Hatching the Platform Ecosystem: Mobilizing Complementors by Creating Social Foci*

Tommy Pan Fang†
Harvard University

Andy Wu‡
Harvard University

David R. Clough§
University of British Columbia

February 25, 2019

ABSTRACT

Software platforms create value for their users by cultivating an ecosystem of complementary products and services. Existing explanations for how firms attract complementors to a nascent ecosystem prioritize economic mechanisms such as subsidies to platform joiners. By comparison, social mechanisms are overlooked. In this paper, we examine a specific strategy that firms use to cultivate relationships with potential complementors: sponsorship of software development hackathons. We conceptualize hackathons as social foci that orient potential complementors towards the platform and towards each other. We analyze a novel dataset of 1,302 software developers participating in 167 hackathons sponsored by 29 platforms. We find that hackathons act as a locus for social learning that supports the diffusion of platform adoption, over and above the effect of economic subsidies.

Keywords: Innovation Ecosystems, Multi-Sided Platforms, Social Foci, Hackathons

* **Acknowledgements**. The following individuals provided helpful feedback on prior versions of the manuscript: Shane Greenstein, Grace Gu, Marco Iansiti, Daniel Keum, Do Yoon Kim, Wesley Koo, and Feng Zhu. Seminar participants at the University of Connecticut, Harvard University, and Simon Fraser University also provided important suggestions. Raghav Pemmireddy, Eric Wu, and Kevin Wu provided excellent research assistance. All opinions expressed herein are those of the authors only, and all errors are the responsibility of the authors.
† **Tommy Pan Fang**. Address: 60 Harvard Way, Cotting House 302C, Boston, MA 02163 USA. Telephone: (267) 471-0966. Email: tpanfang@hbs.edu.
‡ **Andy Wu**. *Corresponding author*. Andy Wu acknowledges financial support from the HBS Division of Research and Faculty Development. Address: 15 Harvard Way; Morgan Hall 243, Boston, MA 02163 USA. Telephone: (781) 266-8088. Email: awu@hbs.edu.
§ **David Clough**. David R. Clough acknowledges financial support from the Canada Excellence Research Chair in Data-Intensive Methods in Economics. Address: 2053 Main Mall, Vancouver, BC, V6T 1Z2 Canada. Telephone: (604) 822-5413. Email: david.clough@sauder.ubc.ca.

INTRODUCTION

> *How will you get users? If your idea is the type that faces a chicken-and-egg problem in the sense that it won't be attractive to users till it has a lot of users (e.g. a marketplace, a dating site, an ad network), how will you overcome that?*
>
> Application question for Y Combinator seed accelerator (2018)

Software platforms create value for their users by cultivating an ecosystem of complementary products and services (Boudreau, 2012; Eisenmann, 2006; Kapoor & Agarwal, 2017; Rietveld & Eggers, 2018; Tiwana, 2015). Extant literatures on innovation ecosystems (e.g., Adner & Kapoor, 2010; Jacobides, Cennamo, & Gawer, 2018; Pierce, 2009) and platform competition (e.g., Cennamo & Santalo, 2013; Shankar & Bayus, 2003; Zhu & Iansiti, 2012) highlight the strategic importance of securing complementors for sustaining competitive advantage (Kapoor, 2018; McIntyre and Srinivasan, 2017). Research on established ecosystems provides a detailed account of how firms can encourage complementor contributions using levers of pricing (e.g., Hagiu, 2006; Rochet & Tirole, 2003), governance (e.g., Boudreau, 2010; Chu & Wu, 2018; West & Wood, 2013), and platform architecture (e.g., Cennamo, Ozalp, & Kretschmer, 2018; Ethiraj & Posen, 2013; Gawer & Henderson, 2007; Tee & Gawer, 2009; Thomas, Autio, & Gann, 2014).

However, limited attention has been paid to how firms mobilize an initial population of complementors (Dou & Wu, 2018). In nascent ecosystems, firms must produce complements internally (Hannah & Eisenhardt, 2018; Kaul, 2013) or actively cultivate relationships with external complementors (Dattée, Alexy, & Autio, 2018; Hannah & Eisenhardt, 2018; Ott, Bremner, & Eisenhardt, 2018). These initial relationships can jumpstart the virtuous cycle of network effects, i.e., solve the chicken-or-egg problem that platform entrepreneurs face (Caillaud & Jullien, 2003). While practitioners and scholars recognize the importance to nascent platforms of fostering relationships with complementors, empirical research on how to achieve this has been largely anecdotal (e.g., Iansiti & Levien, 2004; Parker, Van Alstyne, & Choudary, 2016).

In this article, we examine a specific strategy that firms use to cultivate relationships with potential complementors: the sponsorship of software development hackathons. Hackathons bring developers together to create new software applications in a short time frame (Lifshitz-Assaf, 2018).[1] Software developers come to these events to learn new skills, create new applications (with teammates), and compete for prizes. As sponsors of the hackathon, platform owners provide financial, in-kind, and in-person logistical support to attending developers. These events are an opportunity for platform owners to evangelize their platform to third-party software developers (Parker, Van Alstyne, & Jiang, 2017).

To understand how hackathons help mobilize complementors, we build on theories of technology adoption from economics and organization theory (Rogers, 2003). Sponsorship of a hackathon constitutes an economic subsidy that encourages a complementor to join a platform (Economides & Katsamakas, 2006). Beyond this well-recognized economic incentive effect, we propose that hackathons also act as forums for social influence (Abrahamson & Rosenkopf, 1997; Wade, 1995). We conceptualize a hackathon as a social focus that orients third-party developers towards the sponsoring platform(s) and towards each other (Feld, 1981; Lomi *et al.*, 2014).

We construct a novel dataset of 1,302 software developers participating in 167 hackathons supported by a set of 29 separate platforms. We track these software developers monthly over time from January 2012 to November 2017 as they each enter a hackathon competition at different times, where the hackathons are sponsored by different platforms. Thus, software developers are differentially treated over time and across hackathons by specific platforms, allowing us to empirically identify the baseline treatment effect on platform use or

[1] While hackathons originate from software development culture, they now appear in use for creative problem-solving in a variety of other managerial contexts, such as strategy formulation, brand transformation, and general product development (Arena *et al.*, 2017; Frolund, Murray, and Riedel, 2018).

adoption of a hackathon that is sponsored by a particular platform. We find that after developers attend a hackathon sponsored by a given platform, their annual hazard of adopting the platform's Application Programming Interface (API) in their coding projects rises by 20.6 percentage points. We find support for a social contagion effect—over and above the effect of economic subsidies—suggesting that hackathons act as a locus for the diffusion of platform adoption.

In the discussion section of this article, we build on the empirical study to consider how firms may leverage a broader range of social foci to mobilize complementors. The face-to-face interactions that social foci enable can constitute a coordination device that presents the platform in question as the most obvious—i.e., the "focal"—solution to a given problem (Schelling, 1980). When taking the view that orchestrating an innovation ecosystem constitutes a coordination problem (Farrell & Klemperer, 2007; Teece, 2007), creating social foci can be a vehicle for generating "focality advantage" (Halaburda & Yehezkel, 2019).

Our study contributes to the strategy literature in two main ways. First, we advance the literature on the antecedents of platform–complementor relationships. Both formal and social mechanisms influence whether a complementor joins a platform ecosystem (McIntyre and Srinivasan, 2017). While prior work has tended to examine the formal mechanisms (e.g., Huang *et al.*, 2013), we add to the field's understanding of social antecedents of platform–complementor relationships. Second, we contribute to the literature on how social structure influences technology diffusion (Coleman, Katz, & Menzel, 1957; Rogers, 2003; Strang & Soule, 1998). Studies of diffusion on networks often treat the social network structure as stable (Jackson, 2016). We work towards a dynamic and agentic model (Teece, 2007) in which actors can intervene in the network by creating social foci, which in turn act as forums for technology diffusion.

THEORY AND HYPOTHESES

Chicken-or-egg problem

Drawing an analogy from biology, the system of actors who join a multi-sided platform is sometimes referred to as the *platform ecosystem* (Ceccagnoli *et al.*, 2012). The concept of business ecosystems has been discussed in practitioner-oriented literature since the mid-1990s (Iansiti & Levien, 2004; Moore, 1996), referring to the way that some firms cultivate a population of partner firms (i.e., complementors) which provide components and services that are complementary to their core product. In a platform ecosystem, the complementary components connect to a platform, defined as a central, stable, hardware or software component in the complex product system (Baldwin & Woodard, 2009).

