Congestion in Onboarding Workers and Sticky R&D

Justin Bloesch* Jacob P. Weber†

May 31, 2023

Abstract

R&D investment spending exhibits a delayed and hump-shaped response to shocks. We show in a simple partial equilibrium model that rapidly adjusting R&D investment is costly if the probability of converting new hires into productive R&D workers ("onboarding") is decreasing in the number of new hires ("congestion"). Congestion thus causes R&D producing firms to slowly hire new workers in response to good shocks and hoard workers in response to bad shocks, providing a microfoundation for convex adjustment costs in R&D investment. Using novel, high-frequency productivity data on individual software developers collected from GitHub, a popular online collaboration platform, we provide quantitative evidence for such congestion. Calibrated to this evidence, a sticky-wage new Keynesian model with heterogeneous investment-producing firms subject to congestion in onboarding and no other frictions yields hump-shaped responses of R&D investment to monetary policy shocks.

*School of Industrial and Labor Relations, Cornell University. Email: jb2722@cornell.edu
†Department of Economics, UC Berkeley. Email: jacob_weber@berkeley.edu

We thank Bronwyn Hall, Chen Lian, Emi Nakamura, Christina Romer, David Romer, Benjamin Schoefer, Jón Steinsson, and seminar participants at Baruch College, Columbia University, the Federal Reserve Bank of San Francisco, the Federal Reserve Bank of New York, the Federal Reserve Board of Governors, Haverford College, UC Berkeley, UC Santa Cruz, and Williams College for valuable comments and discussion. Jacob Weber gratefully acknowledges hospitality and financial support received from the Federal Reserve Bank of San Francisco while working on this project, which was a chapter of his dissertation. We also thank Leah Attai, Sean Bae, Joy Chen, Daniel Hoffman, Gaurav Manek, Aaron Stern and others with professional software development experience who generously provided advice.

JEL Codes: E22, O36.
Keywords: Intangibles, Monetary Policy, R&D, Innovation, Team Specific Capital, Labor Adjustment Costs.

1 Introduction

R&D investment, like other kinds of investment, is "sticky": the rate of investment spending is persistent both at the firm level and in the aggregate in response to shocks. To generate this result, a growing literature on intangible investment models R&D spending as subject to convex adjustment costs to the rate of investment spending.[1] More generally, mainstream macro models need these specific adjustment costs to capture the delayed and hump-shaped response of investment to monetary policy shocks. While helpful to fit the data in each case, this critical friction is *ad hoc*, meaning there are few explanations for its source.[2] Further, no proposed explanation focuses on R&D and other "Intellectual Property Products" (IPP) investment, which has grown steadily in importance and is now the single largest component of U.S. fixed investment.[3]

This paper provides an explanation for convex costs to adjusting the rate of R&D and other IPP investment. First, we show in a simple partial equilibrium model how such costs can arise from congestion in onboarding new workers. By onboarding, we mean that new, "junior" workers acquire firm- or project-specific skills on the job in order to transition to becoming productive "senior" workers. Since scarce attention and supervision from existing seniors is necessary for this transition, hiring many juniors at once decreases the probability that juniors successfully transition: a property we call congestion in onboarding. Firms subject to congestion in onboarding optimally hire new junior workers slowly in response to good shocks and hoard senior workers in response to bad shocks. Provided that the shocks affecting the firm are not too big, we show analytically that our congestion model is identical to a model of convex investment adjustment costs, thus providing a microfoundation for them.[4]

Next, we estimate the degree of congestion in onboarding for an important subset of these workers who produce IPP: software developers, who produce about 1/3 of all R&D invest-

[1] See e.g. Moran and Queralto (2018); Bianchi et al. (2019); Cloyne et al. (2022).

[2] See Christiano et al. (2005) and Smets and Wouters (2007) for this friction's importance; Christiano et al. (2018) review proposed foundations for these adjustment costs, which are distinct from intuitive features like convex capital installation costs (Hayashi, 1982), fixed adjustment costs, irreversible investment, etc.

[3] NIPA Table 1.1.5, years 2020 and 2021. Appendix A details the secular trend and components of IPP.

[4] Our use of labor adjustment costs to explain investment adjustment costs reflects the labor intensive nature of R&D, which requires specialized, project-specific knowledge to produce (Hall and Lerner, 2010).

ment and the majority of IPP investment.[5] We use data on individual software developers collected from GitHub, a popular online collaboration platform boasting over 80 million users across 4 million organizations as of 2022.[6] GitHub tracks the contributions of each user on software projects, documenting who authored each change to the code, allowing us to follow software developers and track their productivity over time on public, open source software projects. The data available from GitHub's Application Programming Interface (API) is on a terabyte scale. Rather than collect this data ourselves, we turn to the GHTorrent project (Gousios, 2013), long-used by software developers to study the productivity of other software developers.[7] Using this dataset, we find substantial congestion: when a project has many juniors joining at the same time, the probability that an individual junior successfully onboards and becomes a productive senior team member declines. The nature of the production process and narrative evidence suggest this stems from the fact that successful onboarding requires attention and supervision from senior workers while the junior worker acquires the project-specific knowledge necessary to contribute, as in our model.

Finally, we embed congestion in onboarding R&D workers in an otherwise standard new Keynesian model where R&D investment is produced by heterogenous firms facing large idiosyncratic productivity shocks. This allows us to consider the effects of monetary policy shocks in general equilibrium while relaxing the "small shocks" assumption made earlier for analytical tractability. We solve for the model's response to monetary policy shocks using sequence space methods (Auclert et al., 2021) and show that our calibrated onboarding frictions generate realistic, hump-shaped impulse responses.

This analysis supports a long-conjectured explanation for the observed stickiness in the empirical literature on R&D: that for firms engaged in knowledge production, substantial firm-specific human capital is bound up in the minds of workers and lost when workers leave. Firms thus behave "as if" they face steep costs of adjusting labor inputs (Hall and Lerner, 2010; Kerr and Nanda, 2015). Consistent with this, recent empirical work establishes an

[5] In the NIPAs Software is included in IPP both as R&D and in other subcategories. See Appendix A.

[6] See `https://github.com/about` (accessed 10/24/22).

[7] As this public dataset has been largely overlooked by researchers in economics, and may be unfamiliar to many readers, Section 3.1 and Appendix D provide a thorough description with citations to more technical discussions published by software developers, which we hope will encourage researchers without a background in software development to work with this data.

important role for team- or firm-specific capital in knowledge creation (Jaravel et al., 2018; Kline et al., 2019). This explanation implicitly assumes that such firm-specific knowledge is difficult to transmit to newcomers – a property we establish as quantitatively relevant for an important subset of R&D workers. Thus when firms produce an investment good primarily with labor, as with R&D production, the steep costs of adjusting the stock of labor producing that investment good is a source of costly adjustment in the flow output of investment goods.[8]

Our empirical results provide a foundation specifically for convex costs to adjusting the rate of investment, which aggregate DSGE models incorporate *ad hoc* to capture the response of investment to monetary policy shocks (Christiano et al., 2005; Smets and Wouters, 2007). This friction is critical, and Smets and Wouters (2007) refer to it as the single most important real friction in improving model fit.[9] The secular rise of R&D and other IPP investment has not reduced the importance of these adjustment costs, as such intangible investment is if anything stickier than traditional tangible investment (equipment and structures): models fitting data for tangible and intangible investment separately find a much larger role for convex adjustment costs on intangible investment.[10] By providing an explanation for why R&D and other IPP investment is costly to adjust, we directly inform models of capital accumulation applied to such intangible investment, and for simpler aggregate models with only one type of investment spending, provide justification for the practice of retaining traditional frictions even as the nature of investment changes.[11] Specifically, we provide evidence that such adjustment costs for R&D and IPP production are "deep" features of the production process invariant to changes in government policy, which is an implicit assumption

[8]Estimating the shape and source of labor adjustment costs is a long-standing goal in labor economics, particularly for skilled workers (Oi (1962), Hamermesh and Pfann (1996)). We provide direct evidence for one form of labor adjustment costs in the context of R&D workers and elaborate conditions under which it is isomorphic to convex adjustment costs.

[9]Justiniano et al. (2010) argue that this stems from an overly smooth investment concept (excluding e.g. inventories) but continue to emphasize the critical role of investment in business cycle dynamics.

[10]Moran and Queralto (2018), Bianchi et al. (2019), and Cloyne et al. (2022) fit models to aggregate R&D, estimating much higher investment adjustment costs than for tangible investment (seven, four, and over twenty times as large, respectively). At the firm level, Peters and Taylor (2017) also estimate higher adjustment costs for intangible investment.

[11]Another important set of explanations includes Casares (2006), Edge (2007) and Lucca (2007) who illustrate how extensions of the "Time to Build" formulation of Kydland and Prescott (1982) can yield hump-shaped investment responses or are equivalent to convex adjustment costs.

whenever using models with *ad hoc* adjustment costs to conduct any sort of counterfactual exercise or welfare analysis.

The paper proceeds as follows: Section 2 describes the problem of a firm producing a labor-intensive investment good (R&D or other IPP investment) subject to congestion in onboarding in partial equilibrium. We show that in a special case where shocks are sufficiently small, the model is identical to a model of investment adjustment costs. When shocks are large, we work through a numerical example in partial equilibrium to demonstrate that the firm still behaves "as if" it is subject to adjustment costs. Section 3 describes the GitHub data and estimates congestion in the onboarding of juniors on open source software projects. Section 4 calibrates the onboarding function in Section 2's problem to match Section 3's estimates and embeds it in an otherwise standard general equilibrium new Keynesian model with nominal wage rigidity and idiosyncratic risk in the production of investment goods. This model extends the partial equilibrium, numerical results of Section 2 to a general equilibrium setting, demonstrating that the response to monetary policy shocks is hump-shaped as in a model with convex investment adjustment costs. Section 5 concludes.

2 Simple Congestion Model

This section develops a simple partial equilibrium model of congestion in onboarding, with three main results. First, under a relatively strict set of assumptions, we show that subjecting investment-producing firms to congestion in onboarding yields an optimization problem which is equivalent to the problem of a firm facing convex investment adjustment costs, thus providing a microfoundation for such costs. Second, under a more general set of assumptions, we show numerically that firms subject to congestion in onboarding hire workers slowly in response to good shocks and hoard workers in response to bad shocks. This confirms that firms continue to behave as if they are subject to convex adjustment costs in partial equilibrium.[12] Finally, studying the firm's problem introduces the key, novel feature of the model that we can estimate in the data: the onboarding function ρ. It also formalizes a key testable

[12]By partial equilibrium, we mean that the analysis here considers the firm's response to an idiosyncratic shock holding critical prices, like the wage, fixed. Section 4 relaxes this assumption.

assumption on the shape of the onboarding function, motivating the empirical analysis in Section 3.

We begin by outlining the firm's objective function and constraints. A representative investment-goods firm produces intangible investment (e.g., R&D or software) I_t and sells it to a representative household at price P_t^k.[13] There are decreasing returns to scale at the firm level and labor is the only factor of production. Letting S_{t-1} be the stock of onboarded (s)enior workers, firm output is $I_t = S_{t-1}^{\nu}$ with $\nu < 1$ as in e.g. Anzoategui et al. (2019) and Schmöller and Spitzer (2021).

So far, we have assumed nothing novel. The simplifying assumption that intangible output is produced with labor as the sole factor of production reflects the fact that a distinguishing feature of R&D spending is that the majority is spent on the wages and salaries of "highly educated scientists and engineers" (Hall and Lerner, 2010).[14] Diminishing marginal returns reflects results in Griliches (1990) on the relationship between patents and R&D spending. In practice there is much uncertainty about ν and we will calibrate it to be close to one, as the assumption of diminishing marginal returns is not critical to our results (see Section 4). What is critical to obtaining sticky behavior for investment spending I_t, and novel to this paper, is the assumption that the workers who produce it, S_{t-1}, are chosen by the firm subject to congestion in onboarding new workers.

Specifically, we assume senior workers come from junior workers J_t who will successfully onboard with endogenous probability ρ, which we will assume—and then test—is a declining function of J_t/S_{t-1}. The law of motion for S_t is thus

$$S_t \leq (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right)J_t, \tag{1}$$

where $d \in (0,1)$ governs exogenous separations. Our preferred interpretation of endogenous

[13] I_t could either be accumulated into a capital stock and rented out directly as in a vertical model of innovation (Bianchi et al., 2019) or represent new "ideas" or varieties in a horizontal model of innovation, which produce monopoly rents that the household values at some P_t^k (Moran and Queralto, 2018). Section 4 will assume the latter.

[14] See Bloesch and Weber (2021) for estimates of the aggregate labor content of IPP overall, which is similar to construction after accounting for the input-output structure of investment spending. Altering the model to include capital in the production of intangible investment goods I_t would diminish the ability of congestion to explain sticky investment output I_t only to the extent that capital is both (a) substitutable with labor and (b) easy to adjust. We abstract from this possibility.

probability ρ is that the onboarding process requires attention and supervision from workers while the juniors acquire firm- or project-specific capital necessary to become productive. Underlying this functional form, we can think of seniors as having a fixed time budget to allocate to onboarding juniors which is less effective when stretched across more and more juniors.[15]

Given these constraints, the firm maximizes the expected, present discounted value of current and future profits. Letting $\Lambda_{0,t}$ be the discount rate between time 0 and t, the firm maximizes

$$\mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t(S_{t-1}+J_t)\right]\right], \tag{2}$$

subject to the constraint that $J_t \geq 0$ and where the wage W_t paid to junior and senior workers is assumed to be identical. While unimportant for establishing the correspondence between our model and a model of convex investment adjustment costs, this simplifying assumption highlights the fact that when human capital acquired on the job is firm-specific, wages will not track productivity because workers cannot threaten to take their firm-specific capital to a different employer.[16] To avoid these difficulties with using on-the-job wage growth to infer the acquisition of firm-specific human capital, we turn to productivity data from GitHub. The model abstracts from human capital that is *not* firm-specific for simplicity.

We can gather these assumptions into the following optimization problem: firms choose paths for $\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}$ to solve

$$\max_{\{I_{t+1},J_t,S_t\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t(S_{t-1}+J_t)\right]\right]$$

[15] An alternative foundation for this functional form could be that seniors' time and attention is necessary for on-the-job screening for highly idiosyncratic skills or idiosyncratic match quality, without which juniors will not be productive or cannot be trusted to work independently.

[16] We assume no R&D firm can pay below W_t due to the presence of an outside sector (producing consumption goods, in Section 4) which does not face congestion and treats all workers identically, so that any S or J worker can always immediately take a job at W_t in this sector. S workers can still threaten to leave in an attempt to convince the firm to pay $W_t + \epsilon$. The fact that we assume wage growth is zero as workers transition from J to S reflects a limiting case in which S workers have no bargaining power after they onboard ($\epsilon \to 0$) and are hence indifferent between staying, leaving for a job in the outside sector, or leaving to begin anew as a J worker at a different R&D firm.

subject to

$$
\begin{aligned}
I_t &= S_{t-1}^{\nu} \\
S_t &\leq (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right) J_t \\
J_t &\geq 0.
\end{aligned}
$$

We next elaborate a special case in which this problem simplifies to the problem of a firm choosing investment production subject to convex investment adjustment costs.

2.1 Congestion in Onboarding and Exact Equivalence

Under some mild assumptions regarding $\rho(x)$, when shocks are small this problem is identical to the problem of a firm choosing the optimal level of investment subject to convex adjustment costs. To see this, assume the law of motion for S binds so that equation (1) becomes:

$$S_t = (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right) J_t. \tag{3}$$

and assume that optimal $J_t > 0$, so that we can ignore the constraint that $J_t \geq 0$. In other words, assume that bad shocks are always small enough that the firm only ever reduces its size by slowing the pace of hiring to below the quantity necessary to replace exogenous separations, and not by implementing a hiring freeze (i.e. $J_t = 0$) or firing senior workers (i.e. choosing $S_t < (1-d)S_{t-1}$). In this case, the following proposition holds:

Proposition 1. *Consider the problem of a firm choosing paths $\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}$ subject to the law of motion (1) and the production function $I_t = S_{t-1}^{\nu}$ to maximize the expected, present discounted value of current and future profits (2). In a solution where (1) binds always and $J_t > 0$ always, then the firm's problem can be written as:*

$$\max_{\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty} \Lambda_{0,t}\left[P_t^k I_t - W_t(S_{t-1} + J_t)\right]\right]$$

subject to

$$I_t = S_{t-1}^{\nu}$$
$$S_t = (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right)J_t$$

where $\rho(x) \in [0,1]$ *on* $x \in [0,\infty)$ *and* $\rho'(x) < 0$. *Let* $f(x) \equiv \rho(x)x$ *be strictly increasing on some domain* D *that does not restrict the firm's optimal choice. Then there exists an equivalent maximization problem yielding the same solution for* I_t*:*

$$\max_{\{I_{t+1}\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t\left(1 + \underbrace{\Phi\left(\frac{I_{t+1}}{I_t}\right)}_{\substack{\textit{Convex Adjustment Costs} \\ \textit{from Onboarding}}}\right)I_t^{\frac{1}{\nu}}\right]\right]$$

and a domain G *which does not restrict firm's optimal choice and where* $\Phi' > 0$ *on* G. *Further, if* $f''(x) < 0$ *on* D *then* $\Phi'' > 0$ *on* G.

See Appendix B for proof and a discussion which demonstrates that the assumption that $f(x) \equiv \rho(x)x$ is strictly increasing and strictly concave on some interval D does not restrict $\rho(x)$ to some exotic function, and would be satisfied by $\rho(x) = b - ax$ or $\rho(x) = \frac{1}{ax-b} + c$, for example. We will show that $\rho(x)$ is likely better approximated by the latter function (i.e. ρ is not globally linear) but use the former in our quantitative exercises. The key testable assumption is that $\rho(x)$ is decreasing.

To understand why our investment adjustment costs are denominated in terms of the wage, W_t, note that in an intermediate step we show that $\rho(x)$ decreasing implies the existence of convex *labor adjustment costs* to changing the stock of S workers. To see this, note we can plug in the binding law of motion (3) to eliminate J_t and recast the firm's problem in terms of choosing S_t and J_t to maximize

$$\max_{\{I_{t+1},S_t\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t\left(S_{t-1} + \underbrace{\mathcal{F}\left(\frac{S_t}{S_{t-1}}\right)}_{\substack{\text{Labor} \\ \text{Adjustment} \\ \text{Costs}}}S_{t-1}\right)\right]\right],$$

subject to the constraint that $I_t = S_{t-1}^{\nu}$ with $\nu < 1$. It can be shown that the labor ad-

justment cost function $\boldsymbol{\mathcal{F}}(\cdot)$ is an increasing, convex function whose existence and properties rely on a key testable assumption: that $\rho\left(\frac{J_t}{S_{t-1}}\right)$ is decreasing (see Appendix B).

Given these convex labor adjustment costs, using the constraint $I_t = S_{t-1}^{\nu}$ to substitute out S_t yields the maximization problem in Proposition 1:

$$\max_{\{I_{t+1}\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty} \Lambda_{0,t}\left[P_t^k I_t - W_t\left(1 + \underbrace{\Phi\left(\frac{I_{t+1}}{I_t}\right)}_{\substack{\text{Investment} \\ \text{Adjustment} \\ \text{Costs}}}\right) I_t^{\frac{1}{\nu}}\right]\right],$$

where the investment adjustment cost function $\Phi(\cdot)$ is again convex if $\rho\left(\frac{J_t}{S_{t-1}}\right)$ is decreasing.

The next section explores the implications of decreasing $\rho(x)$ (congestion) in a numerical setting with occasionally-binding constraints which relaxes the "small shocks" assumption made here.

2.2 Congestion in Onboarding With Large Idiosyncratic Shocks

Relaxing the assumption that the R&D firm never lays off workers or implements a hiring freeze does not qualitatively change the results. To see this, consider a numerical solution to the firm's problem where prices (P_t^k and W_t) are taken as given and constant, but there is exogenous risk in the production process for R&D. Output is now given by:

$$I_t \equiv e_t S_{t-1}^{\nu},$$

where e_t is a productivity shock which follows a persistent Markov process. For simplicity in this section, we assume e_t only takes on two states: high or low.[17] Finally, we assume the firm discounts the future at an interest rate $1 + r$ also taken as given and constant. We can then solve for the firm's optimal choices given an appropriate calibration. Critically, this calibration assumes $\rho(x)$ is a decreasing function.[18]

[17] It may seem superflous to introduce the new variable e_t given that the firm's problem treats changes in P_t^k and e_t as identical shocks to the marginal revenue product of S workers. However, we will need this formulation when introducing idiosyncratic risk in Section 4's general equilibrium model with heterogenous firms, where P_t^k is endogenous. Thus, we introduce productivity shocks e_t now.

[18] Other than the number of states in the Markov process, the calibration for $\rho(x)$ and parameters ν and d follows the general equilibrium model in Section 4. Prices W, P^k and r are calibrated here to the endogenous

A firm at time t with idiosyncratic productivity e_t and incumbent, senior workers S_t has the following value function: plugging in the constraint $I_t = e_t S_{t-1}^{\nu}$,

$$V_t(e_t, S_{t-1}) = \max_{J_t, S_t} \left\{ P^k e_t S_{t-1}^{\nu} - W(S_{t-1} + J_t) + \frac{E_t[V_{t+1}(e_{t+1}, S_t)]}{1+r} \right\}$$

Senior workers separate at rate d and juniors J_t become productive at endogenous rate ρ:

$$S_t \leq (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right) J_t$$

$$J_t \geq 0$$

An increase in e_t is a positive shock to the marginal revenue product of the firm's workers. Accordingly, transitioning from the low to the high state will cause the firm to increase in size. The grey arrows in Figure 1 illustrate the adjustment of a firm that has been in the low productivity state for a long time transitioning to the high productivity state. Conditional on remaining in the high state, the firm slowly hires new workers, since congestion in onboarding means that hiring many J's at once is costly, eventually converging to the long run optimum given by S_∞.

