0.040$, $p = 0.030$)

Contract review slows technical hiring



- Foregone hiring, not increased departures
  - $\rightarrow$ Hiring reduced form: $\widehat{\rho}_{12}^{\text{hire}} = -0.087$ (SE $= 0.040$, $p = 0.030$)
  - $\rightarrow$ DID-IV estimate $\approx$ -36%, [95% CI: -3%—65%]

Broad hiring slowdown or only in code-intensive roles?

- Hypothesis 1: Broad technical contraction

  $\implies$ Composition of technical hiring is unchanged

Broad hiring slowdown or only in code-intensive roles?

- Hypothesis 1: Broad technical contraction

  $\Longrightarrow$ Composition of technical hiring is unchanged

- Hypothesis 2: Hiring slowdown concentrated in code-intensive roles

  $\Longrightarrow$ Composition of technical hiring shifts towards less code-intensive roles

Broad hiring slowdown or only in code-intensive roles?

- Hypothesis 1: Broad technical contraction

  $\implies$ Composition of technical hiring is unchanged

- Hypothesis 2: Hiring slowdown concentrated in code-intensive roles

  $\implies$ Composition of technical hiring shifts towards less code-intensive roles

- Distinguishing the two hypotheses
  - $\rightarrow$ Software engineering: developing product and application code (more code-intensive)
  - $\rightarrow$ Systems/infrastructure: engineering tech infrastructure (less code-intensive)
  - $\rightarrow$ $\approx \frac{2}{3}$ of technical-related hiring and 91% of firms employ both

Broad hiring slowdown or only in code-intensive roles?

- Hypothesis 1: Broad technical contraction

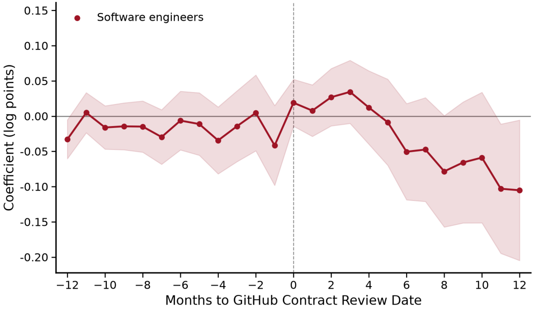
  $\implies$ Composition of technical hiring is unchanged

- Hypothesis 2: Hiring slowdown concentrated in code-intensive roles

  $\implies$ Composition of technical hiring shifts towards less code-intensive roles

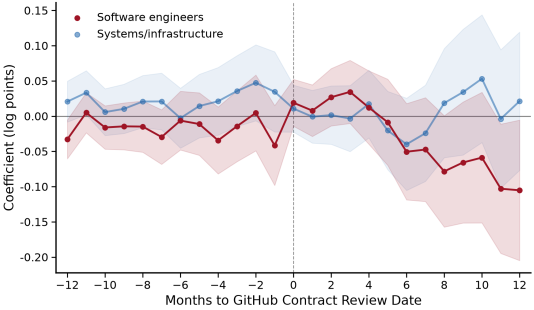
- Distinguishing the two hypotheses
  - $\rightarrow$ Software engineering: developing product and application code (more code-intensive)
  - $\rightarrow$ Systems/infrastructure: engineering tech infrastructure (less code-intensive)
  - $\rightarrow$ $\approx \frac{2}{3}$ of technical-related hiring and 91% of firms employ both
- Testing the hypotheses
  - $\rightarrow$ Role-specific hiring levels
  - $\rightarrow$ Within-firm hiring composition $Y_{it}^{\mathrm{SWE-SI}} = \ln H_{it}^{\mathrm{SWE}} - \ln H_{it}^{\mathrm{SI}} = \ln\left(\frac{H_{it}^{\mathrm{SWE}}}{H_{it}^{\mathrm{SI}}}\right)$