Whether complementors join a given platform is often viewed as a calculative decision guided by whether the expected benefits of adopting the technology outweigh the costs of adoption (Rysman, 2009). The fixed costs of adoption consist of a mix of monetary costs (e.g., purchasing hardware or software licenses) and cognitive costs (e.g., time spent learning to use the platform) (Brynjolfsson & Kemerer, 1996; Rogers, 2003). The expected benefits of adoption depend on the number of users and complementors already on the platform (Boudreau, 2010; Shankar & Bayus, 2003).

With a large enough population of both users and complementors—a threshold referred to as a critical mass (Evans & Schmalensee, 2010)—a platform ecosystem can be self-sustaining, creating and capturing a stream of economic value. However, reaching that state is major challenge that been likened to igniting an auto-catalytic reaction (Evans, 2009). Practitioner-oriented literature uses case study examples of successful platform launches to identify a typology of possible chicken-or-egg strategies (Parker *et al.*, 2016). The academic literature on platform ecosystems has placed a strong focus on empirically quantifying the strength of indirect network effects (Clements & Ohashi, 2005; Gandal, Rob, & Kende, 2000; Venkatraman & Lee, 2004) and formally modeling optimal platform pricing and governance decisions (e.g., Bolt & Tieman, 2008; Parker & Van Alstyne, 2005; Rochet & Tirole, 2003). Some of these formal models specify price schedules to solve the chicken-or-egg problem by setting negative prices (i.e., giving subsidies) to attract the initial critical mass of users (Economides & Katsamakas, 2006). Besides problems of pricing and governance, another chicken-or-egg strategy that has received attention is that of launching a platform by targeting the users of an existing platform ecosystem, a strategy known as *piggybacking* (Dou & Wu, 2018). Owners of existing platform ecosystems may cultivate new platforms by piggybacking on their existing customer base, in a strategy known as *platform envelopment* (Eisenmann, Parker, & Van Alstyne, 2011; Li & Agarwal, 2017).

Our main proposition is that hackathons surmount the chicken-or-egg problem through *social* as well as *economic* mechanisms. In the hypothesis development that follows, we first examine the economic subsidy mechanism familiar to the literature on platform economics, and then we extend beyond this by building on literature from sociology and organization theory. We enumerate three hypotheses that draw, respectively, on mechanisms of economic subsidies (H1), social contagion in technology adoption (H2), and social influence over expected network size (H3). Here we state a baseline expectation that platform-sponsored hackathons encourage platform adoption. Later hypotheses address specific mechanisms that might contribute to this baseline effect.

Baseline Hypothesis: Attendance by a software developer at a hackathon sponsored by a platform raises the likelihood that the developer adopts that platform.

Subsidies to catalyze platform adoption

When a platform ecosystem has achieved a critical mass of users and complementors, the platform owner can capture value by charging positive prices to new platform adopters (Rochet & Tirole, 2003).[2] Prior to reaching critical mass, however, the platform owner might provide free access or even cash subsidies to attract certain users or complementors onto the platform. For example, Paypal incentivized referrals by offering a $10 subsidy to the referrer and to the person signing up (Parker *et al.* 2016). Economides and Katsamakas (2006) formally specify the prices that can attract a critical mass of platform joiners, allowing prices in their model to take negative values. They indicate that the negative prices can be interpreted as subsidies to application developers such as "access to application programming interfaces, resources, and information" (2006: 1060).

In light of the models by Economides and Katsamakas and others, we could consider sponsoring hackathons in order to attract complementors to a platform to be a form of subsidy. Software developers attending hackathons may receive explicit coaching by company personnel on how to incorporate the platform in their code. This type of non-monetary subsidy brings down the cognitive burden of adopting a platform. Developers may also participate in a competition that awards a monetary prize to the best application created during the hackathon that complements a given platform. This provides a monetary incentive to hackathon attendees to experiment with the platform and invest cognitive resources in learning how to use it.

To empirically validate the economic mechanism related to platform attendance, ideally one would directly measure variance in all subsidies provided at a hackathon and relate this to variance in subsequent software developer behavior. For example, if a sponsor provided more

[2] Specifically, Rochet and Tirole (2003; 2006) show that platform owners should charge higher prices to the "side" of the platform for which demand is less elastic. The side with more elastic demand may be charged zero or be subsidized even after a critical mass of platform joiners is achieved.

personnel to train hackathon attendees in how to use the platform interface (specifically, a higher ratio of personnel to attendees), this would constitute a larger cognitive subsidy to the attendees and might strengthen the impact of the hackathon on their subsequent behavior. We do not have the data necessary to quantify cognitive subsidies at the hackathon. However, we do have data on the competition prizes at the hackathon, which we argue constitute a form of monetary subsidy to hackathon attendees. By competing for a prize at a hackathon, the developer is not assured of receiving a monetary subsidy. But they may perceive that their chance of winning the prize is at least as good as anyone else's (Boudreau, Lacetera, & Lakhani, 2011). We therefore treat the "expected subsidy" as the value of the prize divided by the number of hackathon attendees. The economic mechanism for solving the chicken-or-egg problem predicts that higher expected subsidies would strengthen the effect of a hackathon on software developer behavior.

> *H1: The effect of attending a hackathon on adopting a sponsoring platform is stronger the larger the expected subsidy provided by the sponsor to attendees who use the platform at the hackathon.*

Social influence and platform adoption

Economic models of platform adoption tend to treat the decision to use a platform as resting on its price and governance structure, the size of its existing user base, and the stand-alone value (i.e., quality) of the platform itself. These factors feed into a cost-benefit calculation by a potential platform joiner (Rysman, 2009). In this subsection, we complement this existing work and extend platform adoption theory by drawing on the literature on technology diffusion. We contribute towards a theoretical model of platform ecosystems that incorporates both economic and social mechanisms (Afuah, 2013). Our underlying behavioral assumption is that potential platform joiners are boundedly rational, goal-oriented, social actors.

We argue that a platform ecosystem can be conceptualized as a *social focus*.[3] Scott Feld introduced the concept of the social focus to describe a "social, psychological, legal, or physical entity around which joint activities are organized (e.g., workplaces, voluntary organizations, hangouts, families, etc.)" (1981: 1016). Two individuals affiliated with the same social focus are more likely to come into contact with one another than two individuals who do not share a social focus (Dahlander & McFarland, 2013; Laumann, 1973). Feld's work highlights social foci as generators of social structure, i.e., spaces in which social ties are established (Feld and Grofman, 2011). We expand on this by highlighting social foci as forums for mutual social influence.

Well-known examples of multi-sided platforms are also clear examples of social foci, such as the traditional village marketplace. A village marketplace is a meeting point for buyers and sellers of local produce. A municipal government authority typically sets rules for sellers and charges an entry fee for sellers to have a stall (Geertz *et al.*, 1979). As a multi-sided market, more buyers are attracted by a greater variety of sellers, and vice versa (i.e., buyers and sellers are cross-side complements, resulting in indirect network effects) (Parker *et al.*, 2016). The village marketplace is also a social focus because buyers and/or sellers form social ties through interaction (Plattner, 1989). The village marketplace solves a coordination problem in dimensions of time and geographic space: buyers and sellers know when and where to be in order to engage in economic exchange (Geertz, 1978; Schelling, 1980).

Digital technology platforms might seem at first to have little in common with village marketplaces except that both represent a multi-sided market. We argue, however, that digital platforms always also have a social dimension. Platforms do not just connect components; they connect people. For example, on digital freelancing platforms, the relationships formed between

[3] We are not the first to make this observation (see, e.g., Shankar and Bayus, 2003). However, to our knowledge, the theoretical implications of this approach have not been explored in existing research.

buyers and sellers sometimes lead the platform users to interact directly with one another outside of the platform (Gu & Zhu, 2018).

Building on the idea that virtual, digital platforms often map to real-world social foci, we suggest that the creation of social foci is a strategic tool for firms. As an example, users of e-commerce platforms, such as Craigslist, often meet in person to complete a transaction. Craigslist benefits from the existence of designated physical spaces for the exchange of goods that are well-lit and monitored by security cameras (Skahill, 2015). Digital ride-sharing platforms benefit from having known focal points at busy locations (e.g., airports), where drivers and riders can meet. Digital technologies for the sharing of encrypted material still rely on in-person meet-ups to allow for verification of identities and sharing of the "public keys" that underlie the encryption tools. Social focal points enhance the value of a platform by enabling organized interactions between platform users and/or complementors.