To show the delayed response to negative shocks, we can work through the opposite case of a firm that has been in the high productivity state for a long time (choosing S_∞ in Figure 1) and transitions to the bad state with low marginal revenue products. The firm "hoards" S workers and responds by implementing a hiring freeze ($J = 0$), letting exogenous separations bring the firm to the long run optimum for the low productivity state (S_0 in Figure 1). This behavior is optimal because the S workers have option value: if the firm returns to the high state, it will have to pay heavy costs to rebuild the team, and so it avoids letting the size of the team get too small too quickly. Indeed, the subtle kink in the firm's policy function in the low state (the blue dotted line) reflects the point at which setting $J = 0$ sees the firm shrink too quickly, so the firm chooses $J > 0$.

Note that this labor hoarding behavior does not depend on congestion, and would be present even in a standard fixed hiring cost model. However, without congestion, there is a

steady-state solutions that arise from this calibration when solving the model in Section 4.

strong asymmetry as the firm adjusts immediately to positive shocks. Figure 2 shows this by repeating the exercise in Figure 1 but for the case where $\rho(x)$ is nearly constant, i.e. $\rho'(x) \approx 0$, which is identical to a model with a fixed cost of hiring new workers.

This model of congestion in onboarding was motivated by key features of the process of software development observed in GitHub data, which we have shown can map into a model of convex investment adjustment costs given appropriate concavity of $\rho(x)x$. The next section uses data on software developers collaborating on GitHub to investigate whether ρ is a function of J_t/S_{t-1} with $\rho'(x) < 0$ by estimating ρ as a function of J_t/S_{t-1} non-parametrically.

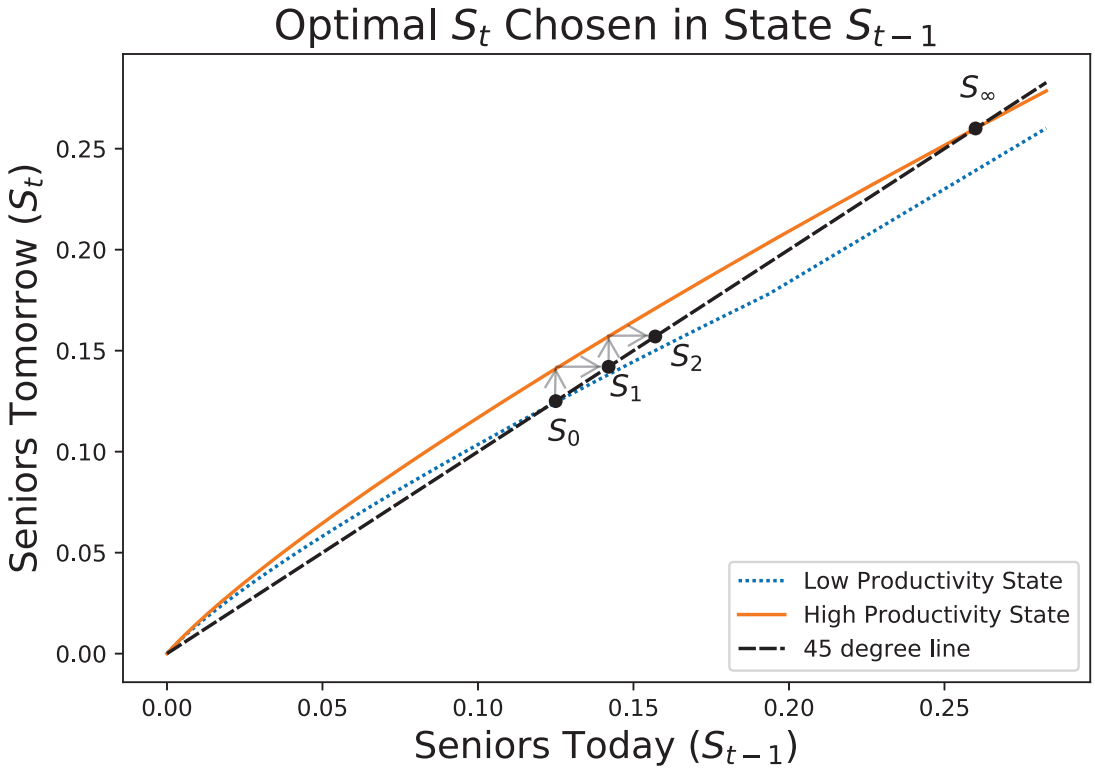
3 Evidence for Congestion in Onboarding from GitHub

Investigating $\rho\left(\frac{J_t}{S_{t-1}}\right)$ requires two steps. First, we identify J workers and S workers. Second, we non-parametrically estimate the probability that a J worker successfully transitions to an S worker as a function of current J_t/S_{t-1} to evaluate the shape of the ρ function.

Leaving aside issues of identification in the second step for the moment, note that measurement of J and S is difficult, since the distinction between J and S that we wish to explore is the acquisition of team- or project-specific capital (which the model collapses to a binary for tractability). Wages imperfectly track marginal productivity increases resulting from the acquisition of this kind of human capital (Caplin et al., 2022; Kline et al., 2019) and need not do so at all as in the limiting case described above where the firm has all the bargaining power. While the limiting case may seem extreme, the empirical prediction that wages do not rise with the initial on-the-job acquisition of project-specific capital seems appropriate for highly educated knowledge workers who are often salaried and/or take compensation as stock options, exercised long after the date of hiring.[19] As we will show, there are substantial productivity gains within the first six months of joining a project in the sample of R&D workers that we study, so that using salaried workers' annual wages to investigate on-the-job productivity growth would be restrictive. To establish this fact and establish a definition for

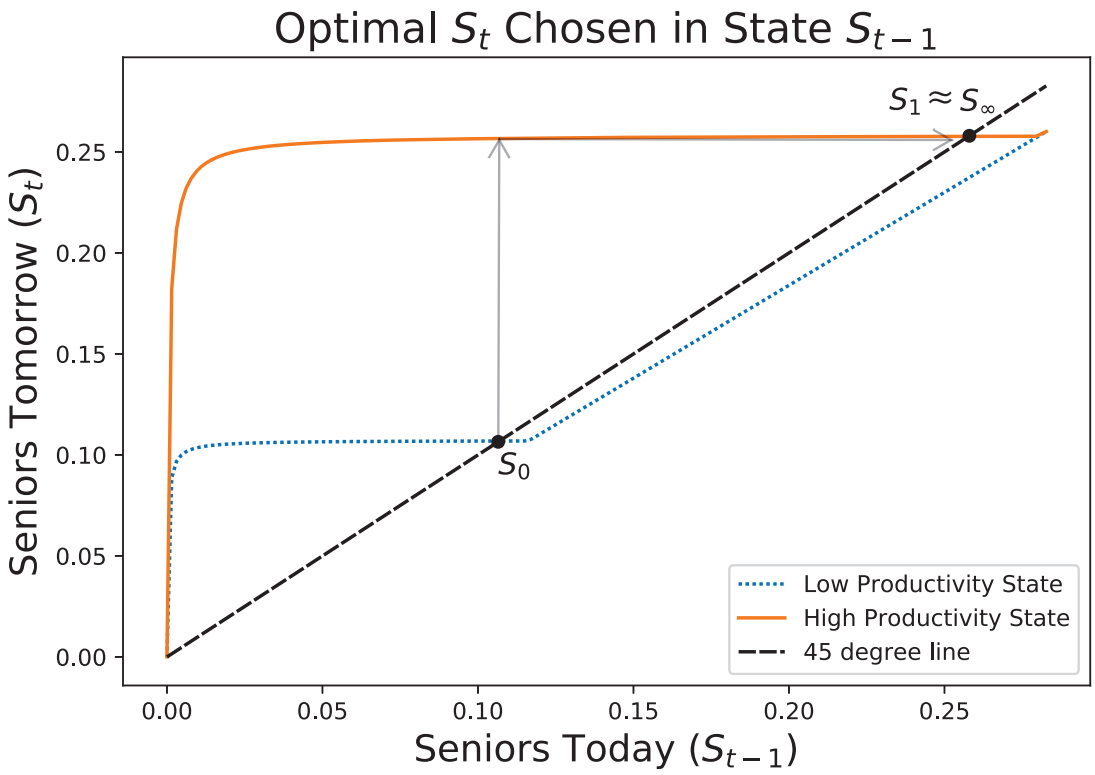
[19]See e.g. Mehran and Tracy (2001). Sun and Xiaolan (2019) show formally how such long-term wage contracts are optimal when human capital acquired on the job is imperfectly portable (i.e. is firm specific).

Figure 1: **With Congestion ($\rho(x)$ Decreasing):** Firm Hires Slowly in Response to Positive Shocks



Notes: The firm's optimal choice of senior workers tomorrow, S_t, given seniors today, S_{t-1}. There are two lines because this choice depends on workers' productivity, which can be low or high. The figure illustrates slow adjustment for a firm with S_0 senior workers transitioning from the low productivity to the high productivity state in period $t = 1$. Grey arrows trace out the firm's choices at $t = 1$ and then $t = 2$ assuming it remains in the high state. Note that these choices S_1 and S_2 remain far below the long-run value S_∞. The firm also slowly adjusts to negative shocks by "hoarding labor" in case it transitions back to the good state. Adjustment is slow because the firm implements a hiring freeze ($J_t = 0$) and lets exogenous separations slowly reduce the stock of senior workers S_t.

Figure 2: **Without Congestion ($\rho(x)$ Flat):** Firm Immediately Adjusts to Positive Shocks



Notes: This figure repeats the exercise in Figure 1 for the case where $\rho(x)$ is nearly constant, i.e. $\rho'(x) \approx 0$. This is identical to a fixed cost hiring model, yielding slow adjustment to negative shocks from labor hoarding (not shown), but rapid adjustment to positive shocks (grey arrows).

J and S workers that we can use to estimate ρ, we turn to productivity data from GitHub.

3.1 GHTorrent Data

GitHub is an online collaboration platform and version control service founded in 2008. It was acquired by Microsoft in 2018 for $7.5 billion USD, reflecting the platform's popularity both for the development of proprietary projects and Open Source Software (OSS). We use data on OSS projects collected systematically from GitHub by Gousios (2013) and made available through Google BigQuery.[20] Collection in GHTorrent began in February of 2012, with information extended back to 2008, and data is available up through 2019Q2.[21] The GHTorrent dataset grows exponentially in size over time and is large (on a terabyte scale). We provide a brief, high-level description of the dataset here; Appendix D provides a detailed description of how the GHTorrent data is structured, accessed, and cleaned by us for the purposes of estimating the regressions described below.

GitHub is the dominant version control service in use today: in a 2021 survey, 91% of software developers globally reported using GitHub for either personal projects or at work.[22] While not every company uses GitHub, the production and code review process that GitHub enables – the "Pull/Merge" model of development – is ubiquitous; 84% of developers reported using this model while at work, which makes it nearly as common as email at 90% (JetBrains, 2021). This development process works as follows:

1. A user creates a project ("repository") and allocates power to other trusted users to approve changes (seniors).

2. Potential contributors, junior and senior, propose changes (through "pull requests").

3. Seniors examine the submitted code, leave comments and request alterations before approval ("merging the pull request") in a process called "code review."

[20] GHTorrent is the most popular source for researchers using GitHub data as Cosentino et al. (2016), document. For a comparison of the costs and benefits of other methods, see Mombach (2019).

[21] We access GHTorrent through BigQuery. Since collection began in 2012, we do not have information on projects e.g. created in 2008 and deleted in 2010.

[22] JetBrains conducts an annual "State of Developer Ecosystem" industry survey, which in 2021 included responses from "31,743 developers in 183 countries or regions" (JetBrains, 2021).

Figure 3: Comments On Proposed Contributions Made During Code Review on GitHub

body
(In short, `()` + arguments to constructor are missing there in fields definition.) Again, I am not writing code for you. I just show you ideas. You should understand what code is doing and finish things.
Ahh, the coobooks in infra/cookbooks are public cookbooks and I just cloned the locally and copied them over. We could introduce another tool called [Librarian](https://github.com/applicationsonline/librarian) which is like Bundler for chef. I'm cool with
@irium Just push new commits to the *same branch* where you originally sent the [pull request] (https://help.github.com/articles/using-pull-requests) from. They'll automatically show up on this page.
In case features.json is not present or broken (for example the first time you run it) the .bak file is used... Should I rename it?
1: D200 One-line docstring should not occupy 3 lines 4: I102 copyright year is outdated, expected 2014 but got 2013 43:16: E203 whitespace before ':' 44:20: E203 whitespace before ':' 45:18: E203 whitespace before ':' 46:18: E203 whitespace before ':' 47:
这个是draggable运行的demo页面，可以不review，发布前会制作更好的demo。
Let's remove this file.

Notes: Code review is not simple yes/no approval. It requires time and attention from seniors as they interact with juniors, giving juniors the opportunity both to learn how to contribute and signal competence. A good track record leads juniors to be promoted to seniors. Source: Pull Request Comments on GHTorrent, accessed via Google BigQuery.

Code review is thus an opportunity for juniors to learn how to contribute and signal competence. Over time, a good track record leads juniors to be promoted to seniors. However, juniors do not "graduate" from code review: it is common practice for all code to be at least nominally reviewed, no matter how experienced the contributor, on both OSS projects and in private sector, commercial code development.[23] We thus observe, for each user, their history of attempted contributions to various projects, if and when those changes were approved, and the comments made during code review. Figure 3 presents a selection of these comments.

Is Pull/Merge development in OSS projects representative of private sector, commercial development? We consider the following dimensions: the way GitHub and the Pull/Merge model is used; the nature of the users; and the nature of the projects.

Regarding the Pull/Merge model, survey evidence suggests that the way pull requests are used in private GitHub projects—to self-assign tasks and facilitate code review—is identical for both OSS and commercial development. This reflects the fact that most commercial software development is collaborative and that most commercial software developers on

[23]See Kalliamvakou et al. (2015) Figure 1 for developer workflow in commercial projects using GitHub.

GitHub report contributing to OSS projects as well (Kalliamvakou et al., 2015). The now widespread commercial adoption of the Pull/Merge model and OSS development methods for the use of *proprietary* software development (so-called "Inner Source" development) reflects the historical success of the open source model in developing a number of high quality, successful products including Linux, Apache, MySQL and PHP/Perl/Python (Stol et al., 2014). The adoption of these methods was often driven from the "bottom up" by developers who realized they would be helpful for proprietary software development (Stol et al., 2014; Kalliamvakou et al., 2015); see Appendix C for additional detail on the history of industry adoption of OSS development methods.

Regarding users, note that some OSS projects are in fact maintained and developed by paid employees. To understand this, note that while the cost to the firm of making code open source is an inability to charge for it later, open source development creates the opportunity for users to alert the firm to problems (free debugging and testing) or to add features (free development) which benefits the firm when the OSS project is a tool used internally; see e.g. Lerner and Tirole (2005) for a deeper explanation of why firms may want their paid employees to work on OSS projects, or to make proprietary projects OSS. Consistent with this, on both OSS and "Inner Source" projects within large, private firms, it is widely acknowledged that it is the users of a project who become contributors through discovering bugs or out of a desire to improve functionality for their own purposes (Stol et al., 2014).

Moreover, many government agencies develop code openly and provide it as a public good: Mergel (2015) finds over 7,000 government owned OSS repositories on GitHub (87% of which were for the development of software, as opposed to e.g. sharing data or joint editing of text documents) including the Department of the Interior, NASA, and the Department of Defense.[24] In practice most observed activity on OSS GitHub projects occurs during business hours, dropping on holidays and weekends (Gousios and Spinellis, 2012; McDermott and Hansen, 2021), which suggests that much OSS development happens at work.

However, most contributors are not directly hired to work by an OSS project's owners,

[24]Our sample of "active" projects as of 2019Q2 includes 36,537 repositories, prohibiting individual inspection. However, we can easily identify some government-maintained projects by filtering for repository names which contain "gov" or ".gov". Fewer than 1% of all repository names contain these strings, which includes repositories belonging to the cities of Boston and Philadelphia, and also a significant UK presence: the Government Digital Service is responsible for over 100 repositories; see `https://github.com/alphagov`.

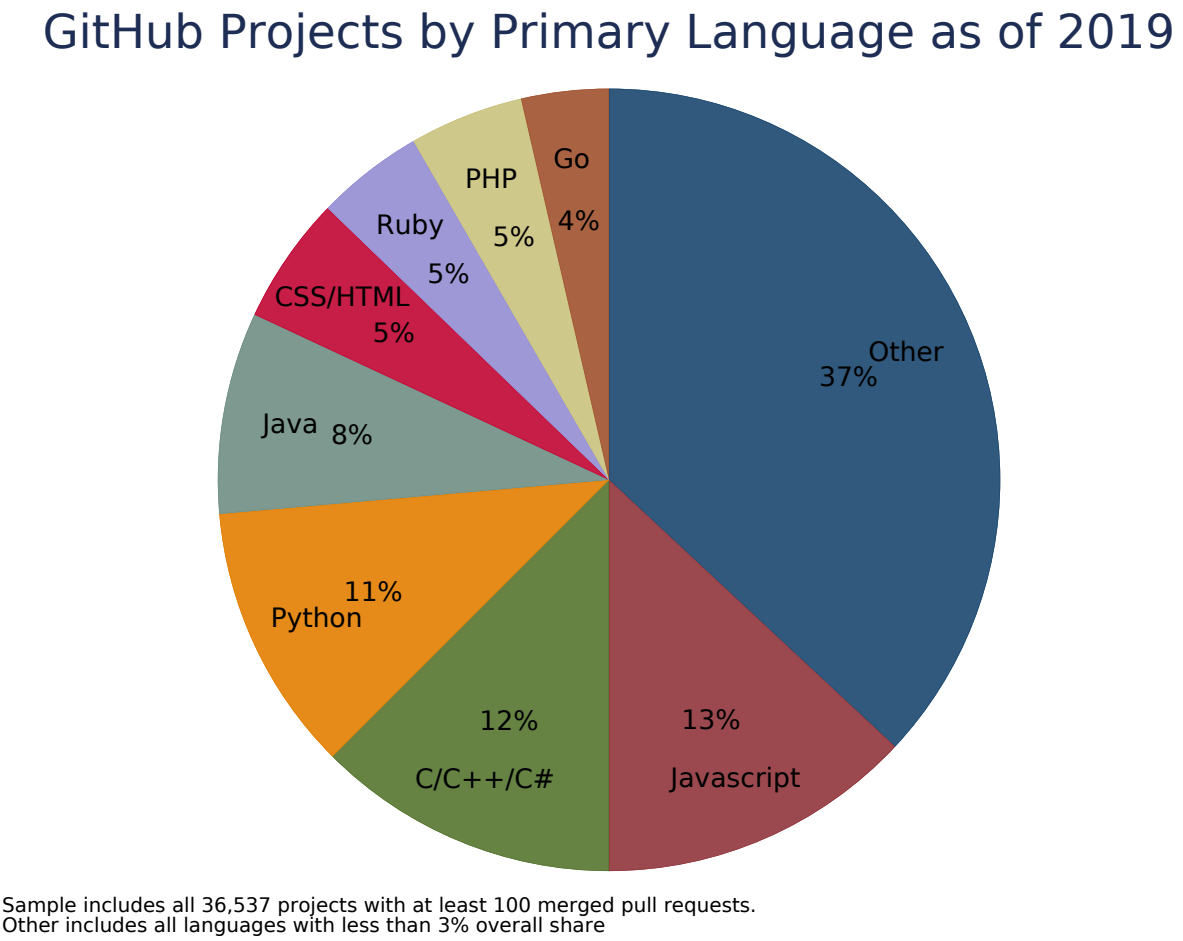
who are often private individuals rather than companies or governments. Beyond the fact that volunteers contribute to the OSS projects that they use in order to adapt them for their own ends, as just discussed, motives for volunteer OSS contribution can include learning or reputation building, and turnover on projects is likely high relative to the private sector; see Vasilescu et al. (2015) for a discussion. This does not mean that most users are students, and indeed most OSS contributors are professionals: survey evidence suggests that the median GitHub user is 29 years old (mean 30) with 8 years of IT experience (mean 10.5) in the United States or Europe (Vasilescu et al., 2015). Moreover, survey evidence generally reveals a contributor's own need for software as the primary reported motivation for OSS contributions.[25] Though their activities have significant positive spillovers to other users, volunteer contributors are not pure altruists.

Finally, OSS projects in our sample are not dominated by personal projects or spam websites. This is because our sample of repositories restricts to large repositories with at least 100 contributions (i.e. merged pull requests) and which we call *projects*.[26] Focusing on such "active" projects with a minimum number of contributions is considered best practice in the literature on OSS software development to ensure that we are isolating projects which are true attempts to collaboratively develop software, although there is no specific guideline for what counts as "active" (Kalliamvakou et al., 2014). Given this, we have checked that our results are robust to changes in this threshold value (we tried 100, 120, and 200; see Appendix D for details). This approach naturally restricts the sample of repositories to large collaborative projects because it is technically possible to work on GitHub without using the pull request model, which effectively adds extra steps to aid in code review. While almost no commercial projects use GitHub this way (Vasilescu et al., 2015) many small or personal projects proceed by making changes ("commits") without pull requests and are thus effectively excluded from our analysis. As Figure 4 shows, the most popular primary languages in our sample are Javascript, C, and Python; projects written in CSS or HTML

[25] See Hertel et al. (2003) for a study of Linux contributors; Lakhani and Wolf (2003) for various projects on Sourceforge; and Hann et al. (2004) for the Apache project.

[26] Note that Kalliamvakou et al. (2014) warn against conceptually thinking of "repositories" as projects like we do here. This is because certain activity measures, like the number of commits, are mis-measured unless one combines each repository with all related "forked" repositories. A downside is that some forks are indeed new projects. We refrain from combining repositories with their forks for analysis since our measure of time-to-merge for merged pull requests (discussed in Section 3.2) does not suffer this measurement problem.