Software-engineering hiring slows; systems/infrastructure does not



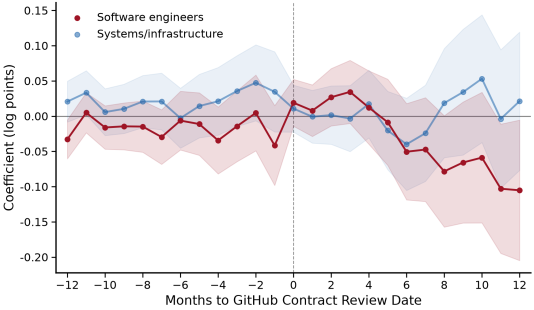
- Slower hiring in software engineering
  $\rightarrow$ $\widehat{\rho}^{\mathrm{SWE}}_{Q4} = -0.093$ (SE $= 0.050$, $p = 0.066$)

Software-engineering hiring slows; systems/infrastructure does not



- Slower hiring in software engineering
  - $\rightarrow$ $\widehat{\rho}^{\mathrm{SWE}}_{Q4} = -0.093$ (SE $= 0.050, p = 0.066$)
  - $\rightarrow$ Q4 DID-IV $\approx -35\%$ [95% CI: -67.0% to -0.4%]
- No detectable response in systems/infrastructure

Software-engineering hiring slows; systems/infrastructure does not



- Slower hiring in software engineering
  - $\rightarrow$ $\widehat{\rho}^{\mathrm{SWE}}_{Q4} = -0.093$ (SE $= 0.050, p = 0.066$)
  - $\rightarrow$ Q4 DID-IV $\approx -35\%$ [95% CI: -67.0% to -0.4%]
- No detectable response in systems/infrastructure
- 1.66:1 to 1.48:1 SWE:Sys/Infra position starts
  - $\rightarrow$ $\widehat{\rho}^{\mathrm{SWE-SI}}_{Q4} = -0.118$ (SE $= 0.073, p = 0.11$)

Measuring coding intensity across technical roles

Test Automation Engineer

SpaceX · Full-time

Jun 2017 - Jan 2019 · 1 yr 8 mos

Hawthorne, CA · On-site

During my time at SpaceX I wrote LabVIEW and Python programs, created data acquisition & control systems, and designed/built liquid nitrogen heat exchangers to enable flight-like valve testing. Hundreds of valves were validated, flying in Falcon boosters and Dragon capsule's manned ISS missions. Some highlights include:

Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity

Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity
- Goal: Construct a fixed, pre-launch measure of the coding intensity for firms and roles in this sample

Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity
- Goal: Construct a fixed, pre-launch measure of the coding intensity for firms and roles in this sample

Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity
- Goal: Construct a fixed, pre-launch measure of the coding intensity for firms and roles in this sample
- Data: 1M pre-period worker position descriptions from LinkedIn, stratified by firm $\times$ role $\times$ seniority

Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity
- Goal: Construct a fixed, pre-launch measure of the coding intensity for firms and roles in this sample
- Data: 1M pre-period worker position descriptions from LinkedIn, stratified by firm $\times$ role $\times$ seniority
- Measurement: code intensity
  $\longrightarrow$ LLM assessment of the share of described work involving writing, testing, documenting code

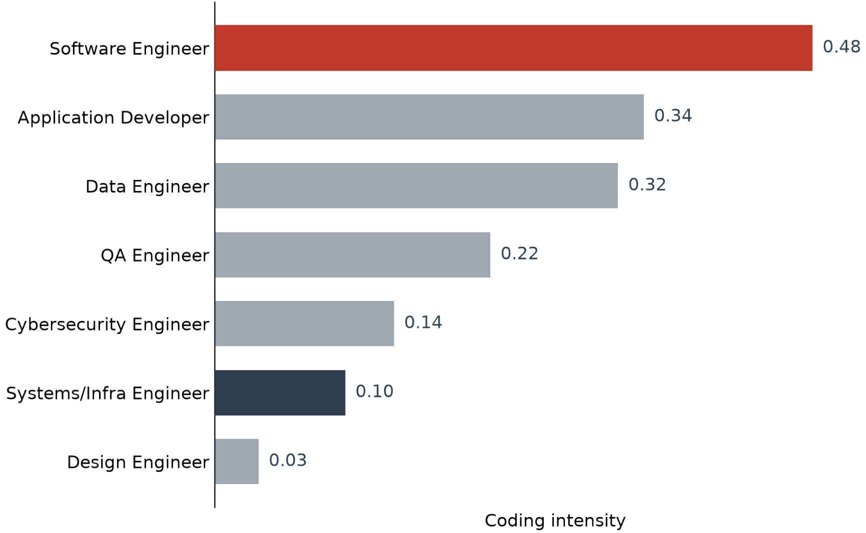
Measuring coding intensity across technical roles

