# Granular knowledge spillovers: Evidence from software developers

Aaron Lohmann

University Bielefeld and IfW Kiel

**Motivation: Part 1**

A collaborative community:

- Innovative industries tend to be commnunity efforts.
- Raises questions about **peer influence**.

# Motivation

**Motivation: Part 1**

A collaborative community:

- Innovative industries tend to be commnunity efforts.
- Raises questions about **peer influence**.

**Motivation: Part 2**

Modes of innovation:

- **Exploration**: adoption of novel, radical technologies.
- **Exploitation**: creative recombination of knowledge.

# Motivation

**Motivation: Part 1**

A collaborative community:

- Innovative industries tend to be commnunity efforts.
- Raises questions about **peer influence**.

**Motivation: Part 2**

Modes of innovation:

- **Exploration**: adoption of novel, radical technologies.
- **Exploitation**: creative recombination of knowledge.

*Research Questions*

1. How do peers influence developers' technological choices?
2. Does the peer-network size shift the balance between exploration and exploitation? If so, in what direction?

# This Paper

1. **Theory**: Develops a simple model of technology choice under peer influence.
2. **Data**: Uses novel, fine-grained micro-data from the Rust OSS community.
3. **Evidence**:
   - Developers are more likely to adopt technologies their peers used — especially true for **young and inexperienced** developers.
   - Technological choices are highly **persistent** (path dependence).
   - Larger peer networks $\rightarrow$ **more exploitation**, **less exploration**.

# Literature

- **Growth** — Lucas (2009) , R. E. Lucas and Moll (2014) , Akcigit et al. (2018) , Jarosch, Oberfield, and Rossi-Hansberg (2021) , Herkenhoff et al. (2024)
  ***Contribution****: Bring new micro evidence from software development to inform these theoretical mechanisms.*

- **Diffusion and type of innovation** — Schumpeter (1942), Weitzmann (1993), Sandvik et al. (2020), Atkin et al. (2022), Uzzi et al. (2013), Berkes and Gaetani (2022)
  ***Contribution:*** *Classify innovation types directly from technological choices. Empirical evidence on exploration vs. exploitation.*

- **Open Source & Software** — Schueller et al. (2022) , Wachs et al. (2025)
  ***Contribution:*** *Propose classification of software projects via software dependencies. Extend empirical evidence on peer effects.*

# Model Setup

- **Developers**

  Each developer has a core expertise (e.g., front-end $A$ or back-end $B$) and an exogenous peer network of past collaborators. Let the peers of developer $i$ be called $P_i$.

- **Technologies**

  - Developers choose among technologies $j \in \{A, B, C\}$ to combine with own core expertise.
  - Payoff from a technology depends on the *best blueprint* a developer can access:

$$z_{ij} = \max_{d \leq D_{ij}} T_{ijd}, \quad D_{ij} = 1 + P_{ij} + \phi P_i$$

    where $P_{ij}$ is the number of peers with expertise $j$ and $P_i$ is the total amount of peers.

# Choice problem

- **Developer's Choice Problem**

  Output combines core technology and a secondary field:

$$\max_k y_{ik} = (z_{i,E_i})^{\eta}(z_{ik})^{1-\eta}$$

# Predictions

- **Testable Predictions**
  1. **Peer Technology Adoption**
     More peers in technology $j$ $\Rightarrow$ higher probability of adopting $j$.
  2. **Recombination (Exploitation)**
     Larger peer networks $\Rightarrow$ more recombination of established technologies.
  3. **Novel technologies (Exploration)**
     Theoretically ambigious: depends on $\phi$ i.e. the strength of global vs. technology specific spillovers.

# Context: Open Source Software and Rust

- **Open Source Software (OSS)** is the **backbone of digital infrastructure**.
  - 90% of codebases rely on OSS components. (OSSRA, 2024)
  - Estimated **demand-side value: USD 9 trillion**. Hoffmann, Nagle, and Zhou (2024)
  - Potentially more relevant for practice than patents (Mannheim Innovation Panel, 2024).
- **Rust community**
  - Modern systems programming language.
  - Popular, young language, first version in 2014 $\rightarrow$ Full history available.
  - Small standard library $\rightarrow$ strong reliance on external packages ("crates", dependencies).