Figure 4: Distribution of Programming Languages



Notes: Weights each project by total number of contributions (merged pull requests). Unweighted results are similar. "Other" includes languages like R and Matlab which are a very small share of the projects in our sample. Source: GHTorrent and authors' calculations.

make up only 5% of our sample (e.g., large, jointly-developed websites).

Even after keeping only "large" repositories that contain many merged pull requests, there are a few "test repositories" that do not represent collaborative software development. These are characterized by many pull requests with very short merge times. Similarly, some users are actually bots. These are not difficult to detect and we remove them manually by filtering for repositories with the phrase "test" in the name or with implausibly low average approval times, and by removing users with variations of the name "bot" following Wyrich et al. (2021).[27] However, this highlights the fact that the exact way GitHub is used may

[27]Some GitHub accounts are "Organizational" and stand in for groups of users. We drop such "users"

vary across both users and projects, which informs our analysis below.

3.2 Onboarding: Identifying J vs. S in GHTorrent

This section estimates how productivity evolves over time on OSS projects in GHTorrent, establishing that there are non-trivial productivity gains over time in the first six months of experience. With this fact in hand, we will define workers J who successfully onboard and become S as those newcomers that remain over six months and/or begin to engage in reviewing the code of other contributors. This definition will then enable us to observe how this onboarding probability varies with the ratio of newcomers to incumbents, J_t/S_{t-1}.

Of the various productivity metrics considered in the literature, we use approval time (i.e. the length of the code review process) for a user's contributions as our measure of that user's productivity.[28] Since this is both a direct measure of how long it takes a user to close an issue and a direct measure of how much "hand-holding" the team thinks a user needs, it is a natural metric to study the onboarding process. As we will show, approval time shrinks dramatically with initial increases in project-specific tenure. Consistent with this interpretation, we examine the number of comments each contribution receives during code review, finding that there is less discussion as juniors gain experience on a project.

Many factors determine approval time beyond individual competence, which motivates the inclusion of controls in our regressions. Forsgren et al. (2021) criticize single-factor measures of performance for the purposes of employee evaluation on the grounds that they are influenced by project-specific factors beyond the control of individual programmers, aligning with prior work on the determinates of approval times in OSS projects from GHTorrent. While changes of good quality and changes that match a project's "roadmap" have a better chance of being accepted, and while a developer's track record can positively influence approval time, project size and complexity also affect approval times. While these can be handled with project fixed effects, there are also project-individual specific features which

from our analysis.

[28] This is measured as time-to-merge, or the time between opening and merging a pull request. For an overview of this and other commonly used software productivity metrics, see Forsgren et al. (2021). Time-to-merge also has practical advantages in our longitudinal context, as footnote 26 notes. Commits are also a common metric in the literature, partly reflecting a focus on cross-sectional, project-level analyses; longitudinal studies like this paper's following individual developers are rarer (Cosentino et al., 2016).

may cause both longer tenure and faster approval times: for example, a strong pre-existing social connection between the contributor and the project manager. For a survey of papers establishing these facts, see Wyrich et al. (2021). Moreover, a good match in terms of skills between a junior and a particular project (Lazear, 2009) could lead to both longer tenure and faster approval times. An advantage of our setting is that we have rich enough data to estimate individual-by-project fixed effects, controlling for all such confounders.

Finally, we may observe that juniors improve over time on a project because they are acquiring general software development experience. To disentangle the effects of *overall* experience from *project-specific* experience, we control for the overall age of a user's GitHub account, or total tenure on GitHub, in addition to project-specific experience. This is made possible by the fact that we observe the same user working on multiple projects, potentially at the same time, over the course of their career.[29]

Let $y_{i,p,t}$ be either the approval time or total comments received for a contribution opened by user i on project p at time t. We can then estimate the following model via linear regression:

$$\begin{aligned} y_{i,p,t} = &\sum_{j=1}^{13} D(\text{Months Project Experience} = j)_{i,p,t} \\ &+ \sum_{k} D(\text{Months Programming Experience} = k)_{i,t} \\ &+ D_{i,p} + \beta_{PA,p} ProjectAge_{p,t} + \epsilon_{i,p,t}. \end{aligned} \tag{4}$$

The first sum consists of dummy variables for having between one and thirteen or more months of experience on project p at time t, and the second sum consists of dummy variables for overall programming experience measured by GitHub account age at time t.[30] We also allow for individual-by-project fixed effects ($D_{i,p}$) and project-specific linear time trends ($\beta_{PA,p} ProjectAge_{p,t}$). See Appendix D for additional detail.

[29] It is common for developers on OSS projects to work on several projects at once, and even in firms where developers are unable to do so, most wish that they could; see Torkar et al. (2011) Appendix B5. More recent survey evidence suggests that most commercial software developers on GitHub report contributing to OSS projects as well (Kalliamvakou et al., 2015), consistent with the view that such multitasking is normal.

[30] This framing reflects the fact that any mis-measurement due to individuals creating accounts at different stages in their career is absorbed by individual-by-project fixed effects.

Figure 5 uses the marginal effects estimated from equation (4) to compare the unconditional mean values for a user with zero months of project-specific experience to predictions for an otherwise identical user with varying degrees of project-specific experience. This reveals that approval time falls precipitously in the first six months of project-specific experience, roughly leveling off thereafter (though standard errors increase). Newcomers also need less "hand-holding" over the same period of time, as the average number of comments per contribution declines for the first six months before leveling off.

Consistent with this, Figure 6 demonstrates that most work is done by users with at least six months of experience, though precisely quantifying "work done" is difficult as we do not observe the content of each contribution. Given that large or complex tasks take longer to be approved (Gousios et al., 2014; Wyrich et al., 2021) and that more experienced developers take on more difficult tasks (Torkar et al., 2011; Subramanian, 2020), this should bias our results against finding positive effects from tenure. In light of this, we view our results as a plausible lower bound on the returns to project-specific tenure.

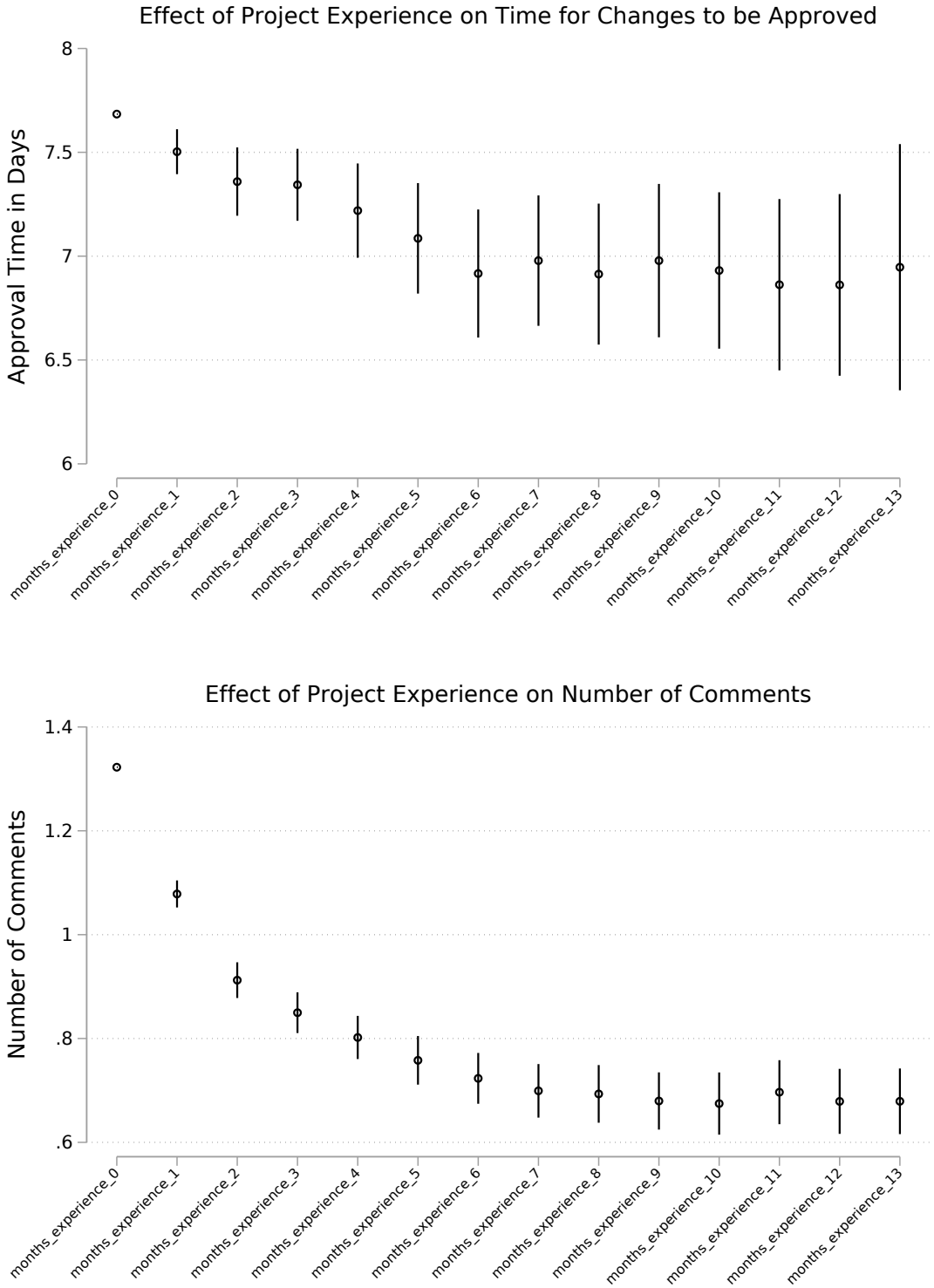
We interpret these documented returns to project-specific tenure as reflecting a combination of skill-acquisition and earned trust or reputation within a team, which our model in Section 2 is general enough to encompass. We emphasize the acquisition of project-specific skills, as this frequently arises in interviews with practitioners. Appendix C elaborates on this narrative evidence. All this suggests that attention from incumbents should matter for successful onboarding. We test this in the next section.

3.3 Congestion in Onboarding: Estimating $\rho\left(\frac{J_t}{S_{t-1}}\right)$

In this section, we demonstrate that there is congestion in onboarding by estimating ρ non-parametrically as a function of J_t/S_{t-1}. Specifically, we provide evidence that ρ is a decreasing function, which Section 2 shows implies that a firm will behave "as if" it had high adjustment costs to changing the level of investment.

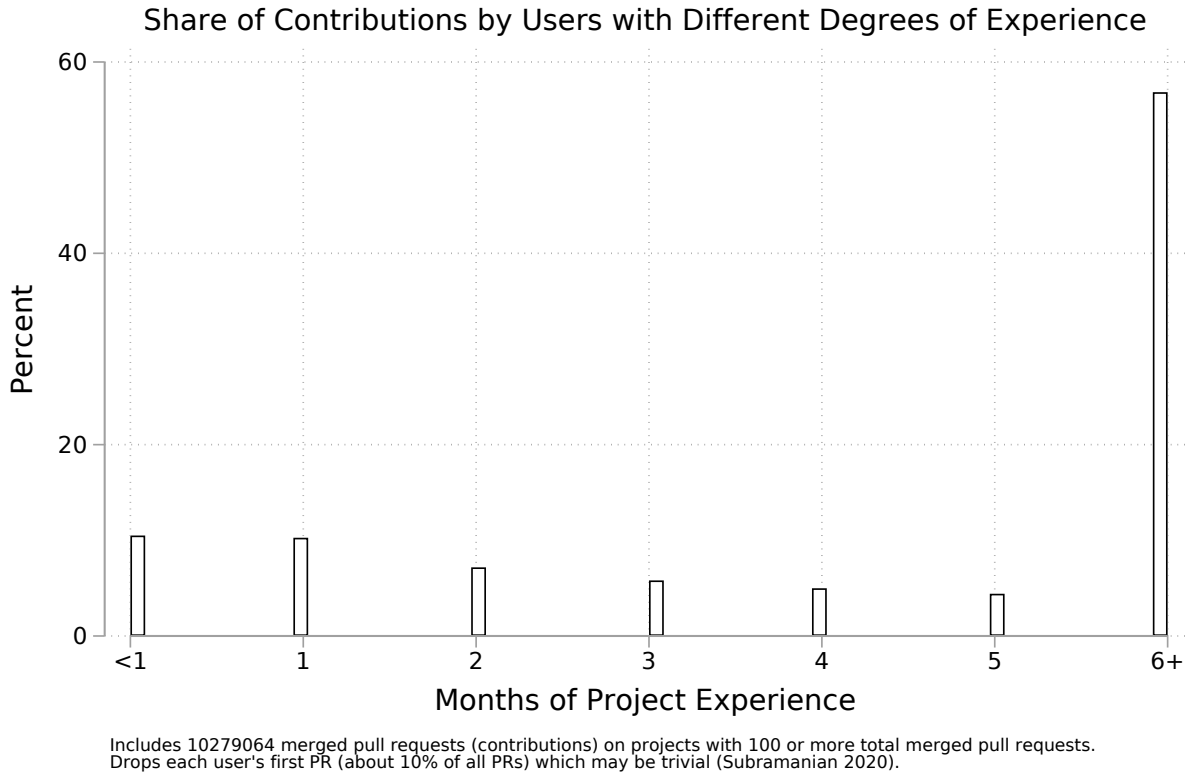
We begin by identifying junior J type and senior S type workers. In each calendar month t and each project p, we assign each user with activity on at least one pull request in p at t into either category J or category S. We drop users who never contribute, and restrict attention to those who will eventually contribute at least once (i.e. open a pull request

Figure 5: **Becoming Productive Requires Onboarding:** over time, new contributors' proposed changes are approved faster, with less discussion



Notes: Unconditional mean values for a user with zero months of project-specific experience (first hollow dot) compared to predictions for an otherwise identical user with one or more months of project-specific experience (capped at 13). See text. Standard errors are clustered at the project level. Source: GHTorrent and authors' calculations.

Figure 6: Most Work is Done by Experienced Team Members



Notes: This figure plots the share of contributions by users with different degrees of project experience at the time of that contribution, showing that most work is done by those who have at least six months of project-specific experience. Since we do not otherwise control for complexity or importance of these contributions, and given that longer-tenure workers take on more complex and important tasks, this figure likely understates the importance of work done by senior contributors. Source: GHTorrent and authors' calculations.

that is merged). A J type transitions to an S type on a particular project either when they have reached tenure of at least six months, or when we observe them reviewing code written by others. Formally, we identify code review in the data when we observe a user merging/closing/commenting on pull requests authored by other users, and project tenure is measured as the length of time between a user's first observed activity and their last observed activity on a project.

Note that this definition implies that some workers are S types from the beginning – presumably e.g. project founders – and never transition.[31] Our binary definition reflects the fact that a majority of juniors have negligible tenure and contribute precisely once, presumably to fix a bug or add a feature they need, while a nontrivial subset continue to contribute for at least six months. These two groups comprise over 80% of all junior-project observations; see Figure 7.

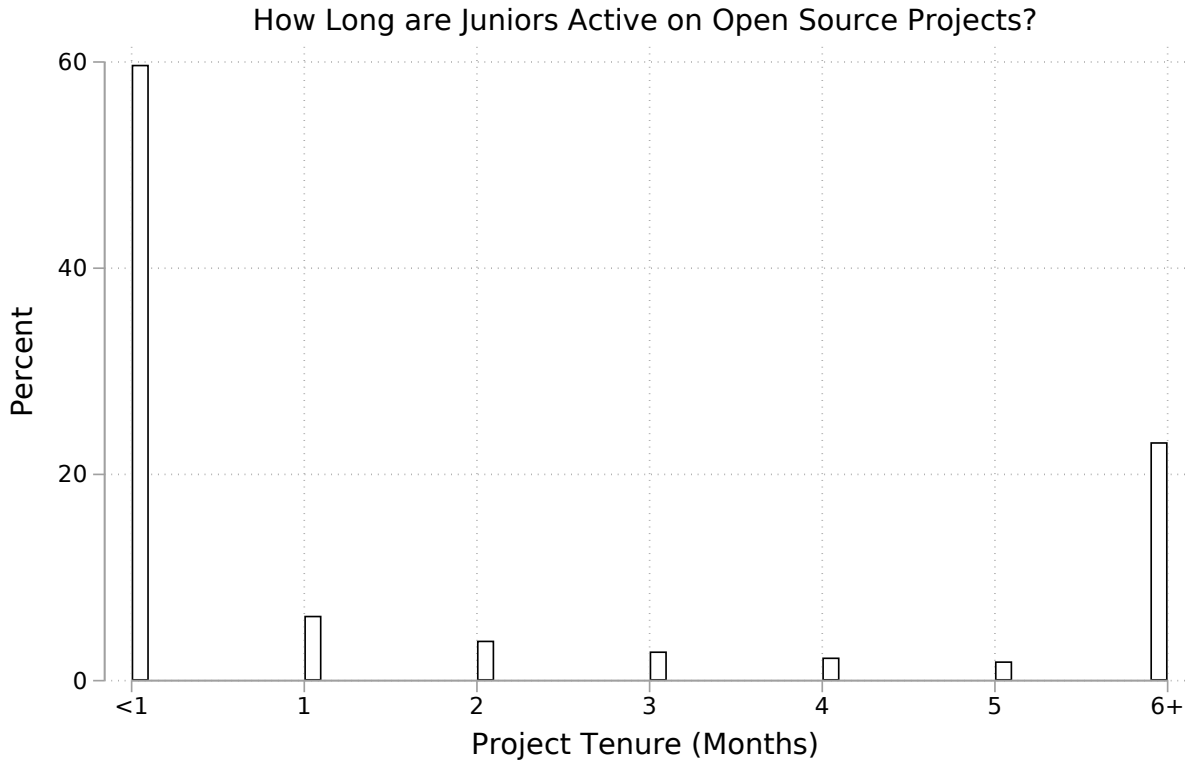
We define the quantity of J types on project p at time t as $J_{p,t}$, tabulated as the number of users who have contributed to that project (i.e. authored at least one pull request that was eventually merged) at time t with less than six months of tenure and who do not engage in code review (i.e. who have not been observed merging/closing/commenting on a pull request opened by someone else). The other active users are summed into $S_{p,t}$. We then estimate a linear probability model: let $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards})$ denote an indicator function for whether a newcomer i on project p (counted in the sum $J_{p,t}$) will eventually transition to being an S type on project p. We then estimate the following via linear regression:

$$\begin{aligned} \mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) &= \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } b\right) \\ &+ D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}. \end{aligned} \tag{5}$$

We estimate the effect of $J_{p,t}/S_{p,t}$ non-parametrically by measuring it as a set of dummy variables representing equidistant bins for junior-senior ratios. Project specific dummies D_p control for unobservable project-specific features that may make some projects easier to join, while $ProjectAge$ is a project-specific time trend meant to capture the project life cycle,

[31] Also note that once a J type worker transitions on a project, they are counted as an S type in any calendar month when they "re-appear" on that project in the pull request data.

Figure 7: Newcomers Either Contribute Once, or Stay a Long Time



Notes: This figure plots the share of all new, junior contributors on various projects by their subsequent observed tenure on that project. Tenure is measured as the length of time between a user's first observed activity and their last observed activity on a project. Most juniors will have very short tenure (rounded to the nearest month) and contribute once, followed by a nontrivial second group who remain much longer. Source: GHTorrent and authors' calculations.

since some projects may become harder to join as they age; $\gamma_{i,t}$ captures newcomer-specific factors, such as account age, which change over time, and X_t are year fixed effects. See Appendix D for additional detail.

We cannot include user-project specific fixed effects here because they are collinear with the outcome variable (we only observe one outcome per project for each individual: either they onboard, or they do not). Relatedly, we cannot well-estimate individual fixed effects because in practice most individuals join very few OSS projects in sample over time. Note if someone joins only one project in our sample of large OSS projects, we cannot estimate a fixed effect for them. Appendix E.2 discusses this and shows that our results are qualitatively unchanged by adding individual fixed effects, though the sample size shrinks.

Figure 8 plots the results for equidistant bins of ratios from just above zero to just over 1:1. In practice, over 75% of all project-month observations have $J/S \in [0, 1]$ but there is a significant fat-tail.[32] The figure compares the unconditional mean onboarding probability for a junior on a project in the smallest bin for $\frac{J}{S}$ ratios (first hollow dot) compared to predicted probabilities for an otherwise identical user as the ratio $\frac{J}{S}$ increases (capped at 1.105), and standard errors are clustered at the project level. The results demonstrate that as the ratio of juniors to seniors increases, the onboarding probability falls. Note the jump in the probability for the bin which contains the exact ratio 1:1, which the regression intuitively associates with a relatively higher onboarding probability. Interpreted causally, these estimates literally show us the shape of ρ.

This causal interpretation requires that the ratio J/S be uncorrelated with the error term $\epsilon_{i,p,t}$. In considering potential violations, it seems most natural to worry that juniors not only choose projects which may be easy to join (captured by project fixed effects) but also choose to join projects *at points in time when projects are easy to join.* For example, certain points in a project's development might make for natural "entry points" and our project-specific time trends may imperfectly capture this. Thus, high $\frac{J_{p,t}}{S_{p,t}}$ may occur when newcomers flock to a project at t to take advantage of a high draw for $\epsilon_{i,p,t}$ which is common to many people, and thus correlated with $J_{p,t}$. Thus, it is possible that we are biased towards

[32] See Appendix E.1 for results with more bins, capturing this tail. This results in a slightly flatter estimate for ρ which leaves the quantitative results in section 4 qualitatively unchanged, though a flatter ρ means less congestion and a less-hump shaped response for investment to monetary policy shocks.

finding an opposite result, or upward-sloping curve instead of the downward sloping nonlinear relationship in Figure 8.