Software Engineer
August Schools · Full-time
Jun 2024 - Mar 2026 · 1 yr 10 mos
New York, New York, United States · Hybrid

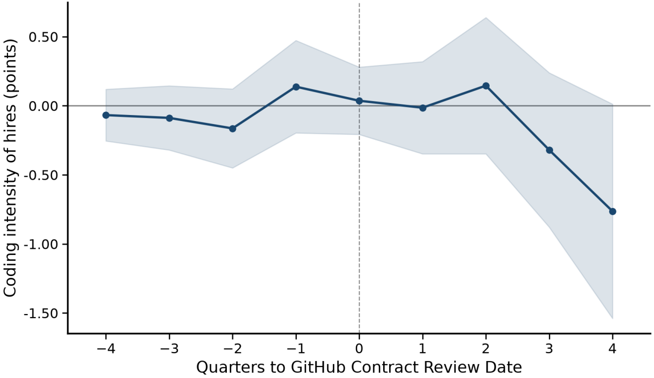
Built a centralized platform to manage deployment, provisioning, and scheduled jobs across our fleet of clients. Used daily by engineering, implementations, and customer support
Designed and implemented a system for managing client deployment health and event logs using Kubernetes and Heroku APIs

- **Observation:** Technical roles differ in code-writing intensity
- Goal: Construct a fixed, pre-launch measure of the coding intensity for firms and roles in this sample
- Data: 1M pre-period worker position descriptions from LinkedIn, stratified by firm $\times$ role $\times$ seniority
- Measurement: code intensity
  - $\rightarrow$ LLM assessment of the share of described work involving writing, testing, documenting code
  - $\rightarrow$ Description text only, not job titles, occupation labels or firm names

Coding intensity varies across roles



Hiring shifts toward less code-intensive roles



- Code intensity of the average new technical hire falls by:
  - $\rightarrow$ $\approx 2.5\%$ in the reduced form
  - $\rightarrow$ $\approx 10.9\%$ decline among compliers

Does junior hiring fall broadly or where work is code intensive?

- Worker seniority important for understanding the incidence of labor market shocks Brynjolfsson et al. (2025), Forsythe (2022), Hosseini and Lichtinger (2025), Hoynes et al. (2012), and Mahajan et al. (2025)

Junior/Senior hiring descriptives

Does junior hiring fall broadly or where work is code intensive?

- Worker seniority important for understanding the incidence of labor market shocks Brynjolfsson et al. (2025), Forsythe (2022), Hosseini and Lichtinger (2025), Hoynes et al. (2012), and Mahajan et al. (2025)
- Two hypotheses:
  1. Juniors are the marginal hire; $\implies$ fall in junior hiring

Junior/Senior hiring descriptives

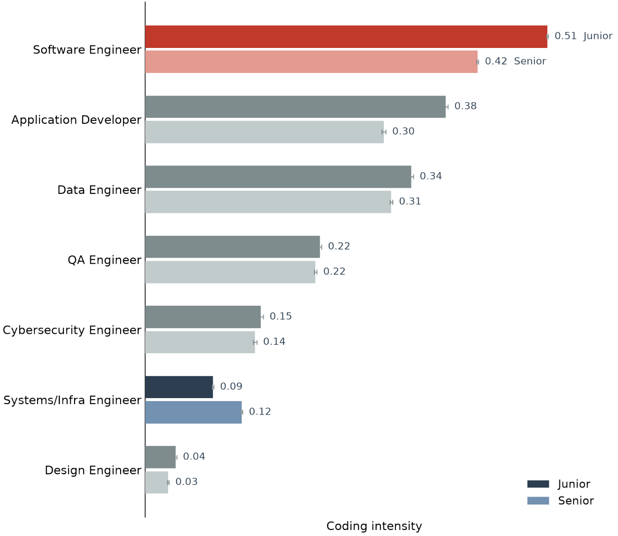
Does junior hiring fall broadly or where work is code intensive?