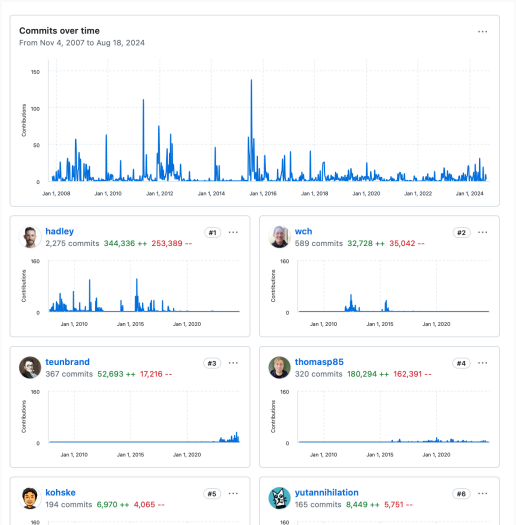
# Data.

Several data sources, main ones are:

- Schueller et al. (2022):
  - Developer and dependencey network.
- GHtorrent:
  - Additional developer specific information.
- Libs.rs:
  - Categorization of packages.

# Data continued.

**Developers jointly work on projects**




**and use inputs.**

```
34  Imports:
35      cli,
36      grDevices,
37      grid,
38      gtable (>= 0.3.6),
39      isoband,
40      lifecycle (> 1.0.1),
41      rlang (>= 1.1.0),
42      scales (>= 1.4.0),
43      stats,
44      vctrs (>= 0.6.0),
45      withr (>= 2.5.0)
```

**Figure 1:** ggplot2 imported projects

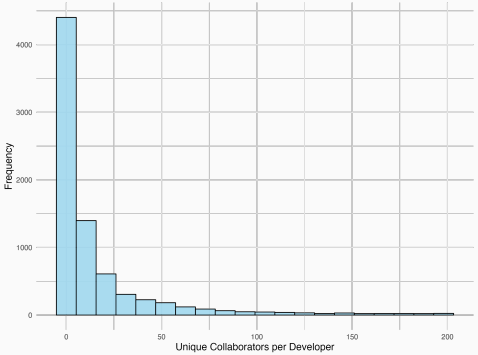
# Basics I.

**Table 1:** Basics

| **Variable** | **Value** |
|---|---|
| Time Frame | 2014 - 2022, September (Quarterly) |
| Developers | $46,265$, project owners: $16,335$ |
| Projects | $23,437$ packages. |
| Inputs | $2,072$ unique inputs (passing a threshold of relevance). |
| Categories | $71$ |

*Notes:* This table summarises some main features of the data. Data by Schueller et al. (2022). Authors own calculations.

# Basics II.



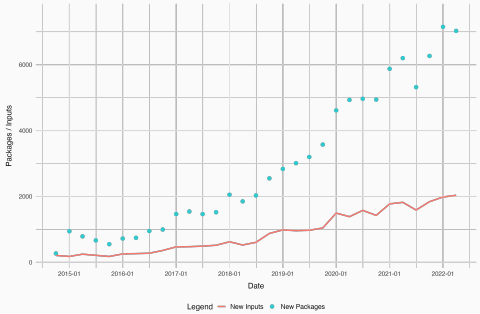




**Figure 2:** Left: Cross citation patterns. Right: Inputs over time.

*Notes:* Left: Unique collaborators per developer. Right Inputs over time.

# Testing prediction I: Knowledge diffusion.

$$D_{ijpt} = \beta_1 \log PE_{ij,t-k} + \beta_2 \log SE_{ij,t-k} + \alpha_{c(p)c(j)} + \gamma_{its} + \mu_{jt}$$

where $D_{ijpt}$ captures whether developer $i$ uses input $j$ in project $p$ at time period $t$.

# Testing prediction I: Knowledge diffusion.

$$D_{ijpt} = \beta_1 \log PE_{ij,t-k} + \beta_2 \log SE_{ij,t-k} + \alpha_{c(p)c(j)} + \gamma_{its} + \mu_{jt}$$

where $D_{ijpt}$ captures whether developer $i$ uses input $j$ in project $p$ at time period $t$.