In practice the decision to contribute to an OSS project seems highly idiosyncratic and is often driven by a desire to add needed features or improve functionality for one's own use, as described in Section 3. Consistent with this, the inclusion of controls does not do much to change the shape of the relationship in Figure 8, as changes in the J/S ratio are not correlated with project characteristics. The fact that a large share of project-month observations occupy the space where $J/S > 1$—where onboarding workers is particularly difficult—further suggests that project maintainers (S) do not have much control over how and when newcomers arrive; indeed, in a model with ρ calibrated to match this data, profit-maximizing firms will generally avoid this region. This highlights an advantage of using OSS projects as opposed to *proprietary* projects: to the extent that firms hire at points in time when it is particularly easy to onboard juniors, we would expect this bias towards a flatter or upward-sloping ρ to be severe. Since project maintainers do not have control over when newcomers join, this bias is mitigated in our setting.

To bring Figure 8's empirical results into the model, we specify a simple linear functional form for ρ which approximates the nonlinear relationship:

$$\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = .47 - 0.7\frac{J_{p,t}}{S_{p,t}} \equiv \rho\left(\frac{J_{p,t}}{S_{p,t}}\right).$$

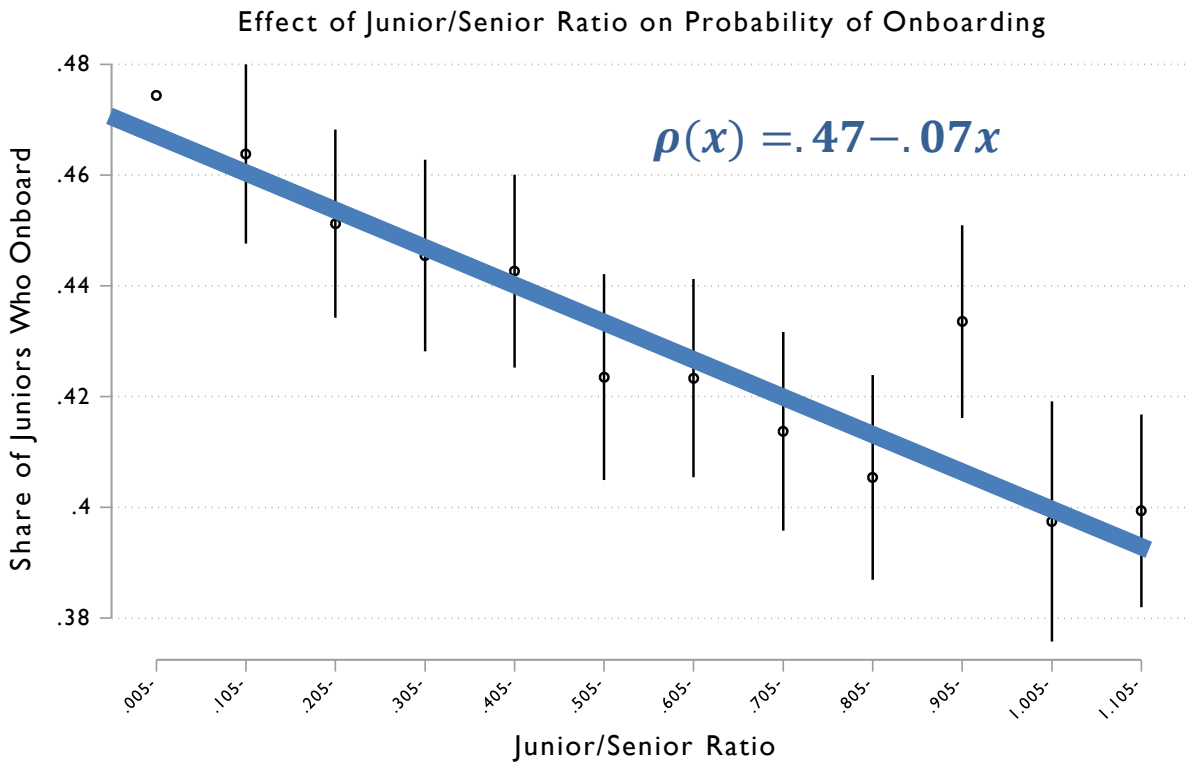
This is plotted as a blue line in Figure 8. We proceed to use this in the next section to illustrate the ability of congestion to generate hump-shaped responses to monetary policy shocks in line with the data.[33]

4 Quantitative Model

This section builds a quantitative model where a continuum of firms produce intangible investment subject to idiosyncratic risk and congestion in onboarding. This extends the results in Section 2.2 to a general equilibrium setting and shows that congestion continues to

[33] Appendix B discusses other potential functional forms and the implications of our linear choice for the firm's problem.

Figure 8: Non-parametric Estimate of the Onboarding Function ρ and Linear Approximation



Notes: Unconditional mean onboarding probability for a junior joining a project at date t in the lowest bin for junior to senior ratio $\frac{J}{S}$ at t (first hollow dot) compared to predicted probabilities for an otherwise identical user as the ratio $\frac{J}{S}$ increases (capped at 1.105). We count juniors as successfuly onboarding if they remain with the project for at least six months or begin reviewing code written by others (merging/closing/commenting on pull requests authored by other users). Note the jump in the probability for the bin which contains the exact ratio 1:1 (i.e. bin .905-1.005) which the regression intuitively associates with a relatively higher onboarding probability. The blue line linearly approximates this relationship for use in Section 4's calibration. See text. Standard errors are clustered at the project level. Source: GHTorrent and authors' calculations.

produce dynamics which are similar to those in a model with standard investment adjustment costs. Specifically, this section shows that congestion yields a hump-shaped and delayed response of intangible investment to monetary policy shocks. We abstract from standard frictions often used to fit the data (e.g. endogenous capital utilization, habit formation, etc.) to isolate the effect of congestion in creating persistent dynamics in the model. We will solve and compare two different models: one with "I-dot" investment adjustment costs following Christiano et al. (2005) and standard investment production, and one where the only friction in investment production comes from congestion as described above.

In both models, a representative household solves a standard optimization problem. The household accumulates intangible capital K_t through intangible investment $INTAN_t$. We assume this is the only capital used by firms and abstract from tangible capital for simplicity. It also trades a riskless bond in zero net supply B_t which pays real interest rate r. The household solves

$$\max_{\{C_t\}_{t=0}^{\infty},\{B_t\}_{t=0}^{\infty},\{INTAN_t\}_{t=0}^{\infty},\{K_t\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\beta^t\left(\frac{C_t^{1-\sigma}}{1-\sigma}-\omega\frac{L_t^{1+\eta}}{1+\eta}\right)\right] \tag{6}$$

subject to standard budget and capital accumulation constraints,

$$C_t + B_t + P_t^k INTAN_t = (1+r_{t-1})B_{t-1} + R_t^k K_{t-1} + W_t L_t + DIV_t \tag{7}$$

$$K_t = (1-\delta)K_{t-1} + \left(1 - \frac{\phi}{2}\left(\frac{INTAN_t}{INTAN_{t-1}} - 1\right)^2\right) INTAN_t \tag{8}$$

where $\phi = 0$ in the congestion model with heterogenous firms. The household earns income from supplying capital K_t and labor L_t to firms, and also potentially from dividends paid by investment-goods producing firms, DIV_t.

A perfectly competitive, representative final consumption good firm produces Cobb-Douglas,

$$C_t = Z_t K_{t-1}^{\alpha} J_{c,t}^{1-\alpha}$$

with the following standard factor demands for capital and labor:

$$R_t^k = \alpha \left(\frac{K_{t-1}}{J_{c,t}} \right)^{\alpha-1}$$
$$W_t = (1-\alpha) \left(\frac{K_{t-1}}{J_{c,t}} \right)^{\alpha}.$$

A continuum of investment goods firms indexed by $i \in [0,1]$ produce intangible investment. Their problem is formally stated below, and involves hiring juniors $j_t(i)$ and seniors $s_t(i)$ to produce intangible investment. We thus define aggregate labor used in the intangible sector as

$$J_t + S_{t-1} \equiv \int_0^1 \left(j_t(i) + s_{t-1}(i) \right) di,$$

so that aggregate labor demand from all firms is given by

$$L_t = J_{c,t} + J_t + S_{t-1}.$$

Regarding wages, we continue to make the simplifying assumption that all workers receive the same wage W_t. This can be thought of as a limiting case of the bargaining problem between each onboarded S worker and their firm, given other assumptions. To see this, note that any worker can take a job in the perfectly competitive consumption goods sector, which does not face congestion and pays all workers their (identical) marginal product. Since any S or J worker can always immediately take a job at W_t in this sector, no R&D firm can pay below W_t. However, S workers in the congestion model can still threaten to leave the firm and attempt to convince the firm to pay $W_t + \epsilon$. The fact that we assume wage growth is zero as workers transition from J to S reflects a limiting case in which S workers have no bargaining power after they onboard ($\epsilon \to 0$) and are hence indifferent between staying, leaving for a job in the outside sector, or leaving to begin anew as a J worker at a different R&D firm.

To introduce wage stickiness, we assume a wage Phillips curve following Erceg et al.

(2000). Denoting gross nominal wage inflation as π_t^w,

$$\pi_t^w(\pi_t^w - 1) = \frac{\epsilon}{\psi}\left(\omega L_t^{1+\eta} - \frac{\epsilon - 1}{\epsilon} W_t L_t C_t^{-\sigma}\right) + \beta \pi_{t+1}^w(\pi_{t+1}^w - 1).$$

Our reliance on this standard formulation (and standard values for ϵ and ψ) for wage stickiness serves the goal of highlighting the role congestion plays in determining aggregate dynamics.

Finally, we assume the central bank sets the nominal interest rate $1 + i_t$ according to a standard Taylor rule. Denoting gross price inflation as $\pi_t \equiv \frac{P_t}{P_{t-1}}$,

$$i_t - i_{ss} = \phi_\pi(\pi_t - 1) + \epsilon_t$$

where ϵ_t is shock following an AR(1) process and ϕ_π determines the responsiveness of the central bank to inflation. The two models we compare differ only in their production of investment goods and choice for adjustment costs, ϕ.

Model 1: Representative Firm with I-dot Adjustment Costs ($\phi > 0$)

The first model assumes simply that all investment firms i are identical and that $INTAN_t = S_{t-1}^\nu$ where S_t is chosen freely each period. J_t is zero here always, so aggregate labor demand is simply $L_t = S_{t-1} + J_{c,t}$. To get hump-shaped impulse response functions, this model needs I-dot adjustment costs with $\psi > 0$.

This model serves as a benchmark for the congestion model, described next.

Model 2: Heterogenous Firms with Congestion in Onboarding

Intangible investment goods firms are owned by households (or, equivalently, a representative venture capital firm that maximizes household utility) and maximize the expected present value of current and future dividends. These firms solve the same optimization problem described in Section 2.2, but with new notation since we now consider a continuum of firms $i \in [0, 1]$ optimizing given time-varying prices. These firms choose individual stocks of senior workers $s_t(i)$ and junior workers $j_t(i)$ which aggregate up to total J_t and S_t in

the intangible investment sector. They face a common price for their output, P_t^k, but the productivity shock $e_t(i)$ is now firm-specific. This means that the onboarding constraint and non-negativity constraint on J_t will bind for some firms but not others in the stochastic steady state that we linearize around.

Each firm $i \in [0,1]$ takes the price of intangible capital P_t^k, wages W_t, and interest rates r_t as given. Production of aggregate investment is $INTAN_t \equiv \int_0^1 e(i) s_{t-1}(i)^\nu di$, where idiosyncratic productivity $e_t(i)$ takes on discrete values and follows a Markov process calibrated to match a persistent AR(1) process. A firm with idiosyncratic productivity $e_t(i)$ and incumbent, senior workers $s_{t-1}(i)$ has the following value function:

$$V_t\left(e_t(i), s_{t-1}(i)\right) = \max_{j_t(i), s_t(i)} \left\{ P_t^k e_t(i) s_{t-1}(i)^\nu - W_t\left(s_{t-1}(i) + j_t(i)\right) + \frac{E_t[V_{t+1}\left(e_{t+1}(i), s_t(i)\right)]}{1+r_t} \right\}$$

where workers separate at rate d and new hires j_t become specialized at endogenous rate ρ:

$$\begin{aligned} s_t(i) &\leq (1-d) s_{t-1}(i) + \rho\left(\frac{j_t(i)}{s_{t-1}(i)}\right) j_t(i) \\ j_t(i) &\geq 0. \end{aligned}$$

Finally, in this model we "turn off" adjustment costs in the household budget constraint and set $\phi = 0$.

4.1 Calibration

For our quarterly calibration we choose standard values whenever possible. The household's discount factor is set to $\beta = .99$ and the inverse intertemporal elasticity of substitution is set to $\sigma = 2$ simply to be away from the log case $\sigma = 1$. The elasticity of labor supply is set to $\eta = 1$. The depreciation rate of intangible capital is set at the standard value used in the literature for capital of $\delta = .025$.[34] The capital share of income in the consumption sector is set to $\alpha = .3$. For nominal rigidities, we choose $\epsilon = 10$ and $\psi = 100$ to target a wage Phillips curve slope of 0.1. Finally, we set the Taylor rule parameter to be $\phi_\pi = 1.5$. Table

[34]Intangible capital like R&D depreciates much faster than this (Li and Hall, 2020). Using higher values for δ does not qualitatively change the results.

1 summarizes these choices.

For the production of intangible investment goods, we choose $\nu = .95$ implying that production is close to linear in labor s_{t-1}. There is much uncertainty surrounding this parameter, which governs the returns to scale in R&D: structural models fit to aggregate data often require lower estimates ranging from 0.3-0.5 (Moran and Queralto, 2018; Anzoategui et al., 2019; Schmöller and Spitzer, 2021), while Griliches (1990) presents cross-sectional evidence that suggests a wide range inclusive of one may be appropriate. We choose a high number to make it clear that the muted response to shocks in our model is not coming from excessive diminishing marginal returns, as low choices for ν can reduce the volatility of R&D as noted by Comin and Gertler (2006).

In Model 2 with idiosyncratic risk and congestion in onboarding, we assume e_t follows a nine-state Markov process calibrated to match a persistent AR(1) process.[35] We choose a separation rate $d = .08$ to match data on the quarterly separation rate of "Professional, Scientific & Technical Services" workers.[36]

Recall for ρ we use a linear form as described above in Section 3 calibrated to $\rho = .47 - .07\left(\frac{j_t}{s_{t-1}}\right)$. Note that this linear specification does exogenously cap the firms ability to grow at any cost, since at some point it counterfactually predicts that $\rho = 0$, and we exploit this feature during grid search in solving the firm's problem. This limitation not terribly restrictive: in our calibration, this implies optimal choices for J_t/S_{t-1} lie in $[0, \frac{b}{2a}] = [0, 5.625]$ (see Appendix B).

4.2 Results

We solve the model in sequence space to first order around this steady state with idiosyncratic risk given an exogenous path for a shock to the monetary policy rule, ϵ_t (Auclert et al., 2021). That steady state features an endogenous distribution of R&D firms, which Figure 9 plots. The distribution of firm sizes is right-skewed, despite the fact that idiosyncratic shocks are

[35] e_t follows a nine-state Markov process calibrated to discretize $X_t = .95X_{t-1} + \gamma_t$ with $\gamma \sim \mathcal{N}(0, .025)$. For the *two-state* calibration presented in Section 2, this implies a high-state productivity of 5% more than in the low state. In practice, the level of idiosyncratic risk barely matters for the aggregate model's dynamics.

[36] This value reflects the average of post-2008, aggregate data from the BLS and LEHD; separation rates were slightly higher prior to this. Using a higher value ($d = .10$) does not materially change the results.

Table 1: Calibrated Parameters Common to Both Models

Parameter	Description	Value
β	Household's discount factor	.99
σ	Inverse intertemporal elasticity of substitution	2
δ	Depreciation rate of capital	.025
α	Capital share of consumption goods sector	.3
η	Inverse Frisch labor supply elasticity	1
ϵ/ψ	Slope of the wage Phillips curve	.1

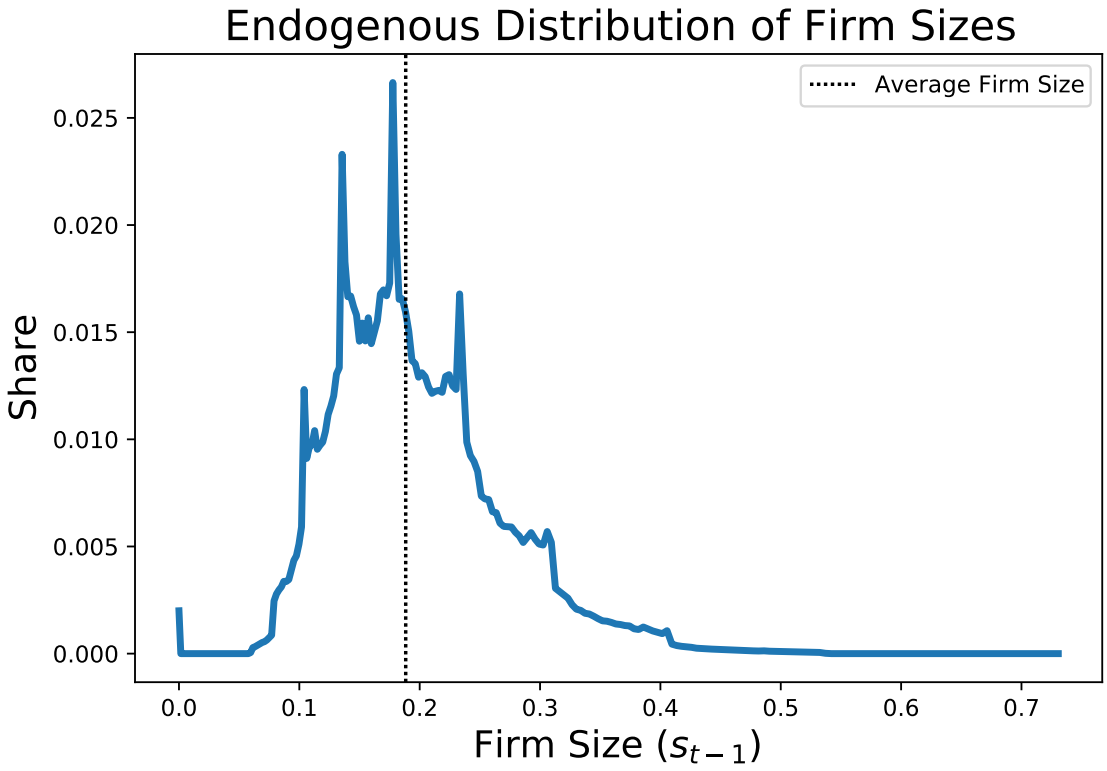
Notes: Standard parameters in the quarterly new Keynesian Model. See text for details.

symmetric, because it is harder for firms to grow than to shrink: firms scale up in the face of positive shocks more slowly than they downsize in response to negative shocks.

Figure 10 presents the congestion model's quarterly impulse responses to an contractionary monetary policy shock ϵ_t calibrated to decay at rate of 10% per quarter. These responses are the red, dotted lines in Figure 10. The shock causes consumption and inflation (not shown) to fall while the real wage slightly rises due to nominal wage rigidity. In the aggregate, intangible investment firms adjust output by firing juniors J_t, which results in a slow response of seniors S_t and their output, intangible investment $INTAN_t$. Since most workers in the intangible investment sector are senior in steady state, given our calibrated values, the aggregate labor supply response looks more like the hump-shaped response of S workers. Finally, since we are interested in the model's ability to capture the sticky and hump-shaped response of R&D in the data, which is measured at cost, we show that the wage bill of workers in the intangible investment sector ($W_t(J_t + S_{t-1})$) is also hump-shaped.

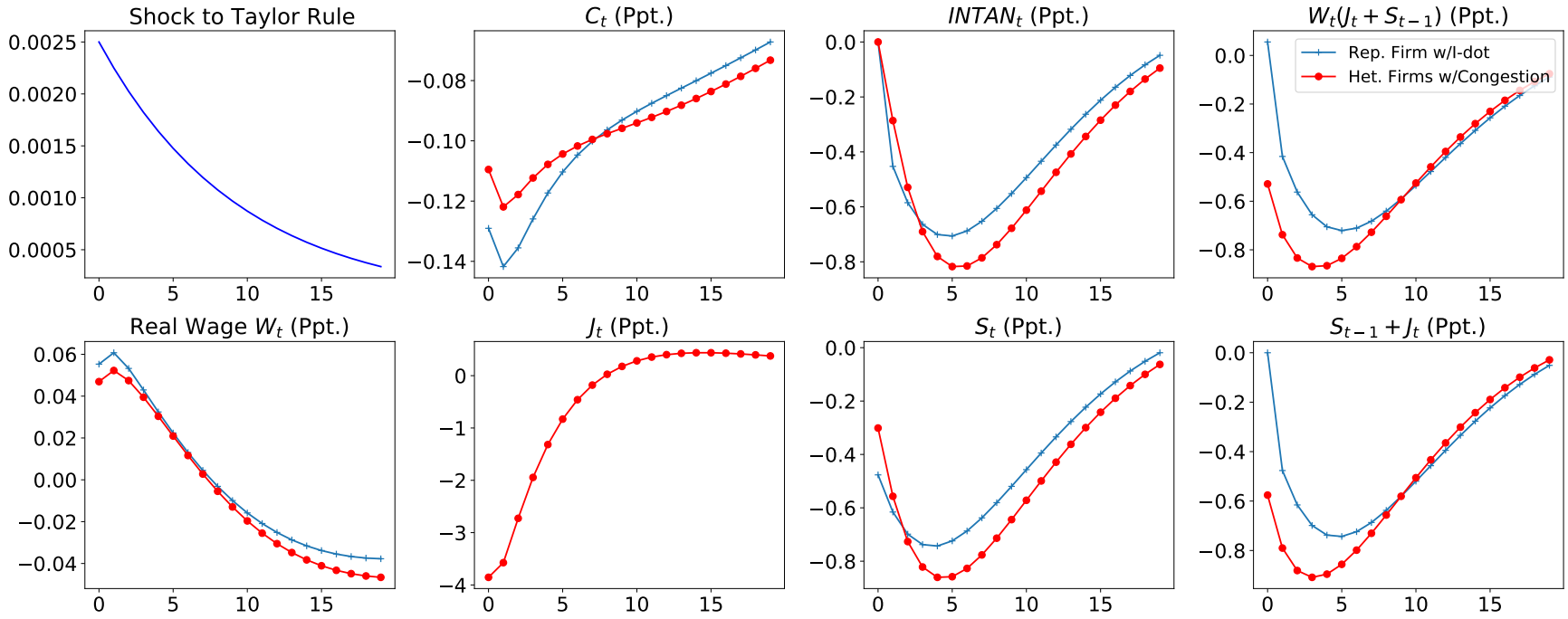
The congestion model's responses are comparable to the standard *ad hoc* model of I-dot adjustment costs, which are shown by the blue, crossed lines in Figure 10. This shows that the results in Section 2 are not reversed in a general equilibrium setting where large idiosyncratic shocks violate the assumptions made in Proposition 1. We conclude that our congestion model provides a highly plausible explanation, or microfoundations, for the investment adjustment costs used in quantitative DSGE models to capture the dynamics of R&D and other intangible investment.