- Worker seniority important for understanding the incidence of labor market shocks Brynjolfsson et al. (2025), Forsythe (2022), Hosseini and Lichtinger (2025), Hoynes et al. (2012), and Mahajan et al. (2025)
- Two hypotheses:
  1. Juniors are the marginal hire; $\implies$ fall in junior hiring
  2. Junior perform different tasks; $\implies$ correlated with code intensity

- Measuring seniority
  - → Junior: early career and entry level workers
    - Revelio seniority levels 1–2, 5 years of potential experience
  - → Senior: Mid/Senior workers
    - Revelio seniority levels 3–7, 9.5 years of potential experience

Junior/Senior hiring descriptives

Code intensity by role and seniority



Hiring slowdown larger in more code-intensive work

$$\underbrace{\boldsymbol{Y_{irt}}}_{\substack{\ln(\text{hires}) \\ \text{firm } i,\ \text{role } r,\ \text{quarter } t}} = \alpha_{ir} + \lambda_{rt} + \sum_{\ell} (\rho_\ell + \beta_\ell x_r) \mathbf{1}\{t - R_i = \ell\} + u_{irt}$$

Hiring slowdown larger in more code-intensive work

$$Y_{irt} = \alpha_{ir} + \lambda_{rt} + \sum_{\ell} \Big(\rho_\ell + \beta_\ell \underbrace{\boldsymbol{x_r}}_{\text{role's pre-launch coding intensity}} \Big) \mathbf{1}\{t - R_i = \ell\} + u_{irt}$$

- $\boldsymbol{x_r}$ Coding intensity of role r
  $\rightarrow$ One SD $\approx$ QA Engineer $\rightarrow$ Application Developer
- Estimated with Borusyak et al. (2024)

Hiring slowdown larger in more code-intensive work

	ln(hires)
	Occupation
Q4 after review, at mean coding intensity (ρ_4)	-0.019
	(0.035)
Q4 after review $\times$ coding intensity, per SD (β_4)	$\boxed{-0.019}$
	(0.024)
Firm $\times$ cell fixed effects	✓
Cell $\times$ quarter fixed effects	✓
Observations	229,708

Notes: Firm-clustered standard errors in parentheses; coding intensity is standardized.

$^{*}p < 0.10$, $^{**}p < 0.05$.

Hiring slowdown larger in more code-intensive work

Coding intensity and Q4 hiring responses

	ln(hires)	
	Occupation	Occupation × seniority
Q4 after review, at mean intensity (ρ_4)	-0.019	-0.014
	(0.035)	(0.036)
Q4 after review × coding intensity, per SD (β_4)	-0.019	$\boxed{-0.038^{**}}$
	(0.024)	(0.019)
Firm × cell fixed effects	✓	✓
Cell × quarter fixed effects	✓	✓
Observations	229,708	332,646

Notes: Firm-clustered standard errors in parentheses; coding intensity is standardized.

$^{*}p < 0.10$, $^{**}p < 0.05$.

Hiring slowdown larger in more code-intensive work

Coding intensity and Q4 hiring responses

	ln(hires)		
	Occupation	Occupation × seniority	Occupation × seniority + junior interaction
Q4 after review, at mean intensity (ρ_4)	-0.019	-0.014	-0.009
	(0.035)	(0.036)	(0.042)
Q4 after review × coding intensity, per SD (β_4)	-0.019	-0.038^{**}	$\boxed{-0.037^{*}}$
	(0.024)	(0.019)	(0.019)
Q4 after review × junior cell (γ_4)	–	–	-0.012
			(0.040)
Firm × cell fixed effects	✓	✓	✓
Cell × quarter fixed effects	✓	✓	✓
Observations	229,708	332,646	332,646

Notes: Firm-clustered standard errors in parentheses; coding intensity is standardized.

$^{*}p < 0.10$, $^{**}p < 0.05$.