We argue that a hackathon is a face-to-face social focus that helps coordinate actors in the platform's ecosystem. It generates social structure in the form of new social relations between software developers who meet and collaborate for the first time, and in the form of adoption ties between developers and the platforms they join. It acts as a forum of social learning which affects attendees' technology adoption decisions and their expectations regarding other actors' future adoption decisions.

Social learning at hackathons

The sociological literature on technology adoption shows that an actor's adoption decision is affected by social influence (Coleman *et al.*, 1957; Rogers, 2003). The usefulness of a new technology is initially uncertain (Mansfield, 1961). Awareness that other individuals or organizations have adopted a technology can raise a focal actor's perception of its value, thus raising their likelihood of adopting it (Burt, 1987). Evidence supporting this mechanism has been found in settings ranging from shipping (Greve, 2009) to aerospace (Greve & Seidel, 2015) to telecommunications (Dekimpe, Parker, & Sarvary, 2000) and venture capital (Gaba & Meyer, 2008). Social information processing can lead to "informational cascades" in which inferential learning from prior adoptions spreads at an accelerating rate (Bikhchandani, Hirshleifer, & Welch, 1992).

The observability of other actors' adoptions depends on the structure of the relationships between actors. For example, Burt (1987) finds that the structural equivalence of physicians in a social network of advice ties raises the influence one physician has on another's adoption of a new drug. Greve (2009) finds that centrality in the network of shipping firm–shipbuilder ties is associated with the earlier adoption of a production technology that provides competitive advantage to the shipping firm. As a catalyst of interactions between actors, social foci both generate new social ties and make visible the past adoptions of other actors, raising the likelihood of actors influencing one another.

In this way, a hackathon exposes potential platform adopters to other individuals who may have already adopted the platform. The software creation at a hackathon is an activity in which past platform adoption or non-adoption is visible and salient to fellow hackathon participants. Hackathons are therefore likely to be forums in which observing past adoptions raises the likelihood that a focal actor will then adopt a given platform.

H2: The association between attending a hackathon and adopting a platform is stronger the higher the proportion of hackathon attendees who have already adopted the platform prior to the hackathon.

Social influence on expected network size

Separate from their inferences about intrinsic platform quality, exposure to other users of a platform can lead a potential adopter to shift their expectation over the likely future size of the installed base of users. When network effects are strong, a complementor prefers to join a platform with a larger installed base of users even if that platform has inferior intrinsic quality (Zhu & Iansiti, 2012). In the case of platforms for collaborative software development, we assume developers perceive direct network externalities to be positive: the more developers have adopted the platform in the past, the higher the likelihood that the given platform becomes dominant and the greater the opportunities for collaborating on software development for that platform in the future. In other words, the choice of which platform to join resembles a coordination game, in which players receive a higher payoff when they select the same option as other players (Halaburda & Yehezkel, 2016).

To investigate whether expected network size affects adoption likelihood, we treat software developers as "intuitive Bayesians" (El-Gamal & Grether, 1995) and suggest that hackathons are an opportunity for them to update their expectation of network size. We suggest their "prior" on a platform's indirect network size is based on the proportion of adopters in the broad population. If the proportion of adopters at a hackathon is higher than the broader population, this leads the developer to update their prior to a higher expected network effect (Lee, Lee, & Lee, 2006). Individuals take into account information on expected network size when making decisions regarding whether to join a given platform (Tucker & Zhang, 2010). An increase in expected network size at a hackathon is therefore predicted to raise the developer's likelihood of subsequently adopting the platform.

H3: The association between attending a hackathon and adopting a sponsoring platform is stronger when the proportion of hackathon attendees who have already adopted the platform prior to the hackathon exceeds the proportion of adopters in the broader population.

Complementarity between economic and social mechanisms

We have argued that hackathons impact platform adoption through both economic and social mechanisms. Central to our theory is the notion that the social mechanism provides explanatory power over and above the economic mechanism. To strengthen this case, we examine how the two interact. If the social variables simply capture economic incentives to adopt, then the social and economic factors considered together would substitute for one another (Anderson, Parker, & Tan, 2014; Cennamo & Santalo, 2013; Hagiu & Spulber, 2013).

However, if social factors are truly distinct from economic factors, we expect that social and economic factors should complement one another (Dou & Wu, 2018). This is because developers' expectations of the platform's future network size are affected by their perceptions of *other* developers' costs of adopting the platform. Thus, a subsidy provided by the platform provider impacts a developer's adoption decision through two channels: a "first order" channel in which it reduces their fixed cost of adopting the platform, and a "higher order" channel in which it reduces their perception of other software developers' fixed cost of adopting. Within a social focus, an individual receives a subsidy to adopt a platform, but they also observe others receiving the subsidy to adopt the platform. Thus, the social focus may potentially amplify the underlying mechanisms that encourage platform adoption. Network effects generate a "feedback loop" in which mechanisms stimulating platform adoption mutually reinforce each other (Sterman, 2000) via actors' shared expectations of other actors' behaviors and expectations.

H4: An expected subsidy strengthens the association between attending a hackathon and platform adoption to a greater extent when the proportion of hackathon attendees who have already adopted the platform prior to the hackathon exceeds the proportion of adopters in the broader population.

EMPIRICAL ANALYSIS

Empirical setting

Our empirical study of platform adoption by software developers centers on hackathon events sponsored by platform owners. Hackathons attract individual or small teams (generally two to five) of software developers who work intensely to create new software applications from scratch. Over the duration of one or more days, the developers co-locate, and they work, eat, and even lodge near one another. As we will elaborate in the sample construction, we focus on physical hackathons planned and operated by an organizer independent of any platform owner.[4] As such, these hackathons place few to no restrictions on the kind of software to be created and, most importantly, they impose no requirements to develop for a particular platform in order to participate.[5]

The organizer solicits financial and in-kind support from sponsors consisting of platform owners and other interested parties to cover the costs of the event, e.g., renting physical space, purchasing food, and setting up high-capacity wireless infrastructure. For the particular sample of hackathons that we consider, developers come to these hackathons largely unaware of the set of sponsors, until they arrive and see the physical presence of the sponsor. The sponsoring platform owners, generally technology firms, use the hackathon to simultaneously serve several

[4] Examples of independent organizers include trade associations, non-profit economic development organizations, and university student groups.

[5] Two other types of hackathons are internal company hackathons and API-specific hackathons focused on developing new applications for the platform. Studying these would reduce our ability to infer causal mechanisms from our empirical analysis since developers self-select into attending hackathons. As a result, we only study broad hackathons whose sponsors are not known to participants in advance. We discuss this in more detail in the Empirical Analysis section below.

of their interests, such as recruiting engineers or developing general brand recognition, but we focus on their specific interest of promoting adoption of their platform by developers.

At the end of a hackathon, each developer team demonstrates their final project in front of the audience to gain recognition and have an opportunity to win monetary and non-monetary prizes. Hackathons universally offer general prizes for which all applications are eligible. Beyond the main overall prizes, sponsoring platform owners offer prizes to incentivize developers to build applications favoring their platform, such as a prize for the best applications specifically utilizing the platform of the sponsoring firm. Prior to the event, developers are largely unaware of the set of platform-specific prizes available. Once developers arrive, they learn which platform-specific prizes are available, and then they evaluate the available set of prizes to decide which one(s) they want to contend for during the application development.

Sample

To study whether hackathons catalyze complementor adoption of a platform, we collect data on software developers who attended hackathons, and we measure their usage of platforms over time. We track the software development activity of 1,302 developers at monthly intervals from January 2012 to November 2017. Our main dataset consists of a developer-platform-month panel with 783,474 observations. We assemble this data in several stages. We first select a sample of hackathons. Second, we identify the participating developers and collect their corpus of publicly accessible software projects, which we analyze to track their use of platforms in their developed code.

Hackathon sample

We collect data on hackathons, and identify developers participating in those hackathons, from the website of Devpost. Devpost provides organizing and registration services to many of the world's in-person hackathons. Hackathon organizers use Devpost to receive software projects from participants for consideration by judges for competition awards, including those awarded in conjunction with the technology firms sponsoring a hackathon competition. For each hackathon, we record the date, participants, location, sponsoring platforms, and the prize(s) awarded.