Figure 9: Right-Skewed Endogenous Firm Distribution When ρ Slopes Down



Notes: Endogenous distribution of firms in the congestion model. Firms are ex-ante identical but ex-post different in size due to idiosyncratic productivity shocks which follow a nine-state Markov process. This is calibrated to match a persistent AR(1) process, and the "spikes" in the distribution are the long-run values for firms that have been in a particular productivity state for a long time (and could be "smoothed out" by adding more states). The vertical line plots the average value of S across all firms (the steady state value of S in the model). The distribution is right-skewed because scaling up in the face of positive shocks takes a long time (due to congestion) but layoffs can happen more quickly.

Figure 10: When ρ Slopes Down (Congestion), Model Responses to a Contractionary Monetary Policy Shock are Delayed and Hump Shaped as in the Standard *ad hoc* Investment Adjustment Cost Model



Notes: Quarterly impulse response functions in the model with calibrated $\rho = .47 - .07\left(\frac{j_t}{s_{t-1}}\right)$, compared to a simple representative firm model with convex investment adjustment costs. See Figure 15 in Appendix E.1 for results with a flatter ρ function, which results in less sticky investment responses.

5 Conclusion

This paper provides a microfoundation for convex adjustment costs to changing the level of R&D investment and other IPP investment, now the single largest component of U.S. investment spending. We showed formally how such costs arise naturally from congestion in onboarding new workers for firms that produce such investment goods. We then provided empirical evidence that such congestion is a significant feature of R&D and IPP production by studying the evolution of individual software developers' productivity on GitHub. Calibrating a specific functional form for our onboarding function to match this GitHub data, we embed it in an otherwise-standard dynamic stochastic general equilibrium model bereft of other real frictions. This model delivers hump-shaped responses of key macroeconomic aggregates in line with the *ad hoc* adjustment costs widely used in aggregate models. By opening up the "black box" of *ad hoc* investment adjustment costs and providing a microfoundation for them, we can confirm that the sluggish adjustment of IPP is invariant to changes in monetary policy. Thus, the common assumption that such investment is sticky *ad hoc* for structural reasons appears appropriate.

This empirical analysis supports a long-conjectured explanation for the observed stickiness in the empirical literature on R&D: that for firms which engage in knowledge production, substantial firm-specific human capital is bound up in the minds of workers and lost if the worker leaves. Firms thus behave "as if" they have high adjustment costs (Hall and Lerner, 2010; Kerr and Nanda, 2015). This paper formalizes and provides empirical evidence on this idea, illustrating how a model of congestion in acquiring firm-specific human capital can map into a model of adjustment costs in the production of investment goods.

Relatedly, note that this paper presents a theory of labor adjustment costs, which we then disciplined on rich data for workers who produce R&D and other IPP investment. Given the nature of the data, this paper focused on explaining the dynamics of such intangible investment. However, the congestion dynamics and narrative evidence presented here seem plausibly applicable to other occupations. Recent work suggests that such congestion, if a broad feature of labor markets, could well-explain the dynamics of unemployment in the aggregate (Mercan et al., 2023). Empirically investigating the extent to which the relative

prevalence of congestion and firm-specific capital could explain the relatively muted business cycle dynamics of high-skill employment represents an intriguing path for future work, which our analysis of software developers suggests is promising. Furthermore, estimating the onboarding function may help determine whether there are "diseconomies of scale" in recruiting (Manning, 2011), which is an important question in determining the distributional consequences of imperfect competition in the labor market.

References

Anzoategui, Diego, Diego Comin, Mark Gertler, and Joseba Martinez, "Endogenous Technology Adoption and R&D as Sources of Business Cycle Persistence," *American Economic Journal: Macroeconomics*, 2019, *11* (3), 67–110.

Auclert, Adrien, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub, "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models," *Econometrica*, 2021, *89* (5), 2375–2408.

BEA, "Concepts and Methods of the U.S. National Income and Product Accounts," Technical Report, Bureau of Economic Analysis, https://www.bea.gov/resources/methodologies/nipa-handbook/pdf/all-chapters.pdf December 2021.

Bianchi, Francesco, Howard Kung, and Gonzalo Morales, "Growth, Slowdowns, and Recoveries," *Journal of Monetary Economics*, 2019, *101*, 47–63.

Bloesch, Justin and Jacob Weber, "Structural Changes in Investment and the Waning Power of Monetary Policy," *Available at SSRN 3809439*, 2021.

Caplin, Andrew, Minjoon Lee, Søren Leth-Petersen, Johan Sæverud, and Matthew D Shapiro, "How Worker Productivity and Wages Grow with Tenure and Experience: The Firm Perspective," Working Paper 30342, National Bureau of Economic Research August 2022.

Casares, Miguel, "Time-to-Build, Monetary Shocks, and Aggregate Fluctuations," *Journal of Monetary Economics*, 2006, *53* (6), 1161–1176.

Christiano, Lawrence J, Martin Eichenbaum, and Charles L Evans, "Nominal Rigidities and the Dynamic Effects of a Shock to Monetary policy," *Journal of Political Economy*, 2005, *113* (1), 1–45.

_ , Martin S Eichenbaum, and Mathias Trabandt, "On DSGE models," *Journal of Economic Perspectives*, 2018, *32* (3), 113–40.

Cloyne, James, Joseba Martinez, Haroon Mumtaz, and Paolo Surico, "Short-Term Tax Cuts, Long-Term Stimulus," Technical Report, National Bureau of Economic Research 2022.

Comin, Diego and Mark Gertler, "Medium-Term Business Cycles," *American Economic Review*, 2006, *96* (3), 523–551.

Cosentino, Valerio, Javier Luis, and Jordi Cabot, "Findings from GitHub: Methods, Datasets and Limitations," in "Proceedings of the 13th International Conference on Mining Software Repositories" 2016, pp. 137–141.

Edge, Rochelle M, "Time-to-Build, Time-to-Plan, Habit-Persistence, and the Liquidity Effect," *Journal of Monetary Economics*, 2007, *54* (6), 1644–1669.

Erceg, Christopher J, Dale W Henderson, and Andrew T Levin, "Optimal Monetary Policy with Staggered Wage and Price Contracts," *Journal of Monetary Economics*, 2000, *46* (2), 281–313.

Forsgren, Nicole, Margaret-Anne Storey, Chandra Maddila, Thomas Zimmermann, Brian Houck, and Jenna Butler, "The SPACE of Developer Productivity: There's More to it than You Think," *Queue*, 2021, *19* (1), 20–48.

Gousios, Georgios, "The GHTorrent Dataset and Tool Suite," in "Proceedings of the 10th Working Conference on Mining Software Repositories" MSR '13 IEEE Press Piscataway, NJ, USA 2013, pp. 233–236.

__ **and Diomidis Spinellis**, "GHTorrent: GitHub's Data From a Firehose," in "2012 9th IEEE Working Conference on Mining Software Repositories (MSR)" IEEE 2012, pp. 12–21.

__ **, Martin Pinzger, and Arie van Deursen**, "An Exploratory Study of the Pull-Based Software Development Model," in "Proceedings of the 36th international conference on software engineering" 2014, pp. 345–355.

Griliches, Z, "Patent Statistics as Economic Indicators: A Survey," *Journal of Economic Literature*, 1990, *28* (4), 1661–1707.

Hall, Bronwyn H and Josh Lerner, "The Financing of R&D and Innovation," in "Handbook of the Economics of Innovation," Vol. 1, Elsevier, 2010, pp. 609–639.

Hamermesh, Daniel S and Gerard A Pfann, "Adjustment Costs in Factor Demand," *Journal of Economic literature*, 1996, *34* (3), 1264–1292.

Hann, I, Jeff Roberts, Sandra Slaughter, and Roy Fielding, "An Empirical Analysis of Economic Returns to Open Source Participation," *Unpublished working paper, Carnegie-Mellon University*, 2004.

Hayashi, Fumio, "Tobin's Marginal Q and average Q: A Neoclassical Interpretation," *Econometrica*, 1982, pp. 213–224.

Hertel, Guido, Sven Niedner, and Stefanie Herrmann, "Motivation of Software Developers in Open Source Projects: An Internet-Based Survey of Contributors to the Linux Kernel," *Research policy*, 2003, *32* (7), 1159–1177.

Howes, Cooper and Alice von Ende-Becker, "Monetary Policy and Intangible Investment," *Economic Review*, 2022, *107* (2).

Jaravel, Xavier, Neviana Petkova, and Alex Bell, "Team-Specific Capital and Innovation," *American Economic Review*, 2018, *108* (4-5), 1034–73.

JetBrains, "The State of Developer Ecosystem 2021," Technical Report, https://www.jetbrains.com/lp/devecosystem-2021/ 2021.

Justiniano, Alejandro, Giorgio E Primiceri, and Andrea Tambalotti, "Investment Shocks and Business Cycles," *Journal of Monetary Economics*, 2010, *57* (2), 132–145.

Kalliamvakou, Eirini, Daniela Damian, Kelly Blincoe, Leif Singer, and Daniel M German, "Open Source-Style Collaborative Development Practices in Commercial Projects using GithHub," in "2015 IEEE/ACM 37th IEEE international conference on software engineering," Vol. 1 IEEE 2015, pp. 574–585.

_ , Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M German, and Daniela Damian, "The Promises and Perils of Mining GitHub," in "Proceedings of the 11th working conference on mining software repositories" 2014, pp. 92–101.

Kerr, William R and Ramana Nanda, "Financing Innovation," *Annual Review of Financial Economics*, 2015, *7*, 445–462.

Kline, Patrick, Neviana Petkova, Heidi Williams, and Owen Zidar, "Who Profits from Patents? Rent-Sharing at Innovative Firms," *Quarterly Journal of Economics*, 2019, *134* (3), 1343–1404.

Koh, Dongya, Raül Santaeulàlia-Llopis, and Yu Zheng, "Labor Share Decline and Intellectual Property Products Capital," *Econometrica*, 2020, *88* (6), 2609–2628.

Kydland, Finn E and Edward C Prescott, "Time to Build and Aggregate Fluctuations," *Econometrica*, 1982, pp. 1345–1370.

Lakhani, Karim R and Robert G Wolf, "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects," *Open Source Software Projects (September 2003)*, 2003.

Lazear, Edward P, "Firm-Specific Human Capital: A Skill-Weights Approach," *Journal of Political Economy*, 2009, *117* (5), 914–940.

Lerner, Josh and Jean Tirole, "The Economics of Technology Sharing: Open Source and Beyond," *Journal of Economic Perspectives*, 2005, *19* (2), 99–120.

Li, Wendy CY and Bronwyn H Hall, "Depreciation of Business R&D capital," *Review of Income and Wealth*, 2020, *66* (1), 161–180.

Lucca, David, "Resuscitating Time-to-Build," *Manuscript, Federal Reserve Board*, 2007.

Manning, Alan, "Imperfect Competition in the Labor Market," in "Handbook of labor economics," Vol. 4, Elsevier, 2011, pp. 973–1041.

McDermott, Grant R and Benjamin Hansen, "Labor Reallocation and Remote Work During COVID-19: Real-time Evidence from GitHub," Technical Report, National Bureau of Economic Research 2021.

Mehran, Hamid and Joseph S Tracy, "The Effect of Employee Stock Options on the Evolution of Compensation in the 1990s," *Economic Policy Review*, 2001, *7* (3).

Mercan, Yusuf, Benjamin Schoefer, and Petr Sedláček, "A Congestion Theory of Unemployment Fluctuations," *American Economic Journal: Macroeconomics, Forthcoming*, 2023.

Mergel, Ines, "Open Collaboration in the Public Sector: The Case of Social Coding on GitHub," *Government Information Quarterly*, 2015, *32* (4), 464–472.

Mombach, Thaís Oliveira, "A Comparative Study of APIs for Querying GitHub Data," 2019.

Moran, Patrick and Albert Queralto, "Innovation, Productivity, and Monetary policy," *Journal of Monetary Economics*, 2018, *93*, 24–41.

Moris, Francisco, "Software R&D: Revised Treatment in U.S. National Accounts and Related Trends in Business R&D Expenditures," Technical Report, National Science Foundation, https://www.nsf.gov/statistics/2019/nsf19315/nsf19315.pdf April 2019.

Moylan, Carol E. and Sumiye Okubo, "The Evolving Treatment of R&D in the U.S. National Economic Accounts," Technical Report, Bureau of Economic Analysis, https://www.bea.gov/system/files/2020-04/the-evolving-treatment-of-rd-in-the-us-national-economic-accounts.pdf March 2020.

Oi, Walter Y, "Labor as a Quasi-Fixed Factor," *Journal of political economy*, 1962, *70* (6), 538–555.

Peters, Ryan H and Lucian A Taylor, "Intangible Capital and the Investment-Q Relation," *Journal of Financial Economics*, 2017, *123* (2), 251–272.

Schmöller, Michaela Elfsbacka and Martin Spitzer, "Deep Recessions, Slowing Productivity and Missing (Dis-) Inflation in the Euro Area," *European Economic Review*, 2021, *134*, 103708.

Smets, Frank and Rafael Wouters, "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 2007, *97* (3), 586–606.

Stol, Klaas-Jan, Paris Avgeriou, Muhammad Ali Babar, Yan Lucas, and Brian Fitzgerald, "Key Factors for Adopting Inner Source," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2014, *23* (2), 1–35.

Subramanian, Vikram N, "An Empirical Study of the First Contributions of Developers to Open Source Projects on GitHub," in "2020 IEEE/ACM 42nd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)" IEEE 2020, pp. 116–118.

Sun, Qi and Mindy Z Xiaolan, "Financing Intangible Capital," *Journal of Financial Economics*, 2019, *133* (3), 564–588.

Torkar, Richard, Pau Minoves, and Janina Garrigós, "Adopting Free/Libre/Open Source Software Practices, Techniques and Methods for Industrial Use," *Journal of the Association for Information Systems*, 2011, *12* (1), 1.

Vasilescu, Bogdan, Daryl Posnett, Baishakhi Ray, Mark GJ van den Brand, Alexander Serebrenik, Premkumar Devanbu, and Vladimir Filkov, "Gender and Tenure Diversity in GitHub Teams," in "Proceedings of the 33rd annual ACM conference on human factors in computing systems" 2015, pp. 3789–3798.

Wyrich, Marvin, Raoul Ghit, Tobias Haller, and Christian Müller, "Bots Don't Mind Waiting, Do They? Comparing the Interaction with Automatically and Manually Created Pull Requests," in "2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE)" IEEE 2021, pp. 6–10.

Appendices

A BEA's Treatment of Software and R&D Spending in Intellectual Property Products (IPP) Investment

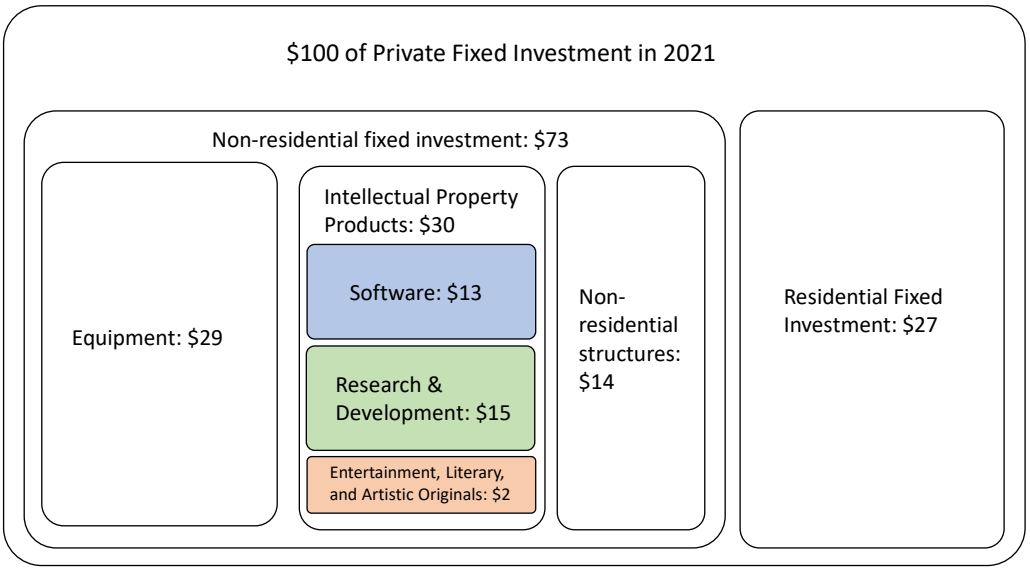
This appendix presents summary statistics illustrating the growing importance of software and R&D in US investment spending, and elaborates on the various ways software spending appears in the NIPAs.

The BEA began capitalizing expenditures on software as investment in 1999, and other R&D expenses as investment in 2013, reflecting their growing importance. These are generally measured at cost, including e.g. the wages and salaries of workers involved in development; see the NIPA handbook Ch. 6 (BEA, 2021) for details. Non-residential, fixed investment (i.e. not counting inventories) thus now consists of structures, equipment, and a new category called "Intellectual Property Products" (IPP). IPP contains both software expenditures, R&D (including software R&D), and a small share of "literary arts and originals" investment, e.g. the production of films, books, etc. The BEA's definition of intangible investment (IPP) is narrow in the sense that the NIPAs do not capitalize e.g. marketing or advertising expenses, finance and insurance costs of new product development, training costs, or organizational capital as investment; see Koh, Santaeulàlia-Llopis and Zheng (2020) for a discussion. Figure 11 illustrates this new breakdown for fixed investment quantitatively for the year 2021.[37] Ignoring the "Literary and Artistic Originals" component, which has remained stable as a share of investment, the remaining components of IPP have risen steadily as a share of US investment, as shown in Figure 12.

The "Software" category of IPP includes purchases of prepackaged software and of customized software from companies that are "primarily engaged in software development," as well as expenditures for the own-account production of new or "significantly enhanced"

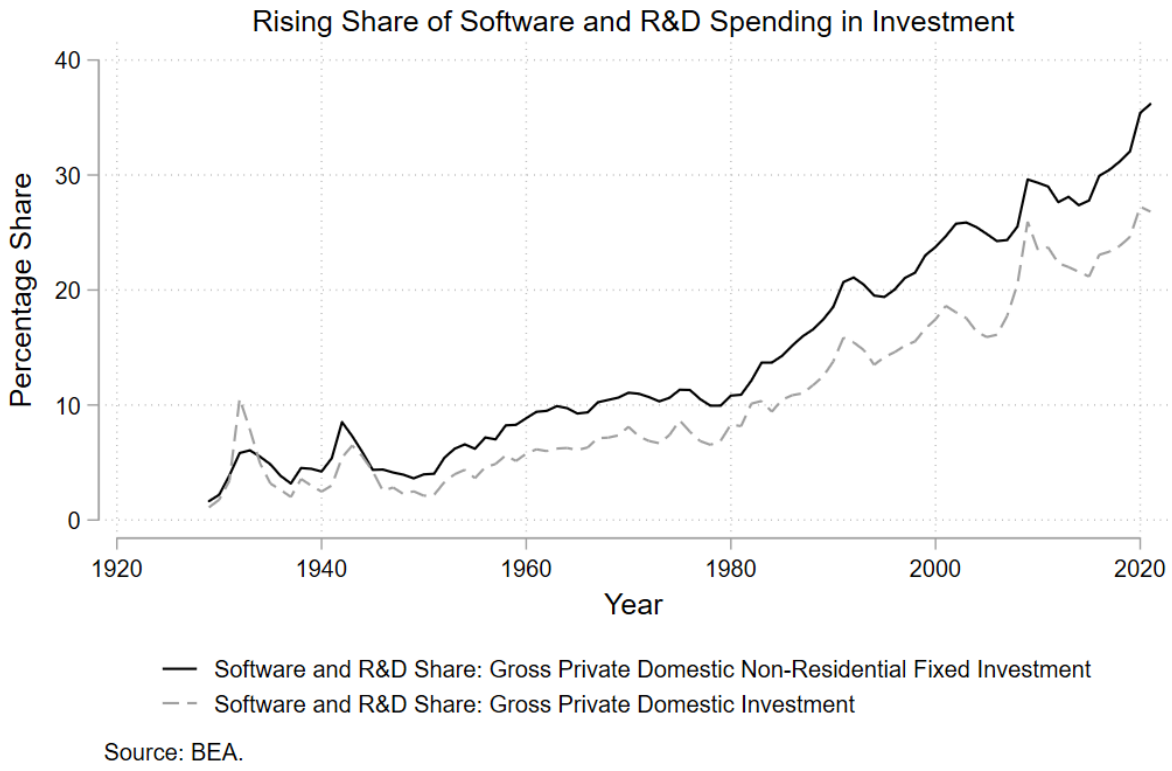
[37]This figure replicates Figure 1 in Howes and von Ende-Becker (2022) but for the year 2021 instead of 2020. Note also that their Figure 1's exact dollar amounts reflect outdated GDP statistics: as of the September 29th, 2022 revision IPP was larger than equipment in 2020 as claimed in the introduction (see NIPA table 1.1.5).