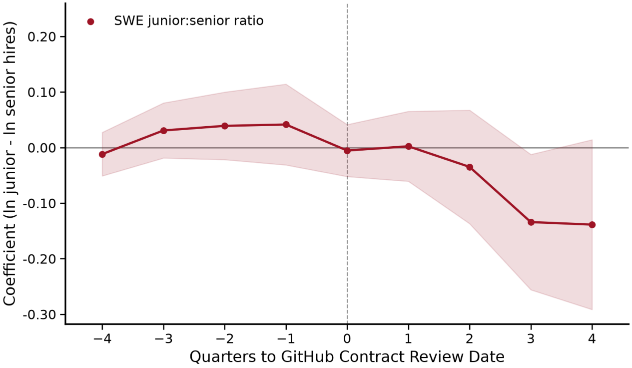
Hiring slowdown larger in more code-intensive work

Coding intensity and Q4 hiring responses

	ln(hires)			ln(jr hires) − ln(sr hires)
	Occupation	Occupation × seniority	Occupation × seniority + junior interaction	Within occupation
Q4 after review, at mean intensity (ρ_4)	-0.019	-0.014	-0.009	-0.025
	(0.035)	(0.036)	(0.042)	(0.049)
Q4 after review × coding intensity, per SD (β_4)	-0.019	-0.038^{**}	-0.037^{*}	$\boxed{-0.085^{**}}$
	(0.024)	(0.019)	(0.019)	(0.036)
Q4 after review × junior cell (γ_4)	–	–	-0.012	–
			(0.040)	
Firm × cell fixed effects	✓	✓	✓	✓
Cell × quarter fixed effects	✓	✓	✓	✓
Observations	229,708	332,646	332,646	119,155

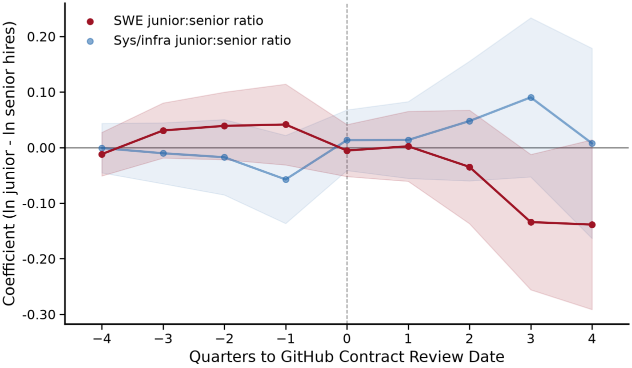
Notes: Firm-clustered standard errors in parentheses; coding intensity is standardized.
$^{*}p < 0.10$, $^{**}p < 0.05$. Column 4 reports the within-firm-occupation junior–senior difference.

Junior hiring falls in SWE, not systems/infrastructure



- Software engineering: $\widehat{\rho}_4^{\text{junior}-\text{senior},\text{SWE}} = -0.139$ (SE $= 0.078$, $p = 0.075$)

Junior hiring falls in SWE, not systems/infrastructure



- Software engineering: $\widehat{\rho}_4^{\text{junior}-\text{senior},\text{SWE}} = -0.139$ (SE $= 0.078$, $p = 0.075$)
  - $\rightarrow$ Junior SWE: $\approx$ 45% decline among complier firms; 95% CI: 1% to 85% decline
- Systems and infrastructure: $\widehat{\rho}_4^{\text{junior}-\text{senior},\text{SI}} = +0.008$ (SE $= 0.087$, $p = 0.93$).
  - $\rightarrow$ Junior Sys/Infra: no detectable effect

Copilot adoption shifts hiring away from code-intensive work

- Among renewal-induced adopters, Copilot adoption leads to:
  - $\rightarrow$ A 36% decline in technical hiring concentrated in code-intensive occupations

Copilot adoption shifts hiring away from code-intensive work

- Among renewal-induced adopters, Copilot adoption leads to:
  - $\rightarrow$ A 36% decline in technical hiring concentrated in code-intensive occupations
- Mechanism: hiring slowing in code-intensive roles $\times$ seniority
  - $\rightarrow$ Net reduction in hiring for code-intensive work