We select hackathons for inclusion in our sample based on a number of criteria that match our theoretical framework and empirical strategy. First, as our theory relates to in-person gatherings, we select hackathons with a physical venue location. These hackathons—in contrast to virtual hackathons taking place online—also have the benefit of relatively homogenous organizing practices, such as a relatively short event duration of less than a few days.[6] Second, to ensure we can construct a time-series of developer platform adoption both before and after the hackathon, we consider hackathons taking place between January 2014 and May 2017. Third, we select hackathons sponsored by the 29 most-frequent platform sponsors of hackathons events.[7] Importantly, we exclude hackathons that prominently featured a single platform sponsor in the event title or that only offered prizes from a single sponsoring platform. By removing these hackathons closely associated with a single platform sponsor, we lower the risk that developer self-selection into the hackathon treatment sample—due to attraction to a given platform sponsor—will bias our results.[8] Finally, we exclude hackathons with fewer than ten identifiable participants because social interaction within a social focus is central to our theoretical framework. The remaining set of 167 hackathons serve as the quasi-experimental treatment events in our study. From Devpost, we identify the developers participating in these hackathons;

[6] Our sample of hackathons has a median duration of 24 hours, ranging from 8 to 60 hours.
[7] For the long tail of sponsoring platforms, non-standardized documentation across platform APIs functionally limited our ability to cover the full set of 19,000+ platform interfaces in use today. While we originally considered the top 30 most frequent hackathon sponsors, which included Github as one of those most frequent hackathon sponsors, we exclude Github in our final sample due to possible confounding effects related to our use of Github as a data source.
[8] We confirm that sponsor information was generally not available prior to the event. From a random 10% sample of our filtered hackathons, we use past websites from the Internet Archive to compare the hackathon details page on Devpost before and after each event.

consequently, all the developers in our sample have attended at least one of these hackathon events and submitted a project for judging via Devpost.[9]

Developer–platform data

For the sample of hackathon-participating developers, we track their software development activity longitudinally before and after they participate in a hackathon based on projects they upload to Github. Github is an online code repository, widely used by software developers to publicly share what they are working on and to help them manage version control. Github serves as a valuable source of data to measure longitudinal developer activity (Gousios & Spinellis, 2014): over 26 million developers host their software projects on Github as of March 2017.[10] Many developers, including those participating in hackathons, use Github to enable team collaboration. Github also enables collaboration among open-source community members, who test and evaluate public projects in addition to contributing code (Mollick, 2016).

We longitudinally identify the platforms used by these developers by text-mining the corpus of underlying code in their software projects on Github. In our download of the full project code, we observe the date of project creation, the active project time window, and the raw source code underlying the software project.[11] We omit projects copied by the developer from elsewhere, since most of the source code in such instances may not have been written by the developer (Wu, Wang, & Evans, 2019). Because our empirical strategy relies on a longitudinal research design, we restrict the sample of developers to those who had at least one project before and at least one project after any hackathons that they attended. In all, we identify 54,487 Github

[9] Projects are submitted by individuals, whether or not the project was created by an individual participant or a team where the submitting developer had a leadership role in the group.

[10] Github was founded in February 2008 and grew quickly to surpass other popular code-hosting sites in the total number of coding file revisions by June 2011. On Github, developers store their "source code," which is any collection of computer instructions written using a human-readable programming language.

[11] Github only allows data access to a developer's first one hundred projects alphabetically by project name. Developers rarely had more projects than that, but nevertheless we were restricted from accessing this data beyond these one hundred projects per developer.

projects for 1,302 developers who participated in 167 unique platform-sponsored hackathons.

We construct a developer-platform-month panel dataset from this body of software projects. We identify projects developed using a platform's technology by searching the code for a set of unique platform-specific API keywords, enabling us to examine specific platform adoption and usage for each developer project; the Appendix provides further details on this API identification process. The window of activity of a project derives from the dates of creation and modifications to the project. From this window, we identify which months the developer used the platform's technology. This process leads to an unbalanced panel data set, with the time series for each developer–platform pair starting either in January 2012 or when the developer creates her first project on Github. To give us at least half a year of post-hackathon activity for developers who attend hackathons in May 2017, we end the time series for each developer in November 2017.

Dependent variables

We follow the standard approach in the literature on technology adoption (e.g. Seamans, 2012) by using event history analysis to estimate the hazard of a developer adopting a platform. Since our variables are updated monthly, we use discrete-time event history analysis (Allison, 2014). In line with this approach, our main dependent variable, *Platform adoption,* is a binary variable coded zero for a developer-platform-month in which the developer has not yet used the platform in their software; it takes a value of one in the first developer-platform-month in which they use the platform, i.e. write code that calls on a platform's API. A developer's decision to adopt a platform is a non-repeatable event: thus, after a developer has adopted a platform, subsequent observations of that pair are no longer included in the risk set (Allison & Christakis, 2006).

In a robustness test, we estimate developers' intensity of use of a platform to capture not just new adoptions, but also the ongoing activity of existing platform users. We specify the dependent variable, *Platform development*, as a count of the developer's active GitHub projects that use a specific platform in a given month; further detail on the construction of this variable appears in the Appendix.

Independent variables

Hackathon attendance

Our main independent variable is a dichotomous variable measuring whether a developer attends a platform-sponsored hackathon. For each developer–platform pair, this variable takes the value of one for the month in which a developer attended a hackathon sponsored by the platform and all the months that follow, and it takes a value of zero otherwise. This variable is zero for platforms that do not sponsor the hackathon attended by the developer. By attending a hackathon event sponsored by the platform, the developer may be affected by the channels that are described in our theory. To gain insight into these mechanisms, we construct additional independent variables based on data from Devpost.

Expected subsidy

We operationalize the use of subsidies to lower adoption costs with the expected prize amount that a developer receives from attending a hackathon event sponsored by a platform. To calculate this measure, we collect information from Devpost about the prizes offered at each hackathon event, as well as the platform that is sponsoring each of these prizes. For each event, we sum the pool of prizes offered by each platform to get the total subsidy provided by a platform-sponsor at a hackathon event. Because developers lack knowledge of the abilities of other developers competing for a prize, we treat the expected subsidy as the total subsidy divided by the number of developers attending the hackathon. In other words, we assume each developer perceives their chance of winning a prize to be the same as the other developers in attendance. We measure the *Exp. subsidy* in thousands of US dollars and log-transform the variable due to skew.

Local adoption rate

To test for whether hackathons act as a forum for technology diffusion we measure the proportion of hackathon attendees who were already platform users prior to the hackathon. In the month of the hackathon and the months that follow, we define *Local adoption rate* as the count of peer developers at a hackathon who have already adopted the focal platform, divided by the total number of peer developers at the hackathon. In months prior to the hackathon, this variable is coded as zero. Social contagion may influence developer behavior regardless of whether a platform is a hackathon sponsor or not, and so we define this variable for all platforms in our dataset.

Network concentration

This variable examines whether social influence at a hackathon shifts a developer's expectations about a platform's network size. We assume that a positive shift occurs if the developer attends a hackathon where the local adoption rate of a given platform is higher than the broader adoption rate of that platform in the wider world. In order to measure this, we create a dummy variable that is set to one if a developer i has attended a hackathon where the proportion of developers who have already adopted platform j at the event exceeds the proportion of total developers in our sample who have adopted platform j, at the time of hackathon attendance. This discrete variable specification is consistent with the idea that platform adoption exhibits complex dynamics akin to a local "winner-take-all" effect in a social network (Lee, Lee, & Lee, 2006).

Attending a hackathon can produce a shift in the developer's beliefs about the future success of a platform. A shift in the developer's expectation of the network size impacts their perceived value of using the platform in future. We expect that the proportion of hackathon attendees who are already platform users becomes salient when the platform is a sponsor of the hackathon. We therefore test regression models in which we interact *Network concentration* with *Hackathon attendance*. The presence of a sponsor can draw attention to the high proportion of platform users present at a hackathon, which leads other developers to update their expectations about the value of the network effects that the platform will provide.

Control variables

We sought to address unobserved time-invariant and time-variant heterogeneity in both developers and platforms through fixed effects and a control variable. A primary concern for an observational study of this type is that developers may use a platform for reasons that are unrelated to the hackathon mechanisms described previously. Platform-month fixed effects—subsuming separate platform fixed effects and time fixed effects—address general trends in platform use over time. Importantly, platform-month fixed effects control for changes in the indirect network size and stand-alone value of each platform. To address time-variant heterogeneity in developers, we include a control variable for *Project experience*. We generate a stock count of a developer's project activity for each month to measure their accumulated experience. We log transform *Project experience* and lag the variable by one month.