Figure 11: The Components of US Fixed Investment



Notes: the components of fixed investment (i.e. excluding inventories) in the US national accounts. Note that a large share of R&D is software R&D. Source: BEA and Authors' calculations.

Figure 12: Secular Rise in Software and R&D Investment



Notes: Software and R&D – the two largest components of IPP investment – have risen steadily as a share of US investment. The excluded category "Literary and Artistic Originals" is a small share of U.S. investment and has been stable over time. Source: BEA.

software that a firm develops in-house.[38] Own-account software does not include the development of software originals from which copies are made for sale (i.e. product development) or incorporated into other products (such as vehicles or appliances); these expenses are instead included in the R&D category of IPP (BEA, 2021) reflecting recent changes in 2018 (Moylan and Okubo, 2020). Roughly 1/3 of R&D is software R&D (32.2% in 2016), reflecting a secular increase over the past two decades (Moris, 2019).

While the BEA does not currently publish the components of R&D separately by type, underlying NSF survey data permits separating software R&D from other kinds of R&D for specific years. Table 2 presents a breakdown of IPP for 2016 using data from the NSF in Moris (2019) and the BEA to break out software R&D from other R&D, showing that software expenditures make up a majority of IPP.

Table 2: Composition of Non-Residential Investment in 2016

Category	**Investment Share (Ppt.)**
Software R&D and Other Software Investment	18.8
Software R&D	5.3
Other Software Investment	13.6
Non-software R&D	11.1
Literary and Artistic Originals	3.3
Equipment and Structures	66.7

Notes: Shares of US Gross Private Domestic Fixed, Non-residential Investment. Source: authors' calculations from BEA data and NSF data in Moris (2019).

B Proof of Proposition 1 and Discussion

Statement: Consider the problem of a firm choosing paths $\{I_{t+1}, J_t, S_t\}_{t=0}^{\infty}$ subject to the law of motion (1) and the production function $I_t = S_{t-1}^{\nu}$ to maximize the present discounted value of current and future profits (2). In a solution where (1) binds and $J_t > 0$ always, then

[38] Prepackaged software excludes software embedded, or bundled, in computers and other equipment (Moris, 2019).

the firm's problem can be written as:

$$\max_{\{I_{t+1},J_t,S_t\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t(S_{t-1}+J_t)\right]\right]$$

subject to

$$\begin{aligned} I_t &= S_{t-1}^{\nu} \\ S_t &= (1-d)S_{t-1} + \rho\left(\frac{J_t}{S_{t-1}}\right)J_t \end{aligned}$$

where $\rho(x) \in [0,1]$ on $x \in [0,\infty)$ and $\rho'(x) < 0$. Let $f(x) \equiv \rho(x)x$ be strictly increasing on some domain D that does not restrict the firm's optimal choice. Then there exists an equivalent maximization problem yielding the same solution for I_t:

$$\max_{\{I_{t+1}\}_{t=0}^{\infty}} \mathbf{E}\left[\sum_{t=0}^{\infty}\Lambda_{0,t}\left[P_t^k I_t - W_t\left(1 + \underbrace{\Phi\left(\frac{I_{t+1}}{I_t}\right)}_{\substack{\text{Convex Adjustment Costs} \\ \text{from Onboarding}}}\right)I_t^{\frac{1}{\nu}}\right]\right]$$

and a domain G which does not restrict firm's optimal choice and where $\Phi' > 0$ on G. Further, if $f''(x) < 0$ on D then $\Phi'' > 0$ on G.

Proof: Since the law of motion (1) from the main text binds and $J_t > 0$ by assumption (so that the complementary slackness condition on this constraint can be ignored), rewrite the binding law of motion for S_t, equation (1), as:

$$\frac{S_t}{S_{t-1}} = (1-d) + \rho\left(\frac{J_t}{S_{t-1}}\right)\frac{J_t}{S_{t-1}} \equiv (1-d) + f\left(\frac{J_t}{S_{t-1}}\right) \tag{9}$$

Assume the function $f\left(\frac{J_t}{S_{t-1}}\right) = \rho\left(\frac{J_t}{S_{t-1}}\right)\frac{J_t}{S_{t-1}}$ is strictly increasing (and therefore invertible) and concave in $\frac{J_t}{S_{t-1}}$ on D. Note D is a subset of $[0,\infty)$ since $J_t \geq 0$ implies $J_t/S_{t-1} \geq 0$. Then (9) implies that $\frac{S_t}{S_{t-1}}$ is a concave, strictly increasing function of the term $(1-d)$ and the ratio $\frac{J_t}{S_{t-1}}$. Define this function as $F(\frac{J_t}{S_{t-1}})$, suppressing dependence on d, such that

$F^{-1}(\frac{S_t}{S_{t-1}}) = \frac{J_t}{S_{t-1}}$ is the inverse of $F(\cdot)$.[39] Then F^{-1} is convex and strictly increasing on $G \equiv F(x) \forall x \in D$. Pulling an S_{t-1} out of the final term in per-period profits in (2) and plugging this in for the resulting J_t/S_{t-1} term yields:

$$P_t^k I_t - W_t \left(S_{t-1} + F^{-1}\left(\frac{S_t}{S_{t-1}}\right) S_{t-1} \right)$$

Now note since $\frac{S_t}{S_{t-1}}$ is an increasing, convex function of $\frac{I_{t+1}}{I_t}$, i.e. $\frac{S_t}{S_{t-1}} = (\frac{I_{t+1}}{I_t})^{\frac{1}{\nu}}$, it follows that F^{-1} is an increasing, convex function of $\frac{I_{t+1}}{I_t}$. Substituting in, we obtain the following for profits in period t:

$$P_t^k I_t - W_t \left(1 + \underbrace{\Phi\left(\frac{I_{t+1}}{I_t}\right)}_{\substack{\text{Convex Adjustment Costs} \\ \text{from Onboarding}}} \right) I_t^{\frac{1}{\nu}}$$

which yields the result.

Discussion: for $\Phi(\cdot)$ to be increasing and convex on G, we need $f(x) = \rho(x)x$ to be increasing and concave on D. Neither follows easily from the assumptions $\rho'(x) < 0$ and $\rho(x) \in [0, 1]$. To see this consider the expressions for f' and f'' in terms of ρ,

$$f'(x) = \rho'(x)x + \rho(x)$$
$$f''(x) = \rho''(x)x + 2\rho'(x)$$

Note that for x small enough, we can always find a neighborhood where $f'(x) > 0$ and $f''(x) < 0$ under the assumption that $\rho'(x) < 0$ *and the additional assumptions* that $\rho'(x)$ and $\rho''(x)$ are bounded as $x \to 0$, since under these added assumptions

$$\lim_{x \to 0} f'(x) = \rho(0) > 0$$
$$\lim_{x \to 0} f''(x) = 2\rho'(0) < 0$$

[39] This function was defined as $F^{-1}(\cdot) \equiv \boldsymbol{\mathcal{F}}(\cdot)$ in the text's Section 2.

The neighborhood with $x \approx 0$ is of interest because this corresponds to a steady state of the model: when $f(x) = d$, the model is in steady state where $S_t/S_{t-1} = 1$. When d is small, x may also be small (or zero if $d = 0$). So we can have local concavity of f (and convexity of Φ) under very mild assumptions about the boundedness of the first and second derivative of ρ at zero. This is relevant since many aggregate models log-linearize around a steady state, and only require that the adjustment cost function be convex when evaluated at that point. Of course when d is large, the model's steady state may be far from $x \approx 0$. Thus, we may remain concerned that our assumptions on f may not hold for large x.

To alleviate these concerns, we note that the requirement that $f'(x) > 0$ and $f''(x) < 0$ does not put overly restrictive requirements on ρ in light of our empirical results. In practice, those results suggest that ρ:

- is a function of J_t/S_{t-1}
- satisfies $\rho(x) \in [0, 1]$ on $x \in [0, \infty)$ and $\rho'(x) < 0$
- is convex, i.e. $\rho''(x) > 0$.

One function that satisfies all these properties is $\rho(x) \equiv \frac{1}{ax+b} + c$, given appropriate choices of $a, c \geq 0$ and $b > 0$ so that $\rho(0)$ is well-defined. To see this, note ρ is decreasing and convex:

$$\rho'(x) = -a\left(\frac{1}{ax+b}\right)^2 < 0 \quad \forall x \geq 0$$
$$\rho''(x) = 2a^2\left(\frac{1}{ax+b}\right)^3 > 0 \quad \forall x \geq 0$$

while f is increasing and concave:

$$\begin{aligned}
f'(x) &= \rho'(x)x + \rho(x) \\
&= -a\left(\frac{1}{ax+b}\right)^2 x + \frac{1}{ax+b} + c \\
&= \left(\frac{1}{ax+b}\right)\left(\frac{-ax}{ax+b} + 1\right) + c > 0 \quad \forall x \geq 0 \\
f''(x) &= \rho''(x)x + 2\rho'(x) \\
&= 2a^2\left(\frac{1}{ax+b}\right)^3 x - 2a\left(\frac{1}{ax+b}\right)^2 \\
&= 2a\left(\frac{1}{ax+b}\right)^2\left(\frac{ax}{ax+b} - 1\right) < 0 \quad \forall x \geq 0.
\end{aligned}$$

Additionally, a linear function or linear approximation will also work: if $\rho(x) = b - ax$ then $f(x) = bx - ax^2$ is quadratic, and no cost-minimizing firm will every choose a point where $x > \frac{b}{2a}$. So for the domain $x \in D \equiv [0, \frac{b}{2a})$ which does not restrict the choices of a cost-minimizing firm:[40]

$$\begin{aligned}
f'(x) &= b - 2ax > 0 \quad \forall x \in \left[0, \frac{b}{2a}\right) \\
f''(x) &= -2a < 0 \qquad \forall x.
\end{aligned}$$

C Narrative Evidence on "Project-Specific Capital"

The returns to project-specific tenure that we document reflects a combination of skill-acquisition and earned trust or reputation within a team, which our model in Section 2 is general enough to encompass. We emphasize the acquisition of project-specific skills, as this frequently arises in interviews with practitioners. While human capital acquired while working on a specific project may in principle be portable (and imperfectly captured by our controls for overall programming experience) in general the evidence that knowledge

[40] There are two cases for any $x > 0$ not in D. For any choice $x \in [\frac{b}{2a}, \frac{b}{a}]$, there is a choice $x \in D$ that weakly dominates because it achieves the same growth in S_t at a smaller cost. Choosing very large $x > \frac{b}{a}$ means paying $W_t J_t$ to *reduce* the stock of S_t given S_{t-1} – assuming that the law of motion for S binds rules out the possibility that this is optimal (in our quantitative exercise, we allow for "free disposal" of S which guarantees firms would never choose $x > \frac{b}{a}$).

gained by working on OSS projects is applicable elsewhere is weak: in a longitudinal study of contributors to the Apache project, Hann et al. (2004) found that increases in human capital as measured by total contributions to the project, did not lead to increased wages.[41]

What is this non-portable, "project-specific" human capital acquired in the first few months? Even for experienced developers, joining a new project or a new team entails acquiring knowledge specific to how that team operates and how existing code is structured ("software architecture"). Interviews with developers reveal that project-specific knowledge such as learning about the needs and requirements of end users, the "dos and don'ts" of design for a particular project or company, and – for tacit knowledge – "knowing who knows what" all plays a role in making a newcomer productive on a software development team (Stol et al., 2014).

Consistent with this, early studies recommending the adoption of OSS software development practices within private firms highlighted the advantages of adapting OSS development methods for private industry because of their ability to reduce onboarding times: these methods include making the entire history of design decisions and code base accessible to newcomers, and also to assign them relatively easier tasks that they can use to build skills and demonstrate their newly-acquired competence. As noted by Torkar et al. (2011),

> "It is important to have a predefined path that allows new developers to learn while doing productive activities... If this issue is left unattended, there is a risk of placing newcomers in positions for which they are unqualified or making their learning curve unnecessarily long. With proper support from experienced developers, bug fixing and technical debt reducing activities are a good entry point for new developers. Such tasks allow new developers to familiarize themselves with the software architecture... Following this strategy, they would be ready to be incorporated sooner in regular development project activities. Additionally, resourceful developers would have a greater chance to stand out sooner, reducing employee frustration..."

[41] The authors took advantage of the Apache project's unusual hierarchy, which includes five rankings, to show that instead these earned credentials explained wage growth, which they interpret as consistent with a signaling theory of the benefits to contributing to an OSS project.

Moreover, having all changes and discussions publicly logged, as GitHub and the Merge/Pull model enable, would both serve to improve onboarding and mitigate the damage done when senior workers left:

> "This archive [of past design and implementation decisions] would form a useful knowledge base that can be used to lower the learning curve for newcomers and ground further decision-making for experienced developers. Moreover, this knowledge would be permanent and independent of key employees leaving a project."

Torkar et al. (2011) contrasted these OSS practices with existing practices at Ericcson, a global telecommunications company, which had typical problems acclimating new workers to their in-house software development methodology, "Streamline:" it took 38% of newcomers over a month just to acclimate to the in-house methodology, and a majority never graduated from the initial "software testing" tasks that Ericsson commonly assigned to newcomers as part of the onboarding process.[42]

OSS software projects also use simple initial tasks to build and assess competency, as Torkar et al. (2011) pointed out. In the words of one developer and OSS project founder surveyed and quoted by Kalliamvakou et al. (2015):

> "Even if you are not sure if the other dev[eloper] is capable of contributing good code, you can review pull requests and if the fifth pull request is good you give him/her commit bit [the power to make direct changes in the OSS repository, i.e. merge others' pull requests]."

Consistent with this, empirical studies find that initial OSS contributions are often more trivial tasks (Subramanian, 2020) and that a developer's track record with a project is the single most important predictor for time-to-merge in OSS projects on GitHub (Gousios et al., 2014).[43] In short, both OSS projects and the private sector use simple initial tasks to build

[42]Ericsson was not alone in this: in a series of workshops, Torkar et al. (2011) asked representatives from three large, multinational software companies to rank a list of their suggested benefits of adopting OSS methods by desirability, and found that both "Define an entry path for newcomers" and "Increase information availability and visibility" were consistently prioritized as the most important potential benefits.

[43]Secondary factors were project-specific or measures of pull request size and complexity. Gousios et al.

competence and evaluate the performance of newcomers, with private industry adopting OSS development practices in no small part because they facilitated this process. The increase in productivity we document, via a decrease in approval times, reflects this process by which juniors take on tasks, learn from seniors during code review, and eventually both write better code (which is merged faster) and build trust, allowing them to take on more serious tasks and have their changes merged with less scrutiny.

All this suggests that attention from seniors, both for education and evaluation, is critical for onboarding juniors.

D GitHub Data Appendix

This section describes how we worked with GHTorrent data to create the datasets necessary to estimate the main text's regressions. Before doing so, we first discuss the structure of GitHub's data, which Gousios (2013) accesses through GitHub's API to build GHTorrent.

In GitHub, every repository can be concieved of as a collection of events on a timeline that keeps track of changes to the repository's files and the discussion surrounding those changes. Figure 13 presents an example of workflow on a GitHub project being developed using pull requests. Each grey circle represents a discrete event, or action, taken on the part of some user. These events are timestamped and stored as JSON files with associated characteristics that depend on the action in question; see Gousios and Spinellis (2012) for a more technical description. These files can be accessed directly through GitHub's API, subject to request limits, and some researchers, studying a small number of repositories, obtain data from GitHub's API directly.

To obtain data on all public repositories, GHTorrent surmounts the request limit by crowdsourcing API keys from multiple donors and processes the raw, timestamped event data into a MySQL database; see Figure 1 in Gousios (2013) for the database schema. The result is a "snapshot" of the timeline for all public repositories on GitHub, taken at different

(2014) also investigate the determinants of pull request *acceptance*, finding that almost all pull requests are eventually merged and suggest the number may be as high as 90% once one corrects for merges occurring outside of GitHub. They report that the single most relevant factor for eventual merging is whether the files touched have been modified recently (i.e. are relevant to ongoing development). We thus do not consider this as an outcome variable for the purposes to determining whether and when a developer achieves competency.

points in time.[44] Various snapshots are currently publically available on Google BigQuery, which allows a user to query GHTorrent's MySQL dataset using SQL. The main text uses the 2019Q2 snapshot, and Appendix E.4 replicates our results on the 2016Q3 snapshot as a robustness check.[45] This broad coverage, ease of access, and preprocessing explains the popularity of GHTorrent with researchers (Cosentino et al., 2016).[46]

The first step in processing the GHTorrent data is to merge data on pull request comments and actions (open, merge, etc.) to create a table of pull request activity on each repository. We then drop all repositories that have less than a specified number of "merge" events: 100 in the main text.[47] This effectively keeps repositories that are actually using GitHub to jointly develop code; as discussed in the main text, repositories that are actually personal projects or websites generally do not use pull requests, and some repositories on GitHub are "mirrors" of projects actually being developed elsewhere, and thus may have only open and closed pull requests, but no merge events.[48] Effectively, by conditioning on projects with many merge events, we select only projects that are being *developed* on GitHub.

We then drop all events initiated by bots using regex filters on logins, following Wyrich et al. (2021). We also drop events initiated by organizational accounts (that stand in for groups of users) and events for accounts that can't be linked to a user ("fake" accounts in GHTorrent parlance; these are real users who have not configured a GitHub account and whose commit activity is tracked via email address in GHTorrent). Finally, we drop events where the user is simply missing. These are minor issues. Collectively, this means dropping only 2.8% of the PR activity data, and the vast majority of this is "bot" activity (2.1% of

[44]For example, the GHTorrent sample, or "snapshot" used in the main text of this paper is from 2019, so that a project that began in 2008 but was deleted in 2018 would not appear in that dataset. However, the project would appear in the 2016 sample used in Appendix E.4, and we could see the timeline of events from 2008 to 2016 there.

[45]For this project, we accessed the GHTorrent dataset through BigQuery's API in Python. To access the 2016 snapshot, the project id is "ghtorrent-bq" and the 2016 vintage has the dataset id "ght". To access the 2019 snapshot, the dataset id is "ghtorrentmysql1906" and the project id is "MySQL1906". For more documentation, see `https://github.com/ghtorrent/ghtorrent.org`.

[46]GH Archive (`https://www.gharchive.org/`) also records this public GitHub timeline data. For a comparison of the costs and benefits of various methods of accessing GitHub's data, see Mombach (2019).

[47]An early version of this paper used 120 merge events as the threshold, which corresponded to keeping exactly 25% of all pull requests from the largest projects in the 2016 GHTorrent snapshot. We have tried using both 120 and 200 merge events as the threshold in the 2019 snapshot and verified that the results in the main text and appendices are not sensitive to the precise choice.

[48]Missing merge *events* in GitHub for pull requests that are actually merged is very common—see Gousios et al. (2014) for a discussion.

all activity) .

There are some errors in GHTorrent: for example, we also drop duplicate events. However, having made all of the above cuts, there are a few instances of multiple recorded "open" or "merge" events for the same pull request at different times. We resolve this by keeping the earliest open event and last merge event, but these constitute a tiny fraction of the data (a few hundred events collectively in a final dataset of over 65 million events). As a last step, we remove some "test" repositories that have the word "test" in the repository name and an average time to merge measured in seconds (less than one minute) though this only removes 16 repositories.

Having made these cuts, we are left with a dataset of over 65 million pull request events on 36,537 large repositories that we consider joint attempts to develop code, which we now call *projects*.[49] We can use this dataset in turn to create two different panel datasets that are used to produce the figures in the main text.

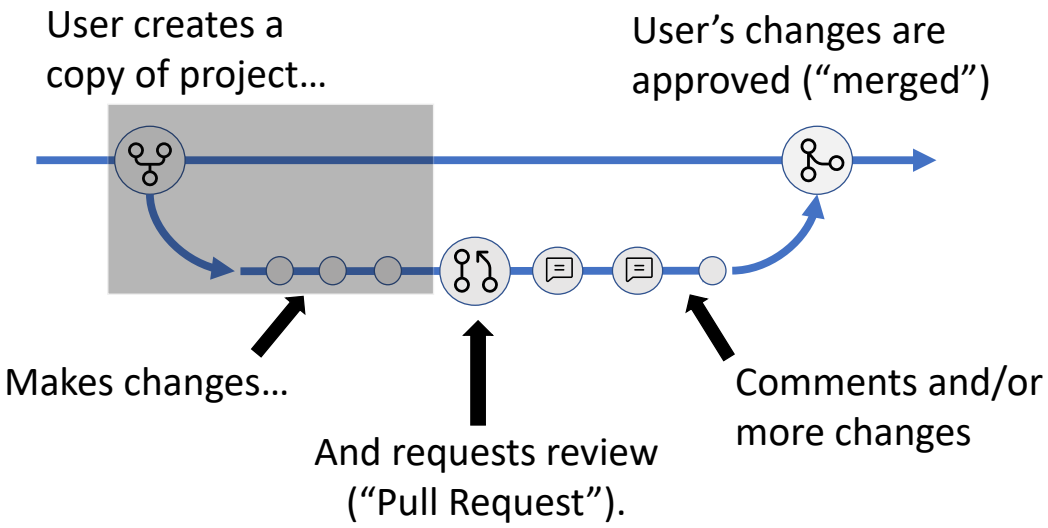
First, we build a panel dataset on individual contributions (merged pull requests) from the events in this dataset. The goal is to estimate equation (4) in the main text, reproduced here: letting $y_{i,p,t}$ be either the approval time or total comments received for a contribution opened by user i on project p at time t, we estimate the following via OLS:

$$\begin{aligned} y_{i,p,t} = &\sum_{j=1}^{13} D(\text{Months Project Experience} = j)_{i,p,t} \\ &+ \sum_{k} D(\text{Months Programming Experience} = k)_{i,t} \\ &+ D_{i,p} + \beta_{PA,p} ProjectAge_{p,t} + \epsilon_{i,p,t}. \end{aligned}$$

The first sum consists of dummy variables for having between one and thirteen or more months of experience on project p at time t, and the second sum consists of dummy variables for overall programming experience measured by GitHub account age at time t, which is capped at 49 months, so that the dummy for having exactly 49 months includes all individuals with more than four years (48 months) of experience on GitHub measured using account

[49] We thus treat forked repositories as separate projects, if they have enough merge events; see Section 3.1.