Headcount

Copilot adoption shifts hiring away from code-intensive work

- Among renewal-induced adopters, Copilot adoption leads to:
  - $\rightarrow$ A 36% decline in technical hiring concentrated in code-intensive occupations
- Mechanism: hiring slowing in code-intensive roles $\times$ seniority
  - $\rightarrow$ Net reduction in hiring for code-intensive work
- Reflect reduced hiring, not greater aggregate unemployment
  - $\rightarrow$ Foregone hires directed to other firms or roles
  - $\rightarrow$ Early effects over 2023–2024

Conclusion

Early evidence from enterprise software development

- Natural experiment of GitHub Copilot at enterprise firms
- Incidence falls on foregone hires rather than incumbent workers

Conclusion

Early evidence from enterprise software development

- Natural experiment of GitHub Copilot at enterprise firms
- Incidence falls on foregone hires rather than incumbent workers

Raises further questions:

- What happens to “deflected” hires
- Impacts on smaller firms
- Wages

lraymond@mit.edu

Appendix I

Market definition: BLS SOC codes (OEWS 2025)

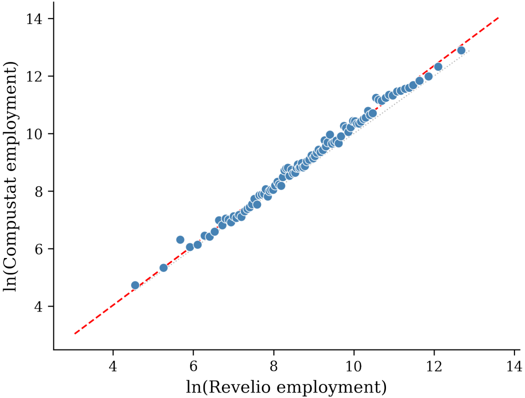
SOC code	Occupation	Employment (2025)
15-1212	Information Security Analysts	190,650
15-1243	Database Architects	67,140
15-1244	Network and Computer Systems Administrators	314,340
15-1251	Computer Programmers	92,230
15-1252	Software Developers	1,687,890
15-1253	Software Quality Assurance Analysts and Testers	186,740
15-1254	Web Developers	70,190
15-1255	Web and Digital Interface Designers	113,330
15-1211	Computer Systems Analysts	519,530
15-1299	Computer Occupations, All Other	435,370
11-3021	Computer and Information Systems Managers	670,570
	Total	**4,347,980**

Note: U.S. national employment from BLS Occupational Employment and Wage Statistics, OEWS 2025. Excludes 15-2051 Data Scientists (262,440), which are also excluded on the revelio side. Sample firms employ ≈1.6M U.S.-based technical workers, implying ≈51% coverage of the 3.16M-worker market.

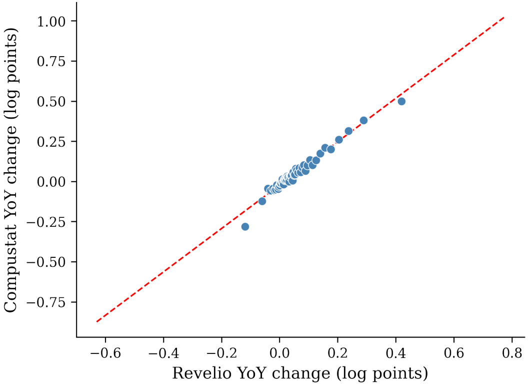
Sample Summary Statistics (September 2022)