Descriptive statistics of the variables and their correlations are presented in Tables 1 and 2. Table 1 reports summary statistics at the platform-developer-month level. *Platform adoption* has fewer observations because the sample used in these regressions consists of only the developers at risk of adoption. In Table 2, *Network concentration* is calculated using *Local adoption rate*, and so these variables are highly correlated. We include both variables in our analysis to disentangle the effects of social learning and increases in expected network size described in our theory.

---------- INSERT TABLES 1 AND 2 ----------

Empirical strategy

In our data, hackathons are staggered over time and developers may be treated by exposure to different platform-sponsored hackathons over time.[12] This design allows us to compare outcomes for developer–platform combinations that have previously attended a hackathon sponsored by the platform (treatment group) against developer–platform combinations that have not attended a hackathon sponsored by the platform (control group).

For developer i, platform j, and month t, we regress an indicator variable for the developer's adoption of a platform $Platform\ Adoption_{ijt}$ on various independent variables based upon hackathon attendance and the characteristics of the attended hackathon. We employ the linear probability model: compared with non-linear models such as logistic regression, this eases the interpretation of interaction effects and avoids the incidental parameters problem, which can affect non-linear models when fixed effects are introduced (Greene, 2012).[13]

We estimate a number of specifications, building up from a parsimonious model to more specified models that address the theoretical mechanisms. Equation 1 shows the most fully saturated model, for developer i, platform j, and month t. All models control for $L\ Project\ Experience_{it-1}$ and platform-month fixed effects $\gamma_j \times \delta_t$. First, our baseline model focuses on the independent variable of $Hackathon\ Attendance_{ijt}$, representing whether the developer has attended a hackathon event at or before month t that was sponsored by the platform j. To measure the theorized mechanisms relating hackathon attendance to subsequent

[12] Our base specification resembles a generalized differences-in-differences design (Angrist & Pischke, 2009).

[13] We include developer fixed effects in our robustness tests predicting *Platform development*. In our main models predicting *Platform adoption* we omit the developer-level fixed effects; in event history models, these can bias the coefficients of other independent variables that vary monotonically over time (Allison & Christakis 2006; Nanda & Sorenson 2010). Nevertheless, we have tested specifications that include developer-level fixed effects, and we find our results are robust to including them.

platform adoption, we then introduce specific independent variables. Second, we examine the effect of $L\ Exp.Subsidy_{ijt}$, which represents the average expected subsidy provided to a developer *i*, by a platform *j*, at a hackathon event at or before month *t*. Third, we consider $Local\ Adoption\ Rate_{ijt}$, which measures the adoption rate of platform *j* at a hackathon event attended by developer *i*, at or before month *t*. Fourth, we introduce $Network\ Concentration_{ijt}$ an indicator for whether a developer *i* receives a positive shift in expectations for platform *j*, at a hackathon event at or before month *t*, interacted with $Hackathon\ Attendance_{ijt}$. Finally, our last and complete specification adds the interaction term, $Network\ Concentration_{ijt}$ X $L\ Exp.Subsidy_{ijt}$.

$$
\begin{aligned}
&Platform\ Adoption_{ijt} \\
&\quad = \beta_1 Hackathon\ Attendance_{ijt} + \beta_2 L\ Exp.Subsidy_{ijt} \\
&\quad + \beta_3 Local\ Adoption\ Rate_{ijt} + \beta_4 Network\ Concentration_{ijt} \\
&\quad + \beta_5 Network\ Concentration_{ijt} \times Hackathon\ Attendance_{ijt} \\
&\quad + \beta_6 Network\ Concentration_{ijt} \times L\ Exp.\ Subsidy_{ijt} \\
&\quad + \theta\ \mathrm{L}\ Project\ Experience_{it-1} + \gamma_j \times \delta_t + \varepsilon_{ijt}
\end{aligned}
\tag{1}
$$

Identification considerations

For the purposes of addressing our theoretical question, the ideal empirical experiment would involve the random assignment of developers to physical gatherings sponsored by platforms. Given the feasible archival data that can be studied, the main identification consideration for this type of observational study is self-selection by developers into hackathons sponsored by platforms they are interested in adopting. To minimize the potential endogeneity from this channel, we specifically design our sample, empirical methodology, and covariate coverage to address this type of omitted variables bias.

In constructing the sample, we select a set of hackathons and platforms that minimize endogeneity concerns. We focus on hackathons where the platform sponsors are not publicized to the developers or the general public prior to the event, making it less likely developers would have the knowledge to self-select into the hackathon based on the sponsors.

In the choice of empirical methodology, we leverage our longitudinal panel data at the developer-platform-month level with a battery of fixed effects and controls to cover explicitly the various mechanisms that might be at play relating hackathon attendance to platform adoption. Table 3 presents a mapping between the measures used in our analysis and the set of mechanisms that affect developer–platform tie formation. Platform-time fixed effects address general trends in platform use over time, such as the indirect network size and stand-alone value of each platform.[14]

Beyond the design choices made in the construction of the sample and empirical methodology, we also conduct two tests of the validity of our empirical design against possible bias. This type of empirical design rests on the parallel trend assumption: in the absence of treatment, the difference between treatment and control must remain constant over time. In our setting, we would like for the trends between developer–platform pairs in our treatment and control groups to be the same before hackathon attendance. We conduct two separate tests—based on what can be identified via observable characteristics—to support our econometric approach by verifying that this critical assumption is met.

First, we check for balance on pre-treatment observables between the two groups across the various platforms considered, and we present a portion of the analysis in the Appendix. For

[14] Our study of *Platform adoption* is robust to the inclusion of developer fixed effects to control for time-invariant unobservable heterogeneity across developers. We also conduct an analysis which included hackathon fixed effects, which control for time-invariant unobserved heterogeneity for each hackathon and the set of developers that attend a particular hackathon, including differences related to the geographic location of hackathons. These hackathon fixed effects are in place of the developer fixed effects, which are absorbed by the hackathon fixed effects: based on the way our sample was constructed, each developer only attends one hackathon, so hackathon fixed effects are collinear with developer fixed effects. In practice, developer fixed effects capture more granular unobserved heterogeneity than hackathon fixed effects, which is why we err towards developer fixed effects in the models presented in the paper.

each platform, we find that the level of platform adoption and development is comparable between the treatment and control group, with no statistically significant difference. We find no statistically significant differences between the two groups across all the platforms considered in our study. In addition, in an analysis pooled across all platforms and developers, we find no statistically significant differences in pre-treatment developer observables, across the two groups. Developer–platform pairs in the two groups appear to be comparable because they exhibit similar levels of project experience and develop on a similar number of total platforms.

Second, we assess a visual representation of the effect of the main independent variables to identify any possible non-parallel trends, shown in Figure 1. We find no evidence to violate the parallel trend assumption. There appears to be a sufficient common pre-trend between treatment and control groups across various independent variables; the groups diverge only after the treatment of hackathon attendance. These two tests support our econometric approach.

---------- INSERT TABLE 3 ----------

---------- INSERT FIGURE 1 ----------

RESULTS

Table 4 reports our main regression results. Model 4.c includes the control variables; subsequent models add independent variables to test our hypotheses. We note that model fit—estimated by the *Adjusted* R^2 coefficient—increases from 0.007 to 0.025 moving from the left to the right of the table. The explanatory power of our model increases as we take account of more theoretical mechanisms that describe a developer's platform adoption behavior.

---------- INSERT TABLE 4 ----------

Model 4.b adds *Hackathon attendance*, which captures the association between attending a platform-sponsored hackathon and a developer's hazard of adopting the sponsor platform. The positive and significant coefficient on this variable provides support for our baseline hypothesis.

The effect size implied by this coefficient is large: upon attending a hackathon sponsored by a given platform, the monthly hazard that the developer adopts the platform rises by 1.9 percentage points. Annualized, this is an increase in adoption hazard of 20.6 percentage points. The baseline hazard that a developer adopts any given platform in any given month is roughly 0.1%; after attending a hackathon sponsored by the platform, that hazard is roughly 2.0%.