Figure 13: Workflow on GitHub: Simplified Example



Notes: A GitHub repository can be concieved of as a collection of events on a timeline tracking changes to the repository's files and the discussion surrounding those changes. Each grey circle represents a discrete event, or action, taken by some user. These events are timestamped and stored as JSON files with additional information depending on the type of event. In this example, a user creates a copy of a repository, and then makes changes (grouped into individual "commit" events by the user) to the copy before opening a pull request to have their changes merged into the original. The grey box over this part of the workflow reflects that we may not observe this part of the contribution process: if the repository is copied as a new repository on GitHub, for example, then we will observe these changes. But if the project is e.g. downloaded by the user and then changed locally, we may not be able to observe this activity.

age.[50] We also allow for individual-by-project fixed effects ($D_{i,p}$) and project-specific linear time trends ($\beta_{PA,p}ProjectAge_{p,t}$). Table 3 presents example of this dataset, with fictitious data, used to estimate this regression. Recall that Figure 5 in the main text uses the marginal effects estimated from equation (4) to show that approval time and the number of comments recieve fall precipitously in the first six months of project-specific experience. This provides evidence of a nontrivial onboarding period for juniors, as discussed in the main text.

Table 3: Panel Data on Individual Contributions (Example)

n	Opened t	Contributor i	Project p	Project Experience	Approval Time	Total Comments
1	1/1/15	Jake	Project A	0 days	6 days	4
2	1/9/15	Jake	Project A	8 days	5 days	2
3	1/9/15	Jake	Project B	0 days	17 days	5
4	1/12/15	Jake	Project B	3 days	15 days	3
5	1/12/15	Justin	Project B	0 days	1 days	0

Notes: An illustrative example of the panel dataset used to estimate regression (4) in the main text. The actual dataset has N = 10,881,355 observations (approved contributions) on 36,537 large projects (with at least 100 approved contributions).

Next, we build a panel dataset of juniors joining new projects that is used to estimate equation (5) in the main text, yielding the estimate for ρ plotted in Figure 8. We begin by identifying junior J type and senior S type workers using their pull request activity. In each calendar month t and each project p, we assign each user with activity on at least one pull request in p at t into either category J or category S. We drop users who never contribute, and restrict attention to those who open at least one pull request that is merged. A J type transitions to an S type on a particular project either when they have reached a tenure of six months on that project, or when we observe them reviewing code (i.e., when we observe them merging/closing/commenting on pull requests opened by others). Project tenure is measured as the length of time between a user's first observed activity and their last observed activity on a project. This definition implies that some workers are S types from their first month on a project. Also note that once a J type worker transitions on a project, they are counted as an S type in any calendar month when they "re-appear" on

[50] A small number of user accounts and projects have pull request activity predating the reported date of account or project creation. We correct these using the time of earliest observed pull request activity.

that project in the pull request data.

We define the quantity of J types on project p at time t as $J_{p,t}$, tabulated as the number of users who have contributed to that project (i.e. authored at least one pull request that was eventually merged) at time t with less than six months of tenure and who have not reviewed code written by others (i.e. who have not been observed merging/closing/commenting on a pull request opened by someone else). The other active users are summed into $S_{p,t}$. Table 4 presents an example of this dataset, with fictitious data, used to estimate equation (5), reproduced here: letting $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards})$ denote an indicator function for whether a Junior i on project p (counted in the sum $J_{p,t}$) will eventually transition to being an S type on project p, we estimate:

$$\begin{aligned}\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) &= \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } b\right) \\ &+ D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}.\end{aligned}$$

We estimate the effect of $J_{p,t}/S_{p,t}$ non-parametrically via OLS by measuring it as a set of dummy variables representing equidistant bins for junior-senior ratios. Project specific dummies D_p control for unobservable project-specific features that may make some projects easier to join, while $ProjectAge$ is a project-specific time trend meant to capture the project life cycle, since some projects may become harder to join as they age; X_t are year fixed effects, and $\gamma_{i,t}$ captures Junior-specific controls, which include fixed effects for the year user i's account was created, and dummies for the age of user i's account.

We cannot include user-project specific fixed effects here because they are collinear with the outcome variable (we only observe one outcome per project for each individual: either they onboard, or they do not). Relatedly, we cannot well-estimate individual fixed effects because in practice most individuals join very few OSS projects in sample over time. Note if someone joins only one project in our sample of large OSS projects, we cannot estimate a fixed effect for them. Appendix E.2 discusses this and shows that our results are qualitatively unchanged by adding individual fixed effects, though the sample size shrinks.

We linearly approximate the non-parametric estimates of congestion in Figure 8 using OLS for use in calibrating our DSGE model. This is the blue line in that figure. Note that

the precise slope is not too sensitive to the choice of threshold for the number of pull request merge events we include in our sample: across the thresholds of 100, 120, and 200 that we have explored, the slope of the line varies from -0.064, -0.067, and -0.075, respectively while the intercept is stable at 0.47; we thus pick a slope of -0.07 for calibrating our DSGE model.

Table 4: Panel Data on Juniors Joining Projects (Example)

n	Month Joined t	Junior i	Project p	Cohort Size $J_{p,t}$	$S_{p,t}$	$\frac{J_{p,t}}{S_{p,t}}$	Onboards?
1	1/2015	Jake	Project A	2	4	.5	Y
2	1/2015	Justin	Project A	2	4	.5	N
3	1/2015	Justin	Project B	1	10	.1	Y

Notes: An illustrative example of the panel dataset used to estimate regression (5) in the main text. The actual dataset has N =1,044,039 observations, one for every junior on each project joined.

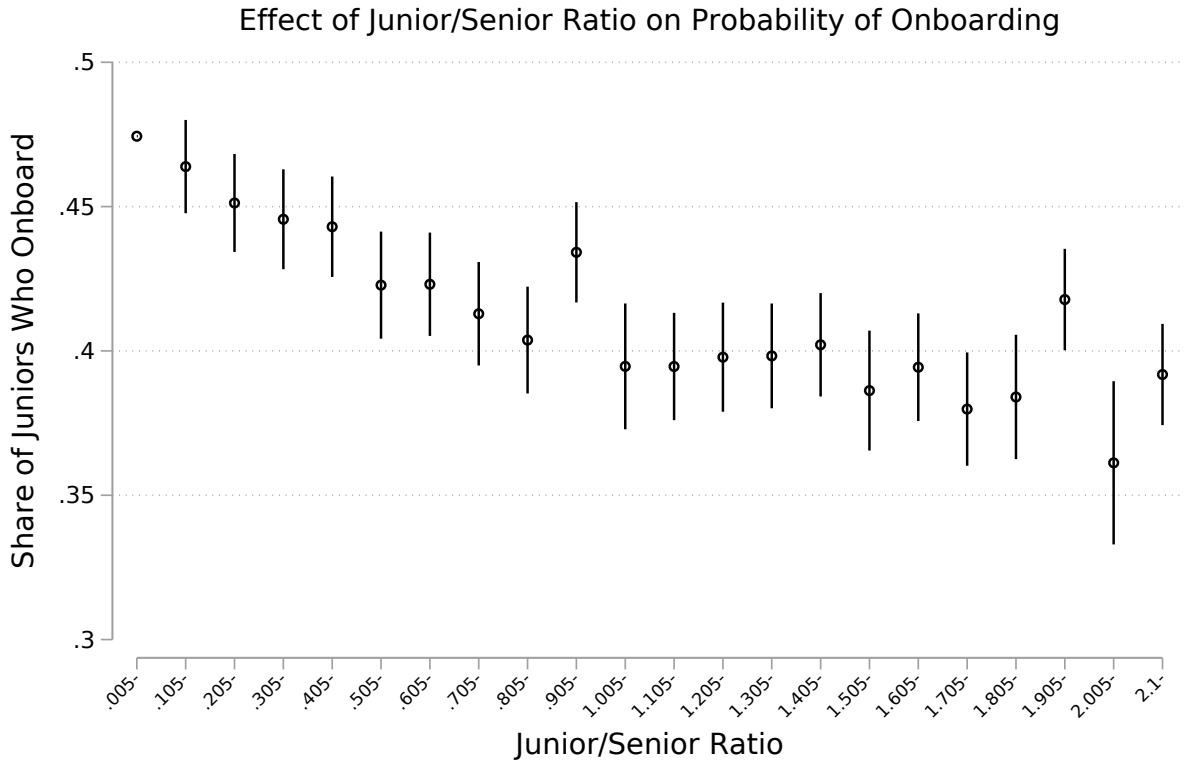
E Robustness of Empirical Analysis with GitHub Data

E.1 Congestion Results with More Bins

Figure 14 extends the congestion analysis of Figure 8 to allow for J/S ratios greater than 2. This results in a slightly flatter calibration for our linear onboarding function, chosen to approximate the nonlinear relationship apparent in Figure 14: $\rho = .45 - .04\left(\frac{j_t}{s_{t-1}}\right)$. Figure 15 replicates the IRFs in Figure 10 using this new, flatter calibration to show that this implies slightly less stickiness relative to the benchmark model with convex investment adjustment costs. Put another way, the results are not terribly sensitive to the ρ calibration, provided there is sufficiently negative slope.

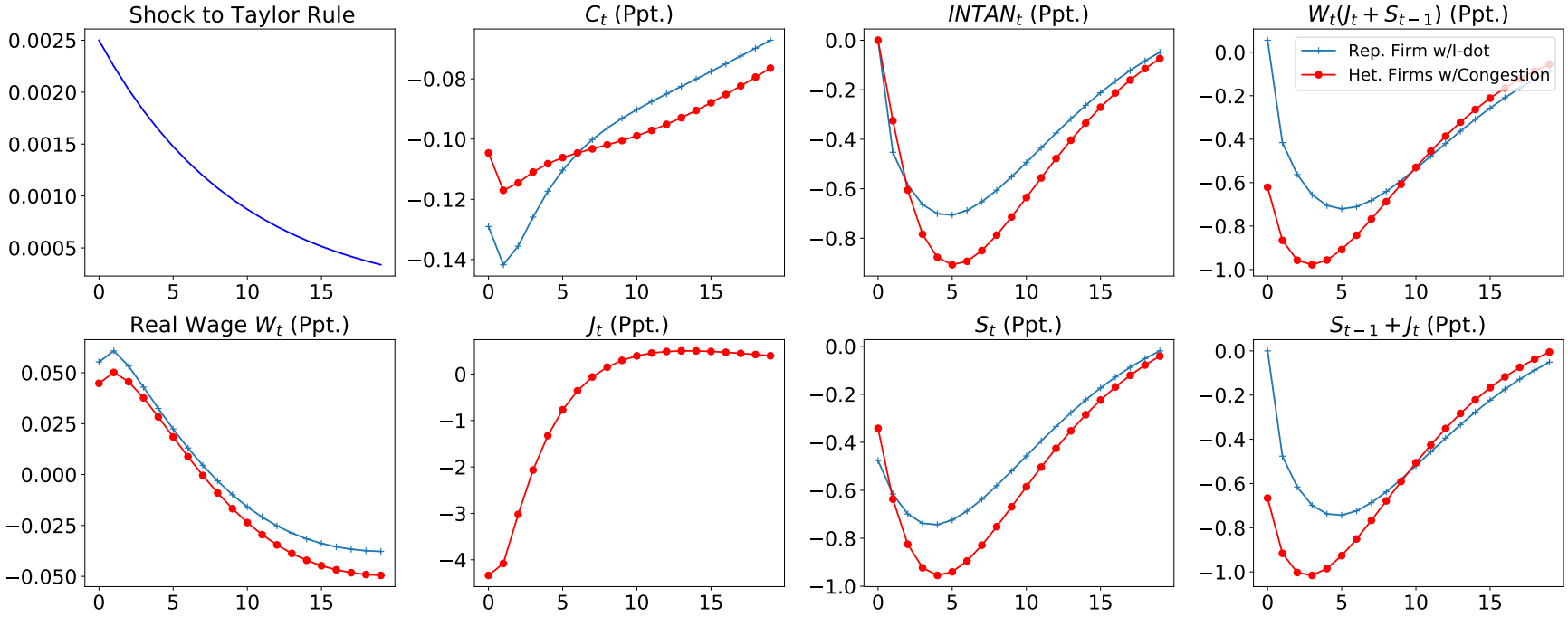
We prefer our headline calibration because we view it as a local linear approximation of the nonlinear function over a space which is more relevant for optimizing firms in our model: the steady-state ratio of J/S is generally much, much less than one, reflecting the fact that at any given point in time most workers are already onboarded in the investment sector.

Figure 14: Non-parametric Estimate of the Onboarding Function ρ



Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin} = b\right) + D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The "spikes" in bins containing $J/S = 1$ or $J/S = 2$ reflects the fact that being "one-on-one" or "two-on-one" with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$ and over 90% have $J/S \leq 2$.

Figure 15: Model Responses to a Contractionary Monetary Policy Shock: Flatter ρ

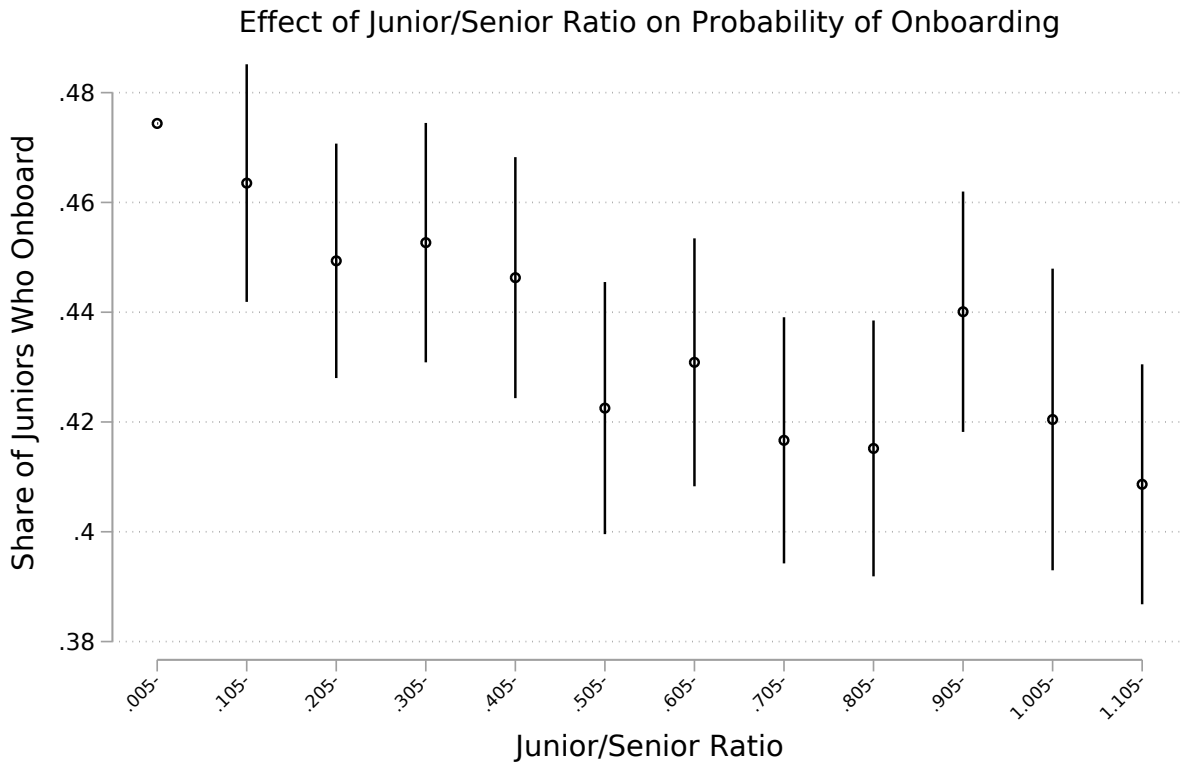


Notes: Quarterly impulse response functions in the model with flatter ρ calibrated to capture the nonlinear relationship in Figure 14: $\rho = .45 - .04\left(\frac{j_t}{s_{t-1}}\right)$, compared to a simple representative firm model with convex investment adjustment costs. All other parameters are the same as described in Section 4.

E.2 Adding Individual Fixed Effects to the Congestion Regression

Adding individual fixed effects to the congestion regressions in the main text shrinks the sample size by about a third. This is because we can only identify individual fixed effects for contributors who successfully merge pull requests on multiple large open source projects (recall we only consider projects with many pull requests; see Section 3). Including fixed effects drops individuals who e.g. only contribute and join one major open source project. We include fixed effects here and note that in spite of the diminished sample size (N = 646,843 as opposed to N =1,044,39 in the headline results in Figure 8), the results are qualitatively similar. See Figure 16.

Figure 16: Non-parametric Estimate of the Onboarding Function ρ with Individual Contributor Fixed Effects



Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin} = b\right) + D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_i + \epsilon_{i,p,t}$. The "spikes" in bins containing $J/S = 1$ or $J/S = 2$ reflects the fact that being "one-on-one" or "two-on-one" with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$ and over 90% have $J/S \leq 2$.

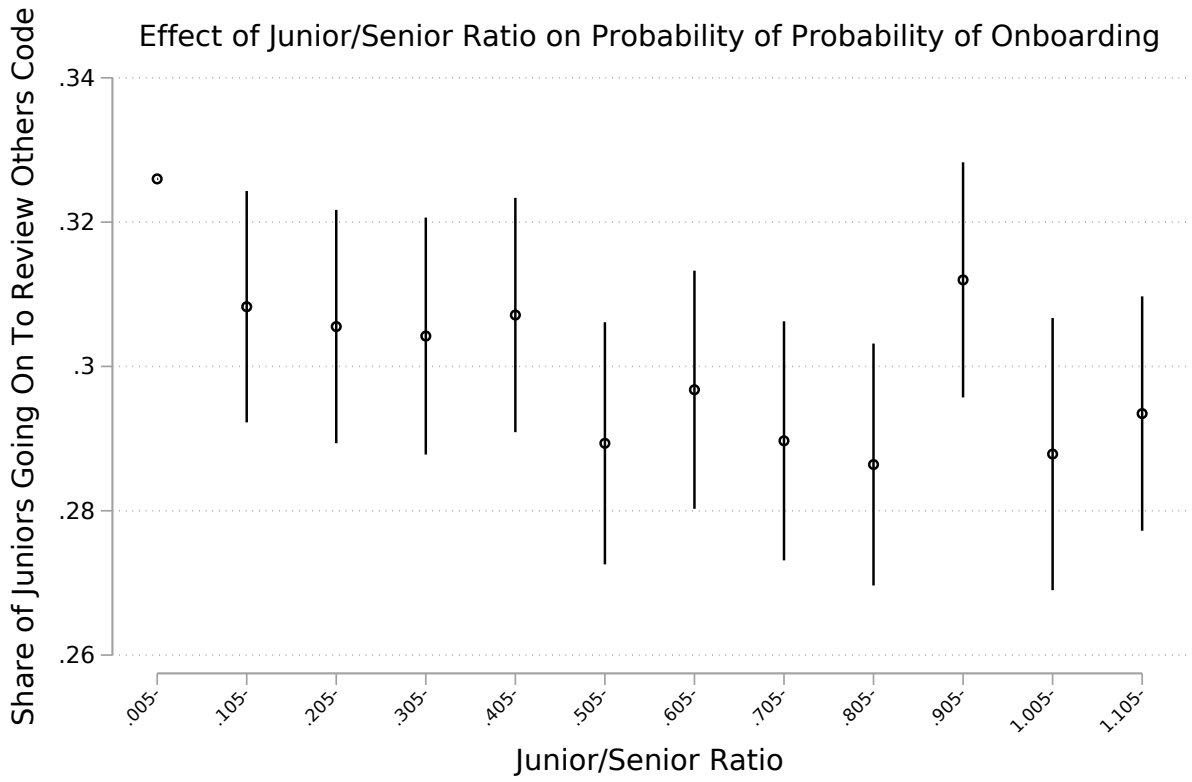
E.3 Congestion Results for Narrower Definitions of Onboarding

We count juniors as "onboarding" successfully in the main text using two observables:

- A junior goes on to remain with the project at least six months
- A junior eventually begins commenting on/merging/closing pull requests opened by others (i.e. code review).