	Full sample	Adopters	Non-adopters
Employment (Revelio)	8,292	10,508	4,164
	(28,502)	(34,075)	(11,698)
Firm Age	47	45	50
	(45)	(45)	(46)
SE share of employment (%)	5.7	6.8	3.6
	(7.1)	(7.2)	(6.4)
Technical share of employment (%)	12.6	14.1	9.9
	(11.2)	(11.1)	(10.8)
Junior share of employment (%)	59.6	57.9	63.0
	(14.5)	(14.4)	(14.1)
Domestic share of employment (%)	72.0	69.6	76.5
	(25.7)	(25.6)	(25.3)
Hiring rate, monthly avg. (%)	2.46	2.64	2.13
	(1.29)	(1.33)	(1.14)
Separation rate, monthly avg. (%)	1.80	1.88	1.64
	(0.73)	(0.72)	(0.72)
Net flow rate, monthly avg. (%)	0.66	0.76	0.49
	(0.83)	(0.87)	(0.69)
Headcount growth, 2021–2022 (%)	8.4	10.3	5.0
	(14.8)	(15.5)	(12.8)
SE headcount growth, 2021–2022 (%)	13.8	15.6	10.3
	(34.0)	(30.8)	(39.4)
Avg. renewal cycle (months)	12.1	11.9	12.3
	(2.0)	(1.4)	(2.4)
Multi-country workforce (%)	99.5%	99.7%	99.2%
U.S. headquartered (%)	61.6%	63.8%	57.4%
U.K. headquartered (%)	7.4%	7.7%	6.9%
Canada headquartered (%)	5.9%	4.8%	7.8%
Information Technology (%)	24.7%	29.4%	16.0%
Financial Services (%)	12.9%	12.4%	13.9%
N firms	5,425	3,530	1,895

Revelio-Compustat Data Validation



(a) 10k Employment



(b) Year over Year Changes

OLS Estimates

	Headcount	Technical	SWE	Hiring	Separations
Firm + month FE	0.011***	0.020***	0.018**	-0.002	0.016**
	(0.002)	(0.004)	(0.007)	(0.007)	(0.007)
+ industry × month FE	0.017***	0.029***	0.028***	0.016**	0.031***
	(0.003)	(0.004)	(0.007)	(0.008)	(0.007)
+ size × month FE	0.015***	0.023***	0.022***	0.018**	0.033***
	(0.003)	(0.004)	(0.006)	(0.008)	(0.007)

Back to first stage

Placebo occupations: no hiring response at 12 months

- Exposed to generative AI broadly, but not to Copilot

Occupation	Margin	ρ_4	Pre-trend p
Financial + investment analysts	Hiring (ln, intensive)	0.003 (0.043)	0.68
Accountants	Hiring (ln, intensive)	0.034 (0.043)	0.04
Accountants	Any hiring (extensive)	$-$0.000 (0.023)	0.70

Note: Reduced-form event-study coefficients at $\ell = 4$ quarters after the first post-launch renewal (Borusyak–Jaravel–Spiess imputation. Intensive margin is $\ln($hires$)$ on baseline-positive firms; extensive is $\mathbf{1}\{$hires $> 0\}$. Pre-trend p is the joint test on the four lead coefficients..

Back

Same Wald-DiD estimator, two different estimands

Both designs estimate the ratio $\theta_\ell = \rho_\ell / \pi_\ell$; they differ in the population for which the ratio is causal

Staggered DID-IV

- LATET for **instrument compliers**: $A_{i1}(1) > A_{i1}(0)$
- Firms whose contract review induces them to adopt Copilot earlier
- Allows heterogeneous effects

Miyaji (2024)

Fuzzy DiD

- ATT for **switchers**: $A_{i0} = 0$, $A_{i1} = 1$, induced by review or not
- Adoption also rises among controls $\Rightarrow$ requires effect homogeneity across groups and over time

Chaisemartin and D'Haultfœuille (2018)

- Both require no carryover: Y_{it} depends only on current adoption status A_{it}, not time since adoption
- Not-yet-reviewed firms also adopt (always-takers) $\Rightarrow$ switchers $\neq$ compliers; we report the LATET

Back to design

Placebo occupations: no hiring response at 12 months

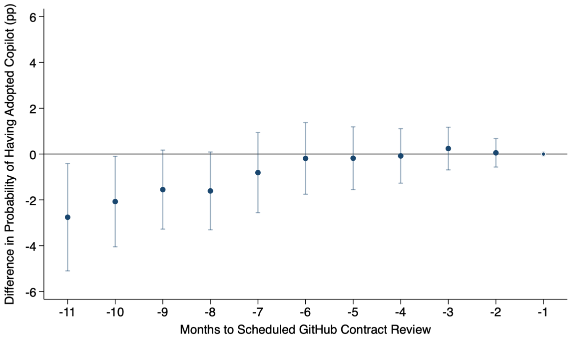
- Exposed to generative AI broadly, but not to Copilot