Model 4.1 adds the variable *L Exp. subsidy*, which tests Hypothesis 1. Consistent with Hypothesis 1, the coefficient on this variable is positive ($p \sim 0.000$): the level of subsidy provided by the platform at a hackathon is positively associated with the developer's likelihood of adopting the platform. For developers who have attended a hackathon, receiving an expected subsidy one standard deviation above the mean level is associated with a 2.7 percentage point increase in the monthly hazard of the developer adopting the platform.

Model 4.2 adds the variable *Local adoption rate*, which tests Hypothesis 2. The positive coefficient ($p \sim 0.000$) estimate on this variable supports the hypothesis. Consistent with the hackathon acting as a forum for social influence, every ten percentage point increment in the *Local adoption rate* of a given platform at a hackathon is associated with a tenth of a percentage point increase in the monthly hazard of adoption.

Model 4.3 adds variables for the main effect of *Network concentration* and its interaction term with *Hackathon attendance*. The variable *Local adoption rate* is also included in this model; thus, the coefficients on *Network concentration* should be interpreted in relation to a given local adoption rate. Consistent with Hypothesis 3, the coefficient on the *Network concentration X Hackathon attendance* interaction term is positive ($p \sim 0.000$). This supports our reasoning that platform sponsorship of a hackathon draws attention to the proportion of developers who already use the platform, which can lead non-users to update their expectation on the platform's future network size. The coefficient size indicates that developers who attend a platform-sponsored hackathon with high local network concentration have a 2.5 percentage point higher monthly hazard of subsequently adopting the platform.

Model 4.4 tests whether the economic mechanism (*L Exp. subsidy*) and the social mechanism (*Network concentration*) mutually reinforce one another or substitute for one another. The positive coefficient on the interaction term *L Exp. subsidy X Network concentration* ($p \sim 0.000$) suggests these mechanisms are mutually reinforcing, consistent with Hypothesis 4. The impact of subsidizing platform adoption by providing prizes at a hackathon is stronger when there are already some current platform users at the hackathon, whose presence helps encourage non-users to adopt. For those who experience a positive shift in expectations from attending a hackathon sponsored by a platform, receiving an expected subsidy one standard deviation above the mean level is associated with a 7.2 percentage point increase in the likelihood of adoption.

Alternate dependent variable

Results presented so far relate to the adoption of a platform by a software developer who had not previously used it. To assess another dimension of the complementor–platform relationship, we also test empirical models that measure the effect of hackathons on the intensity of complementor usage of the platform. We run regressions with the dependent variable *L Platform development* using the full panel of data, including developer–platform pairs that have adopted the platform prior to attending any hackathon. The Appendix reports the results of this analysis.

Results in these regression models are generally aligned with the main results, providing additional evidence in support of our hypotheses. We find that: attendance at a hackathon is associated with an increase in platform development (Baseline Hypothesis); a higher local adoption rate at the event is associated with an increase in monthly development of complements

(Hypothesis 2); high network concentration at the event increases platform activity (Hypothesis 3); and the positive association between network concentration and platform development is positively moderated by the level of platform subsidies (Hypothesis 4). We find that the effect of the expected subsidy (Hypothesis 1) is positive but non-significant ($p = 0.22$). Considered alongside our main results, the finding that subsidies appear to have a stronger effect on platform adoption than on platform usage is highly consistent with a conceptual model in which subsidies help overcome the fixed cost of adopting a platform.

Robustness to alternate explanations

The main alternate explanation for our results is that developers find out in advance of a hackathon which companies will be sponsoring it, and they attend the hackathon because they planned to adopt the platform anyway (i.e., a reverse causality argument). We attempt to mitigate this in our main analysis by (1) eliminating from the sample those hackathons with a sponsor name in the event name, (2) eliminating from the sample those hackathons with a single sponsor, and (3) checking contemporaneous hackathon webpages to ensure sponsors are not mentioned (see footnote 8). Nevertheless, to further test this alternate explanation and validate the social interaction mechanism, we run three sets of additional analyses. We summarize the findings here; the Appendix fully documents the details related to data collection and variable construction, and reports full results tables.

First, we reason that if in-person social interactions are an important factor in the association between hackathon attendance and platform adoption, then hackathons with a longer duration ought to have a stronger effect on developers' behaviors. Data on hackathon duration is available for 148 of the 167 hackathons in our dataset. We use this data to partition the hackathons into those with *long* and *short* durations, based on whether they are greater than (>)

or less than or equal to (≤) 24 hours in duration. We run analyses using separate treatment variables for long and short hackathons. Both long and short hackathons have a significant association with platform adoption; Wald tests confirm that the coefficient for long hackathons is larger than for short hackathons. This adds to our confidence that social interaction at the hackathon itself is one of the mechanisms generating the pattern of results we observe.

Second, we consider the possibility that hackathon participants are drawn to the hackathon by the most prominent sponsor and use the hackathon as a chance to adopt their platform. We partition hackathon sponsors into *major* and *minor* sponsors, where the major sponsor is defined as the one offering the largest prize at that event. We run analyses using separate treatment variables for major sponsors and minor sponsors. If developers are self-selecting into hackathons based on the major sponsor, we would expect to find a stronger association between hackathon attendance and adoption of the major sponsors' platforms than adoption of the minor sponsors' platforms. We do not find such a difference: the coefficients for major sponsors' and minor sponsors' platforms are statistically indistinguishable. This gives us some confidence that developers do not learn about sponsors prior to a hackathon and self-select based on those sponsors.

Third, to further address the possibility that developers self-select into hackathons based on the sponsor, we estimate two-stage instrumental variable models. In the first stage, we predict developer attendance at a particular hackathon using their geographic distance from the hackathon as an exogenous instrument.[15] Geographic distance fulfils the characteristics of a desirable instrument because proximity to a hackathon predicts attendance (i.e., the relevance condition), but geographic distance to a hackathon is unlikely to be related to a developer's propensity to adopt a platform *except* through their attendance at that hackathon (i.e., the

[15] Of the developers in our sample, 530 list their city of residence on Devpost. This analysis is restricted to those developers.

exclusion restriction). Because the instrument is not time-varying, for this analysis we pool our longitudinal data and analyze it as a cross-section. The second stage model predicts platform adoptions as measured at the end of our observation window. Our instrumental variable specification produces results consistent with the main analysis: attendance at a platform-sponsored hackathon has a significant and large effect on the likelihood that a developer adopts the sponsoring platform.

DISCUSSION

Our empirical study demonstrates that hackathons, which we conceptualize as social foci, can be an effective tool for platform owners to attract complementors to their ecosystem. We now generalize our theory as it may apply to other types of social focus.

Social foci as coordination devices

Prior research likens the orchestration of a platform ecosystem to a coordination game where, despite the heterogeneous interests of individual actors, all actors prefer to reach a coordinated equilibrium over remaining uncoordinated (Halaburda & Yehezkel, 2016, 2019). We argue in our theoretical exposition that real-world social foci complement digital platforms by establishing a means of coordination. This theoretical reasoning generalizes beyond our specific empirical context of software development hackathons. For instance, Parker, Van Alstyne, and Choudary (2016: 97) relate the story of the launch of Twitter:

> Twitter's breakout moment occurred at the 2007 South by Southwest (SXSW) Interactive film, music, and tech festival… Twitter invested $11,000 to install a pair of giant flat-panel screens in the main hallways at SXSW. A user could text "Join sxsw" to Twitter's SMS shortcode number… and find his or her tweets instantly appearing on the screens. Seeing the feedback on large screens in real time and watching as thousands of new users jumped into the fray created enormous excitement around Twitter…by the end of SXSW, Twitter usage had tripled, from 20,000 tweets per day to 60,000.

This example highlights a subtle but important benefit of using social foci to overcome the chicken-or-egg problem: the value of *simultaneity*. Exposing two potential platform adopters to the platform *at the same time* and *in the presence of each other* is more powerful than exposing one potential adopter at a time because it sets up a triadic relationship between the two adopters (persons P and O) and the platform (entity X).[16] Balance theory predicts that the resulting triad likely consists of either three positively valenced ties or exactly two negatively valenced ties (Cartwright & Harary, 1956; Heider, 1946; Hummon & Doreian, 2003). The pressure towards psychological balance implies that in a social focus, person P is positively predisposed toward X when she sees O positively interacting with X, and *vice versa*. If a platform succeeds in establishing positive ties at a social focus, it might—like Twitter at SXSW—reach a critical mass of users from sponsoring a single real-world social event.[17]

Theoretical contributions

Our study contributes to the strategy literature in two main ways. First, we advance the literature on the antecedents of platform–complementor relationships. Both formal and social mechanisms influence whether a complementor joins a platform ecosystem (McIntyre & Srinivasan, 2017). Prior work tended to examine the formal mechanisms, characterizing complementor–platform ties as arms-length, market-like relationships in which a price system, formal governance rules, and network size are the key factors affecting a complementor's decision over which platform to join. We add to the field's understanding of the social antecedents to platform–complementor relationships (Afuah, 2013), contributing to an emerging stream of work that takes a more socially embedded view of complementor behavior (Boudreau & Jeppesen, 2015; Eckhardt, 2016; Mollick, 2016; Nagaraj & Piezunka, 2018).