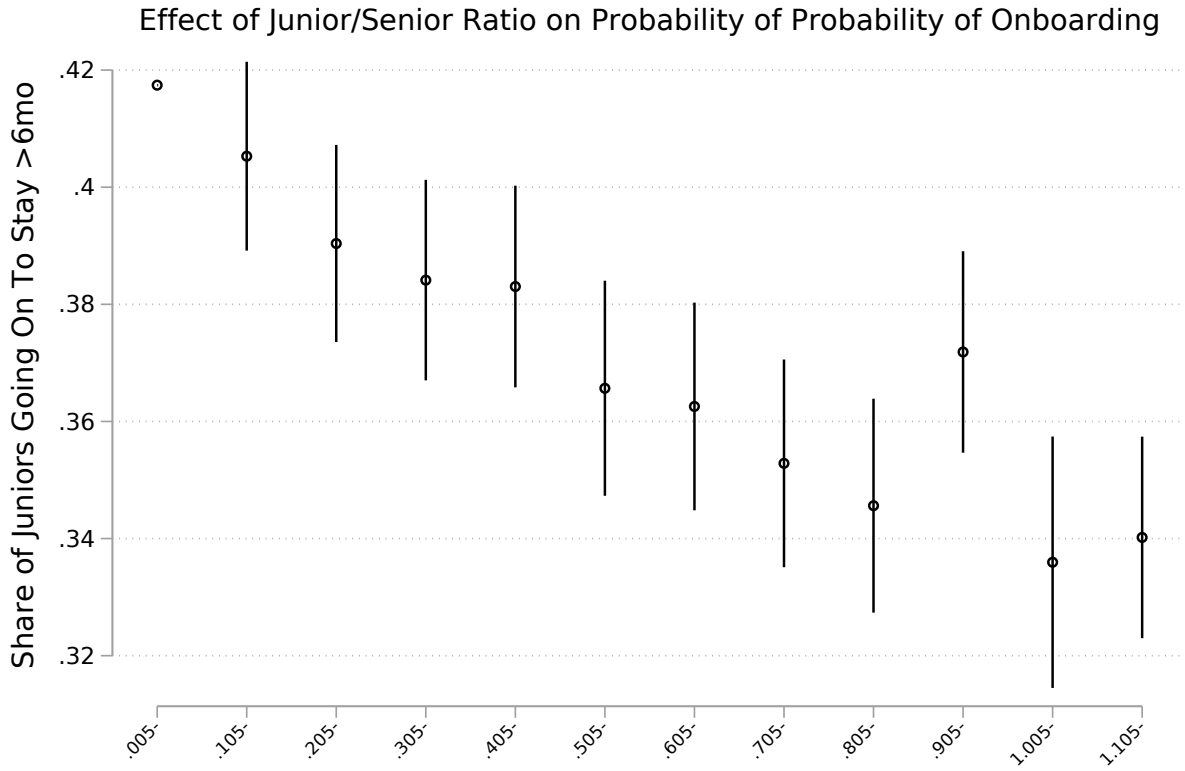
If any junior eventually satisfies one of these two conditions, they get counted as "onboarded." As discussed, many do not and leave within a month of joining the project and without reviewing anyone else's code. Our main specification uses both of these definitions, but using just one or the other yields qualitatively similar results: i.e., an "onboarding function" that looks downward sloping, consistent with congestion.

Figure 17: Non-parametric Estimate of the Onboarding Function with Onboarding Success Determined by Junior's Activity Only



Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin} = b\right) + D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The "spike" in the bin which contains $J/S = 1$ reflects the fact that being "one-on-one" with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$.

Figure 18: Non-parametric Estimate of the Onboarding Function with Onboarding Success Determined by Eventual Project Tenure Only



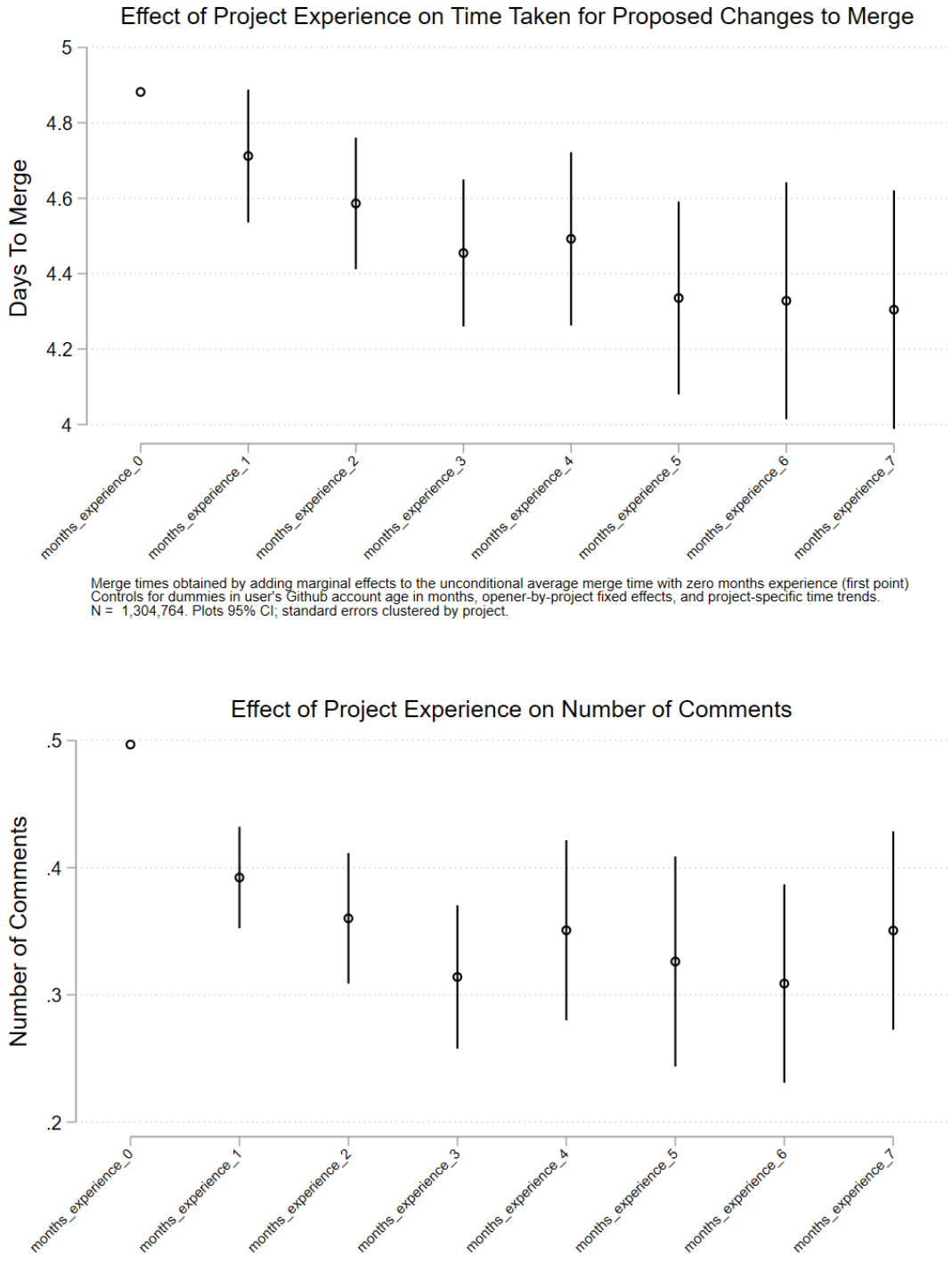
Notes: Estimates from: $\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin} = b\right) + D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$. The "spike" in the bin which contains $J/S = 1$ reflects the fact that being "one-on-one" with an incumbent worker is particularly helpful for successful onboarding. Note over 75% of all project-month observations have $J/S \leq 1$.

E.4 Replicating Main Results on an Earlier GitHub "Snapshot"

The results in the main body of the paper use a snapshot of GitHub from provided by Gousios (2013) on Google BigQuery from June 2019. This section demonstrates that the main results of this paper are qualitatively robust to using an early snapshot: the earliest available on Google BigQuery from September 2016. This predates the acquisition in 2018 by Microsoft, the addition of new features and changes to the API, etc. In short, this section demonstrates that the results of the paper are not sensitive to the "vintage" of data used in the analysis. Note that due to exponential growth in the popularity of GitHub, these three years of data make a large difference: the main text's 2019 sample is almost an order of magnitude larger, which partly explains why results for e.g. pull request comments and approval times in Figure 19 are noisier here.

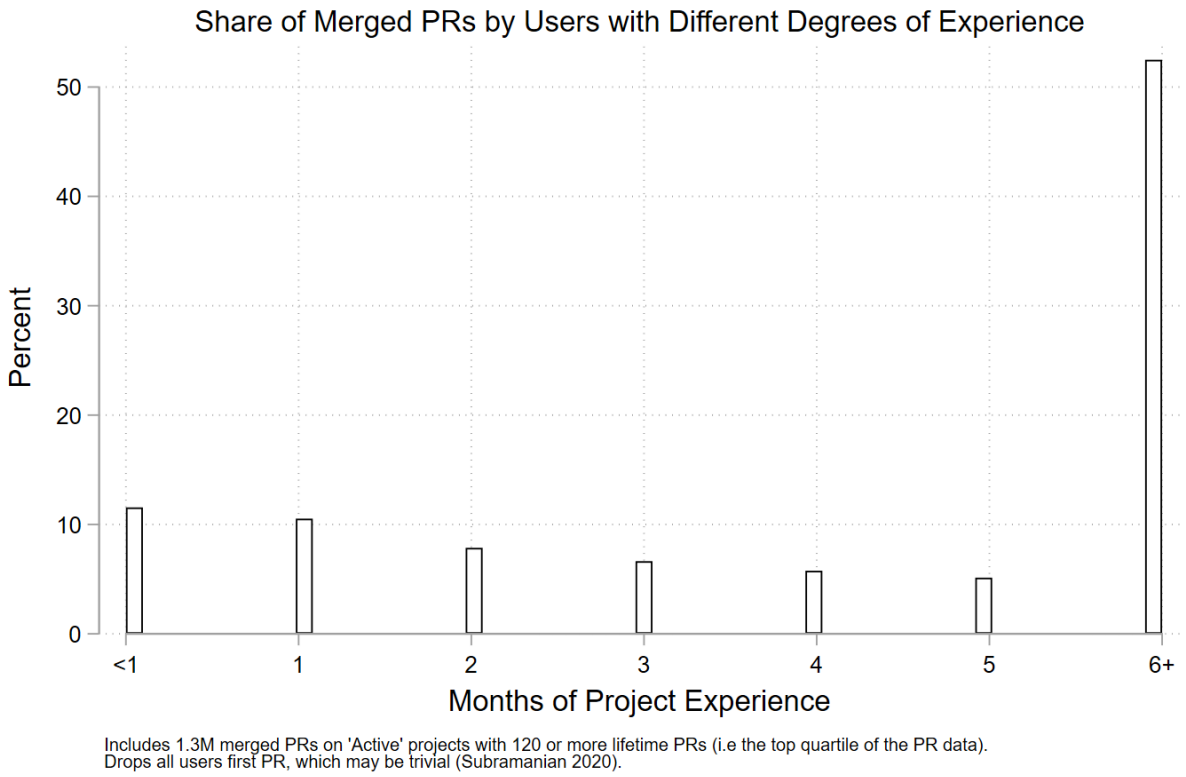
The sample here is otherwise identical to the sample in the main paper, except for the fact that here we kept repositories with at least 120 merged pull request events, instead of 100 as in the main text, which was originally chosen to keep exactly 25% of all pull requests from the largest projects in the 2016 GHTorrent snapshot. The results in the main text and appendices using the 2019 snapshot are robust to using this threshold of 120, as well as higher thresholds (we checked results which kept only repositories with at least 200 merge events as well). See Appendix D (and footnote 47) for details.

Figure 19: Onboarding: with time, new contributors' proposed changes to the code base merge faster and with less discussion (results with 2016 data)



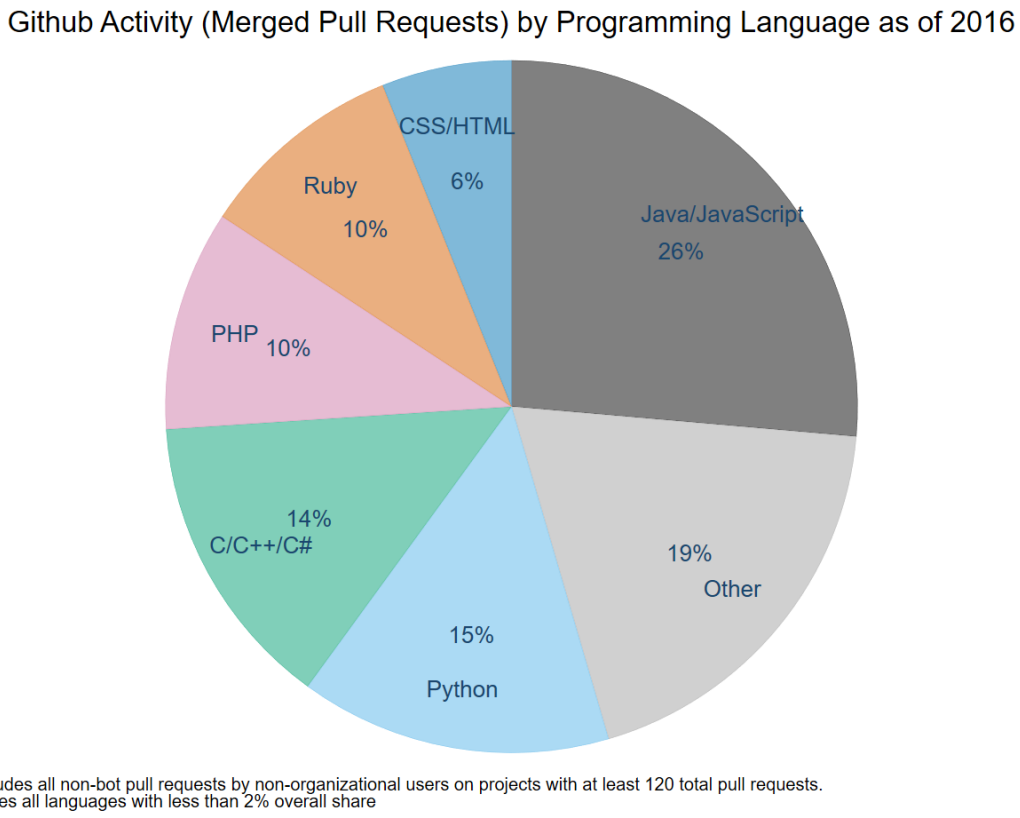
Estimates from $y_{i,p,t} = \sum_{j=1}^{13} D(\text{Months Experience} = j)_{i,p,t} + \sum_k D(\text{Months Ind. Exp.} = k)_{i,t} + D_{i,p} + \beta_{PA,p} ProjectAge_{p,t} + \epsilon_{i,p,t}$ where $y_{i,p,t}$ is either the total time to merge the proposed change in days or number of total number of comments during code review.

Figure 20: Most Work is Done by Experienced Team Members (2016Q3 Sample)



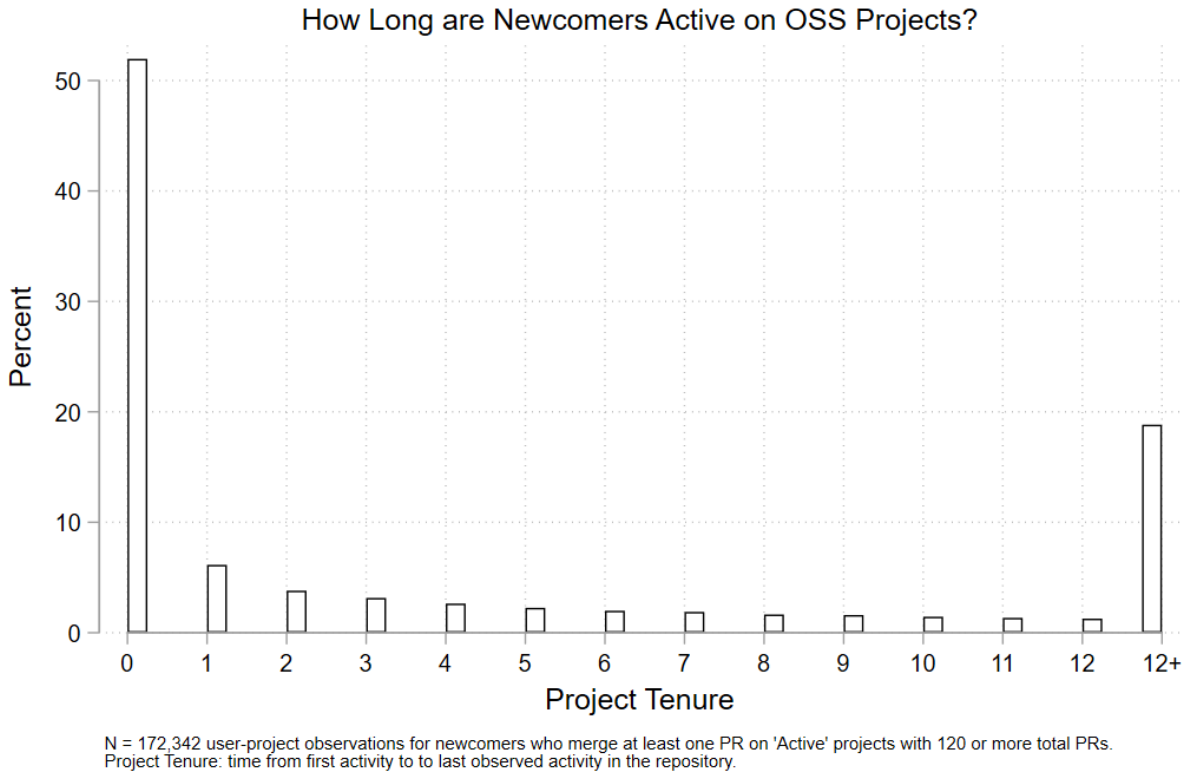
Notes: this figure plots the share of all merged, non-bot pull requests that are opened by users with different degrees of experience, showing that most work is done by those who have at least six months of project-specific experience. We exclude each user's first pull request, given evidence by Subramanian (2020) that these are more often trivial changes. Since we do not otherwise control for complexity or importance of the various tasks completed by these pull requests, and given that longer-tenure workers take on more complex and important tasks, this figure likely understates the importance of work done by senior workers. Source: GHTorrent.

Figure 21: Distribution of Programming Languages (2016Q3 Sample)



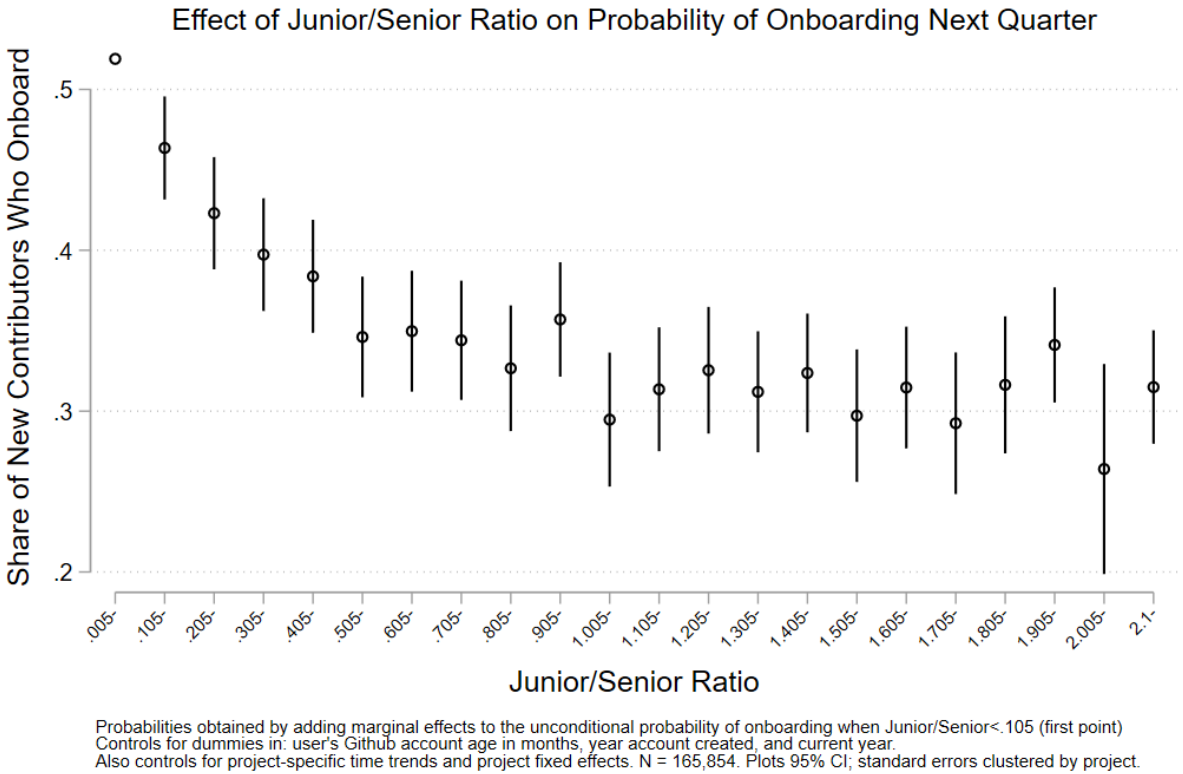
Notes: This does not weight each repository by size, other than dropping all small repositories with less than 120 total merged pull requests. Other includes languages like R, Matlab, and others which are a very small share of the projects in our sample. Source: GHTorrent.

Figure 22: Newcomers Either Contribute Once, or Stay a Long Time (2016Q3 Sample)



Notes: this figure plots the share of all newcomers J (non-bot users who join a project and successfully contribute at least one PR) by their subsequent observed tenure. Most newcomers will go on to have very short tenure (rounded to the nearest month) and contribute once, followed by a nontrivial second group who remain much longer. Source: GHTorrent.

Figure 23: Onboarding Requires Attention from Senior Workers (2016Q3 Sample)



Notes: Over 75% of all project-month observations have $J/S \leq 1$. Estimates from:
$\mathbf{1}(i \text{ joining } p \text{ at } t \text{ onboards}) = \sum_b D\left(\frac{J_{p,t}}{S_{p,t}} \text{ in bin } b\right) + D_p + \beta_{PA,p} ProjectAge_{p,t} + X_t + \gamma_{i,t} + \epsilon_{i,p,t}$