Occupation	Margin	ρ_4	Pre-trend p
Financial + investment analysts	Hiring (ln, intensive)	0.003 (0.043)	0.68
Accountants	Hiring (ln, intensive)	0.034 (0.043)	0.04
Accountants	Any hiring (extensive)	$-$0.000 (0.023)	0.70

Note: Reduced-form event-study coefficients at $\ell = 4$ quarters after the first post-launch renewal (Borusyak–Jaravel–Spiess imputation. Intensive margin is $\ln($hires$)$ on baseline-positive firms; extensive is $\mathbf{1}\{$hires $> 0\}$. Pre-trend p is the joint test on the four lead coefficients..

Back to ID Strategy

Post-launch test for anticipatory adoption



- Sample: post-Copilot-launch months strictly before each firm's scheduled contract review
- Coefficients compare the probability of having adopted to the month prior to contract review $t = -1$
- Calendar-month fixed effects absorb common diffusion, standard errors clustered at the firm level
- No firm fixed effects, since $\text{MonthsToReview}_{it} = R_i - t$

Back to first stage

Firm labor demand reflects scale, reallocation, and productivity

- Firms produce software output Q_i by combining tasks $j \in J$.
- Producing one unit of output requires α_j units of task j
- Labor used in role–seniority cell (r, s) at firm i is:

$$L_{i,rs} = \sum_j \frac{s_{ij,rs}\, \alpha_j\, Q_i}{\theta_{ij,rs}}.$$

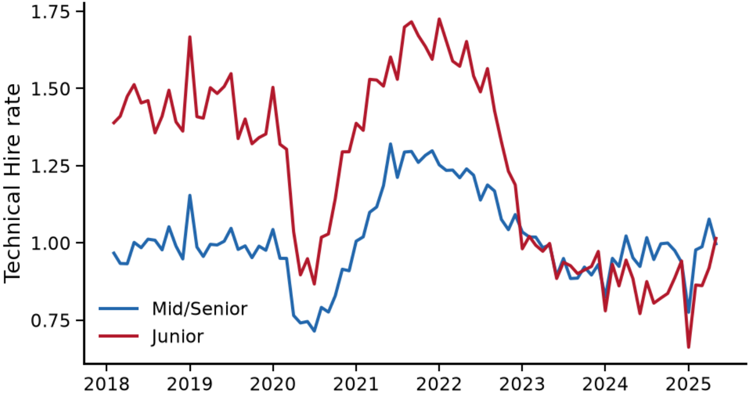
  $\rightarrow$ $s_{ij,rg}$ is the share of task j assigned to cell (r, s); $\theta_{ij,rg}$ is productivity; Copilot is a task specific productivity shock

- Log-differentiating and aggregating across tasks with $\phi_{ij,rs} \equiv L_{ij,rs}/L_{i,rs}$ gives:

$$d\ln L_{i,rs} = \underbrace{d\ln Q_i}_{\text{scale}} + \underbrace{\sum_j \phi_{ij,rs}\, d\ln s_{ij,rs}}_{\text{reallocation}} - \underbrace{\sum_j \phi_{ij,rs}\, d\ln \theta_{ij,rs}}_{\text{productivity}}.$$

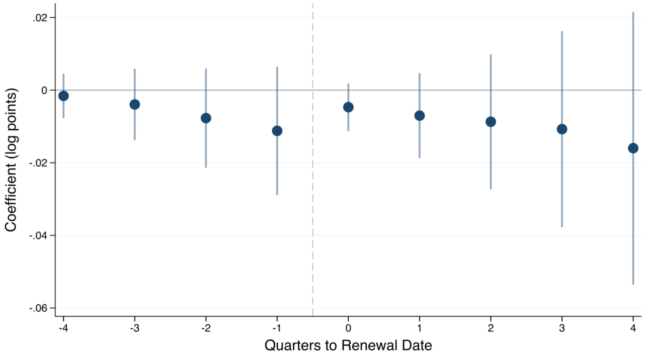
- Our estimates: the net effect of three forces on labor demand at cell (r, s)

Junior hiring more procyclical than senior



Back

No detectable effect on technical headcount after 12 months



Back to end of results