Additionally, we contribute to the literature on how social structure influences technology

[16] The notation here—labeling persons as P and O with the platform as X—follows the notation used in Cartwright and Harary (1956).

[17] Interestingly, the theoretical logic of balance theory also highlights how a single *unsuccessful* event will put off a large proportion of potential platform joiners all at once. The recent attempt by Fyre Media to launch a two-sided platform for booking performers by hosting a (failed) music festival is a dramatic illustration of this (Smith, 2019).

diffusion (Rogers, 2003). Scholars have long recognized that networks of interpersonal relations play a role in the spread of new technologies (Coleman *et al.*, 1957; Jackson, 2016; Strang & Soule, 1998). While studies of diffusion on networks often treat the social network structure as stable (Jackson, 2016), interventions like hackathons can change the social network. Thus, we extend existing theory on social foci as antecedents of social structure (Rivera, Soderstrom, & Uzzi, 2010). We show that organizations can strategically seed the diffusion of a technology by sponsoring social foci. While we do not directly measure changes in the social network between individuals, we show that even temporary social foci such as hackathons can act as a locus for technology diffusion.

Limitations and future directions

We draw attention to three limitations of the present study, which each provide openings for future research. First, while our empirical study strives to provide a causal test using a panel research design with staggered treatments, alongside several robustness checks to rule out selection by developers to particular hackathons, we are unable to rule out all forms of omitted variable bias (Angrist & Pischke, 2009). An ideal identification strategy would randomize the exposure of hackathon participants to a certain sponsor. In future work, field experiments could provide a stronger causal test of the mechanisms put forward in this article. For example, scholars or even interested platform owners could run their own controlled hackathon as a field experiment (Ghosh, 2018).

Second, our theoretical logic suggests that social ties form between hackathon participants who were previously strangers. Ideally, we would measure that inter-participant tie formation. Our existing dataset, while detailed, does not permit such measurement. Longitudinal measures of the social ties of hackathon participants would provide a more direct test of the theorized mechanisms.

Third, we test the efficacy of one type of social focus for mobilizing complementors. To validate the theory's generalizability, we call for studies in a broader set of contexts, such as trade shows or entertainment events.

CONCLUSION

With this paper, we sketch the outline of a theory of platform entrepreneurship via the creation of social foci. In their recent review, McIntyre and Srinivasan (2017) identify three prevailing perspectives on platforms grounded in the industrial organization economics, technology management, and strategic management literatures. These perspectives provide deep insights into the pricing, governance, and architecture of platforms, and the competitive interactions between them. We complement these perspectives by drawing on the sociological and organizational literatures on tie formation and social influence. Mechanisms from these literatures help us understand what happens within social foci, and thus why hackathons have a dramatic impact on the platform adoption behaviors of attendees. We aim to open up a broad new line of enquiry that integrates further insights from organization theory into our understanding of platform ecosystem management.

REFERENCES

Abrahamson E, Rosenkopf L. 1997. Social network effects on the extent of innovation diffusion : a computer simulation. *Organization Science* **8**(3): 289–309.

Adner, Ron; Kapoor R. 2010. Value creation in innovation ecosystems: How the structure of technological interdependence affects firm performance in new technology generations. *Strategic Management Journal* **31**(3).

Afuah A. 2013. Are network effects really all about size? The role of structure and conduct. *Strategic Management Journal* **34**: 257–273.

Allison PD. 2014. *Event History and Survival Analysis*, 2nd ed. SAGE: Los Angeles.

Allison PD, Christakis NA. 2006. Fixed-effects methods for the analysis of nonrepeated events. *Sociological Methodology* **36**: 155–172.

Anderson EG, Parker GG, Tan B. 2014. Platform performance investment in the presence of network externalities. *Information Systems Research* **25**(1): 152–172.

Angrist JD, Pischke J-S. 2009. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press: Princeton.

Arena M, Cross R, Sims J, Uhl-bien M. 2017. How to catalyze innovation in your organization. *MIT Sloan Management Review* **58**(4): 39–47.

Baldwin CY, Woodard JC. 2009. The architecture of platforms: A unified view. In *Platforms, Markets and Innovation*, Gawer A (ed). Edward Elgar Publishing: Cheltenham, U.K.: 19–44.

Bikhchandani S, Hirshleifer D, Welch I. 1992. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy* **100**(5): 992–1026.

Bolt W, Tieman AF. 2008. Heavily skewed pricing in two-sided markets. *International Journal of Industrial Organization* **26**(5): 1250–1255.

Boudreau K. 2010. Open platform strategies and innovation: granting access vs. devolving control. *Management Science* **56**(10): 1849–1872.

Boudreau KJ. 2012. Let a thousand flowers bloom? An early look at large numbers of software app developers and patterns of innovation. *Organization Science* **23**(5): 1409–1427.

Boudreau KJ, Jeppesen LB. 2015. Unpaid crowd complementors: the platform network effect mirage. *Strategic Management Journal* **36**(12): 1761–1777.

Boudreau KJ, Lacetera N, Lakhani KR. 2011. Incentives and problem uncertainty in innovation contests: an empirical analysis. *Management Science* **57**(5): 843–863.

Brynjolfsson E, Kemerer CF. 1996. Network externalities in microcomputer software: an econometric analysis of the spreadsheet market. *Management Science* **42**(12): 1627–1647.

Burt RS. 1987. Social contagion and innovation: cohesion versus structural equivalence. *American Journal of Sociology* **92**(6): 1287–1335.

Caillaud B, Jullien B. 2003. Chicken & egg: competition among intermediation service providers. *RAND Journal of Economics* **34**(2): 309–328.

Cartwright D, Harary F. 1956. Structural balance: a generalization of Heider's theory. *Psychological Review* **63**(5).

Ceccagnoli M, Forman C, Huang P, Wu DJ. 2012. Co-creation of value in a platform ecosystem: the case of enterprise software. *MIS Quarterly: Management Information Systems* **36**(1): 263–290.

Cennamo C, Ozalp H, Kretschmer T. 2018. Platform architecture and quality trade-offs of multihoming complements. *Information Systems Research* **29**(2): 461–478.

Cennamo C, Santalo J. 2013. Platform competition: strategic trade-offs in platform markets. *Strategic Management Journal* **34**(11): 1331–1350.

Chu LY, Wu B. 2018. *Designing online platforms for offline services: the bigger, the better?*

Clements MT, Ohashi H. 2005. Indirect network effects and the product cycle: video games in the U.S., 1994—2002. *Journal of Industrial Economics* **53**(4): 515–542.

Coleman J, Katz E, Menzel H. 1957. The diffusion of an innovation among physicians. *Sociometry* **20**(4): 253–270.

Dahlander L, McFarland DA. 2013. Ties that last: tie formation and persistence in research collaborations over time. *Administrative Science Quarterly* **58**(1): 69–110.