- $PE_{ij,t-k}$ is developer $i$, time specific exposure to technology $j$ via peers of $i$. $k$ is chosen and varied for robustness (base: one year).

# Testing prediction I: Knowledge diffusion.

$$D_{ijpt} = \beta_1 \log PE_{ij,t-k} + \beta_2 \log SE_{ij,t-k} + \alpha_{c(p)c(j)} + \gamma_{its} + \mu_{jt}$$

where $D_{ijpt}$ captures whether developer $i$ uses input $j$ in project $p$ at time period $t$.

- $PE_{ij,t-k}$ is developer $i$, time specific exposure to technology $j$ via peers of $i$. $k$ is chosen and varied for robustness (base: one year).
- $SE_{ijt}$ captures how often $i$ has used $j$ in the past.
- FE effects are at:
  - Category-Category level.
  - Developer-Quarter.
  - Dependency-Quarter.
- Endogeneity: Adressed via traditioal peer effects identification 'Friends-of-Friends' as in Bramoullé, Djebbari, and Fortin (2009).

# Results prediction I.

**Table 2:** OLS Regression Results

| Dependent Variable: | Uses Dependency | |
|---|---|---|
| | (1) | (2) |
| Log(Peer exposure + 1) | 0.0117*** | |
| | (0.0010) | |
| Log(Own exposure + 1) | 0.0975*** | |
| | (0.0070) | |
| Arcsinh(Peer exposure ) | | 0.0091*** |
| | | (0.0007) |
| Arcsinh(Own exposure) | | 0.0762*** |
| | | (0.0054) |
| Observations | 30,065,237 | 30,065,237 |
| $R^2$ | 0.03579 | 0.03585 |
| Within $R^2$ | 0.01249 | 0.01255 |
| Developer-Quarter FE | ✓ | ✓ |
| Proj.-Cat.-Dep.-Cat. FE | ✓ | ✓ |
| Dependency-Quarter FE | ✓ | ✓ |

# Testing predictions II: Classification of project.

Recombination (adapted from Uzzi et al. (2013) and Berkes and Gaetani (2022)):

1. Let a project use technologies $A$, $B$ and $C$.
2. Record all pairwise tuples i.e. $(A, B), (A, C), (B, C)$
3. Compare tuples across projects, many novel tuples are associated with novelty via recombination.

Exploration

1. Note category of project $p$, denoted $c$.
2. Record all inputs project $p$ uses, let it be $A$ and $D$
3. Check how often project of category $c$ uses inputs $A$ or $D$.
4. Many rare inputs conditional on category, label it exploration.

# Testing prediction II: Regression.

$$Y_{ijtm} = \beta_1 \log(\text{Past Coworkers}i, t+1) + \beta_2 \log(\text{Experience}i, t+1) + \tau_t + \varepsilon_{ijtm}$$

,

where $Y$ may be represent one of four different innovation modes:

1. Any Exploit: The project recombines in an unconventional way.
2. Any Explore: The project employs novel inputs.
3. Only Exploit: The project **only** recombines in unconventional ways without exploration channel.
4. Only Explore: Project uses mostly novel inputs, the combination of them are however not uncommon.

# Results prediction II.

**Table 3:** Innovation mode regressions (baseline) – Input measure

| Dependent Variable: | Any Exploit (1) | Any Explore (2) | Only Exploit (3) | Only Explore (4) |
|---|---|---|---|---|
| Log(Past Coworkers+ 1) | 0.0092 | -0.0055** | 0.0117* | -0.0031* |
| | (0.0063) | (0.0023) | (0.0059) | (0.0015) |
| Log(Dev. Experience + 1) | -0.0181** | -0.0008 | -0.0185** | -0.0011 |
| | (0.0070) | (0.0028) | (0.0069) | (0.0027) |
| Observations | 23,282 | 23,282 | 23,282 | 23,282 |
| $R^2$ | 0.00591 | 0.05345 | 0.00439 | 0.03869 |
| Within $R^2$ | 0.00096 | 0.00045 | 0.00116 | 0.00018 |
| Quarter FE | ✓ | ✓ | ✓ | ✓ |

# Conclusion

- **Collaborative industries matter**
  Developers learn about concrete inputs from peers.
- **Persistence**
  Technological choices are highly path dependent.
- **Peer network size**
  Larger networks $\rightarrow$ more **exploitation**, less **exploration**.
  - Consistent with Field-specific spillovers dominating global spillovers.

# Heterogeneity

| Dependent Variable: | Uses Dependency | |
|---|---|---|
| | (1) | (2) |
| Log(Peers Value + 1) | 0.0161*** | |
| | (0.0018) | |
| Log(Self Value + 1) | 0.0975*** | |
| | (0.0070) | |
| Log(Peers Value + 1) × Developer Age | -0.0005*** | |
| | (0.0001) | |
| Arcsinh(Peers Value) | | 0.0139*** |
| | | (0.0012) |
| Arcsinh(Self Value) | | 0.0765*** |
| | | (0.0054) |
| Arcsinh(Peers Value) × Past Projects | | -0.0002*** |
| | | $(2.6 \times 10^{-5})$ |
| Observations | 30,065,237 | 30,065,237 |
| $R^2$ | 0.03584 | 0.03598 |
| Within $R^2$ | 0.01253 | 0.01268 |

Standard errors clustered at the user level. Instrumented variable: Peers Value.

# Categories

| Category | Category.cont. |
|---|---|
| text-processing | development-tools::ffi |
| concurrency | memory-management |
| network-programming | web-programming |
| algorithms | config |
| cryptography | compression |
| asynchronous | internationalization |
| accessibility | science::math |
| rendering::graphics-api | simulation |
| parser-implementations | games |
| development-tools::procedural-macro-helpers | hardware-support |
| game-development | text-editors |
| compilers | database |
| embedded | os |

# IV results for diffusion

**Table 4:** Second Stage Regression Results

| Dependent Variable: | Uses Dependency | |
|---|---|---|
| | (1) | (2) |
| Log(Past peer exposure + 1) (fitted) | 0.0075** | |
| | (0.0036) | |
| Log(Past own exposure + 1) | 0.0941*** | |
| | (0.0070) | |
| Arcsinh(Past peer exposure) (fitted) | | 0.0026 |
| | | (0.0023) |
| Arcsinh(Past own exposure) | | 0.0772*** |
| | | (0.0053) |
| Observations | 29,059,168 | 30,065,237 |
| $R^2$ | 0.03328 | 0.03554 |
| Within $R^2$ | 0.01180 | 0.01223 |
| Developer-Quarter FE | ✓ | ✓ |
| Proj.-Cat.-Dep.-Cat. FE | ✓ | ✓ |
| Dependency-Quarter FE | ✓ | ✓ |

# Construction Peer Exposure

$$PE_{ijt} = M_{ii't} U_{i'jt}$$

where $M_{ii't}$ is a matching matrix with binary elements and $U_{i'jt}$ captures whether developer $i'$ has used input $j$ up until time point $t$.

## References


Akcigit, Ufuk, Santiago Caicedo, Ernest Miguelez, Stefanie Stantcheva, and Valerio Sterzi. 2018. "Dancing with the Stars: Innovation Through Interactions." Cambridge, MA. https://doi.org/10.3386/w24466.

Bramoullé, Yann, Habiba Djebbari, and Bernard Fortin. 2009. "Identification of Peer Effects Through Social Networks." *Journal of Econometrics* 150 (1): 41–55.

Herkenhoff, Kyle, Jeremy Lise, Guido Menzio, and Gordon M. Phillips. 2024. "Production and Learning in Teams." *Econometrica* 92 (2): 467–504. https://doi.org/10.3982/ECTA16748.

Hoffmann, Manuel, Frank Nagle, and Yanuo Zhou. 2024. "The Value of Open Source Software," January. https://doi.org/10.2139/ssrn.4693148.

Jarosch, Gregor, Ezra Oberfield, and Esteban Rossi-Hansberg. 2021. "Learning From Coworkers." *Econometrica* 89 (2): 647–76. https://doi.org/10.3982/ECTA16915.