Dattée B, Alexy O, Autio E. 2018. Maneuvering in poor visibility: how firms play the ecosystem game when uncertainty is high. *Academy of Management Journal* **61**(2): 466–498.
Dekimpe MG, Parker PM, Sarvary M. 2000. Global diffusion of technological innovations: a coupled-hazard approach. *Journal of Marketing Research* **37**(1): 47–59.
Dou Y, Wu DJ. 2018. Platform competition under network effects: piggybacking and optimal subsidization. *SSRN* : 1–39.
Eckhardt JT. 2016. Welcome contributor or no price competitor? The competitive interaction of free and priced technologies. *Strategic Management Journal* **37**(4): 742–762.
Economides N, Katsamakas E. 2006. Two-sided competition of proprietary vs. open source technology platforms and the implications for the software industry. *Management Science* **52**(7): 1057–1071.
Eisenmann T. 2006. Internet companies' growth strategies: determinants of investment intensity and long-term performance. *Strategic Management Journal* **27**(12): 1183–1204.
Eisenmann T, Parker GC, Van Alstyne MW. 2011. Platform envelopment. *Strategic Management Journal* **32**(12): 1270–1285.
El-Gamal MA, Grether DM. 1995. Are people bayesian? Uncovering behavioral strategies. *Journal of the American Statistical Assocation* **90**(432): 1137–1145.
Ethiraj S, Posen H. 2013. Do product architectures affect innovation productivity in complex product ecosystems? In *Collaboration and Competition in Business Ecosystems*, Adner R, Oxley J, Silverman B (eds). Emerald Publishing Limited: Bingley, U.K.: 127–166.
Evans DS. 2009. How catalysts ignite: the economics of platform-based start-ups. In *Platforms, Markets, and Innovation*, Gawer A (ed). Edward Elgar Publishing: Cheltenham, U.K.: 99–130.
Evans DS, Schmalensee R. 2010. Failure to launch: critical mass in platform businesses. *Review of Network Economics* **9**(4).
Farrell J, Klemperer P. 2007. Coordination and lock-in: competition with switching costs and network effects. In *Handbook of Industrial Organization*, Armstrong M, Porter R (eds). North Holland: Amsterdam, 3: 1967–2072.
Feld SL. 1981. The focused organization of social ties. *American Journal of Sociology* **86**(5): 1015–1035.
Feld SL, Grofman B. 2011. Homophily and the focused organization of ties. In *The Oxford Handbook of Analytical Sociology*, Bearman P, Hedstrom P (eds). Oxford University Press: Oxford: 521–543.
Frolund L, Murray F, Riedel M. 2018. Developing successful strategic partnerships with universities. *MIT Sloan Management Review* **59**(2): 71–79.
Gaba V, Meyer AD. 2008. Crossing the organizational species barrier: how venture capital practices infiltrated the Information Technology Sector. *Academy of Management Journal* **51**(5): 976–998.
Gandal N, Rob R, Kende M. 2000. The dynamics of technological adoption in hardware / software systems : the case of compact disc players. *RAND Journal of Economics* **31**(1): 43–61.
Gawer A, Henderson RM. 2007. Platform owner entry and innovation in complementary markets: evidence from Intel. *Journal of Economics & Management Strategy* **16**(1): 1–34.
Geertz C. 1978. The bazaar economy: information and search in peasant marketing. *American Economic Review* **68**(2): 28–32.
Geertz C, Rosen H, Geertz H, Rosen L. 1979. *Meaning and Order in Moroccan Society: Three Essays in Cultural Analysis*. Cambridge University Press: New York.
Ghosh S. 2018. Iterative coordination in search. *Academy of Management Proceedings* (1).
Gousios G, Spinellis D. 2014. Conducting quantitative software engineering studies with Alitheia Core. *Empirical Software Engineering* **19**(4): 885–925.
Greene WH. 2012. *Econometric Analysis*, 7th ed. Pearson Prentice Hall: Upper Saddle River, NJ.
Greve HR. 2009. Bigger and safer: the diffusion of competitive advantage. *Strategic Management Journal* **30**(1): 1–23.
Greve HR, Seidel M-DL. 2015. The thin red line between success and failure: path dependence in the diffusion of innovation production technologies. *Strategic Management Journal* **36**(4): 475–496.
Gu Y, Zhu F. 2018. Trust and disintermediation: evidence from an online freelance marketplace. *Harvard Business School Working Paper* : 18–103.
Hagiu A. 2006. Pricing and commitment by two-sided platforms. *RAND Journal of Economics* **37**(3): 720–737.

Hagiu A, Spulber D. 2013. First-party content and coordination in two-sided markets. *Management Science* **59**(4): 933–949.
Halaburda H, Yehezkel Y. 2016. The role of coordination bias in platform competition. *Journal of Economics and Management Strategy* **25**(2): 274–312.
Halaburda H, Yehezkel Y. 2019. Focality advantage in platform competition. *Journal of Economics & Management Strategy* **28**(1): 49–59.
Hannah DP, Eisenhardt KM. 2018. How firms navigate cooperation and competition in nascent ecosystems. *Strategic Management Journal* **39**(12): 3163–3192.
Heider F. 1946. Attitudes and cognitive organization. *Journal of Psychology*. Routledge **21**(1): 107–112.
Huang P, Ceccagnoli M, Forman C, Wu DJ. 2013. Appropriability mechanisms and the platform partnership decision: evidence from enterprise software. *Management Science* **59**(1): 102–121.
Hummon NP, Doreian P. 2003. Some dynamics of social balance processes : bringing Heider back into balance theory. *Social Networks* **25**(1): 17–49.
Iansiti M, Levien R. 2004. *The Keystone Advantage: What the New Dynamics of Business Ecosystems mean for Strategy, Innovation, and Sustainability*. Harvard Business School Press: Boston.
Jackson M. 2016. The past and future of network analysis in economics. In *The Oxford Handbook of the Economics of Networks*, Bramoulle Y, Galeotti A, Rogers B (eds). Oxford University Press: Oxford: 71–79.
Jacobides MG, Cennamo C, Gawer A. 2018. Towards a theory of ecosystems. *Strategic Management Journal* **39**(8): 2255–2276.
Kapoor R. 2018. Ecosystems: broadening the locus of value creation. *Journal of Organization Design* **7**(12): 1–16.
Kapoor R, Agarwal S. 2017. Sustaining superior performance in business ecosystems: evidence from application software developers in the iOS and android smartphone ecosystems. *Organization Science* **28**(3): 531–551.
Kaul A. 2013. Entrepreneurial action, unique assets, and appropriation risk: firms as a means of appropriating profit from capability creation. *Organization Science* **24**(6): 1765–1781.
Laumann E. 1973. *Bonds of Pluralism: The Form and Substance of Urban Social Networks*. Wiley: New York.
Lee E, Lee J, Lee J. 2006. Reconsideration of the winner-take-all hypothesis: complex networks and local bias. *Management Science* **52**(12): 1838–1848.
Li Z, Agarwal A. 2017. Platform integration and demand spillovers in complementary markets: evidence from Facebook’s integration of Instagram. *Management Science* **63**(10): 3438–3458.
Lifshitz-Assaf H. 2018. Dismantling knowledge boundaries at NASA: the critical role of professional identity in open innovation. *Administrative Science Quarterly* **63**(4): 746–782.
Lomi A, Lusher D, Pattison PE, Robins G. 2014. The focused organization of advice relations : a study in boundary crossing. *Organization Science* **25**(2): 438–457.
Mansfield E. 1961. Technical change and the rate of imitation. *Econometrica* **29**(4): 741–766.
McIntyre DP, Srinivasan A. 2017. Networks, platforms, and strategy: emerging views and next steps. *Strategic Management Journal* **38**(1): 141–160.
Mollick E. 2016. Filthy lucre? Innovative communities, identity, and commercialization. *Organization Science* **27**(6): 1472–1487.
Moore J. 1996. *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*. HarperBusiness: New York.
Nagaraj A, Piezunka H. 2018. Deterring the new, motivating the established— the divergent effect of platform competition on member contributions in digital mapping communities. *INSEAD Working Paper*.
Ott TE, Bremner RP, Eisenhardt KM. 2018. Beyond the chicken and egg: strategy formation in two-sided marketplace ventures. *Working Paper* : 1–50.
Parker GC, Van Alstyne MW, Choudary SP. 2016. *Platform Revolution: How Networked Markets Are Transforming the Economy and How to Make Them Work for You*. W.W. Norton: New York.
Parker GC, Van Alstyne MW, Jiang X. 2017. Platform ecosystems: how developers invert the firm. *MIS Quarterly: Management Information Systems* **41**(1): 255–266.

Parker GG, Van Alstyne MW. 2005. Two-sided network effects: a theory of information product design. *Management Science* **51**(10): 1494–1504.
Pierce L. 2009. Big losses in ecosystem niches: how core firm decisions drive complementary product shakeouts. *Strategic Management Journal* **30**: 323–347.
Plattner S. 1989. Markets and marketplaces. In *Economic Anthropology*. Stanford University Press: Stanford.
Rietveld J, Eggers JP. 2018. Demand heterogeneity in platform markets: implications for complementors. *Organization Science* **29**(2): 304–322.
Rivera MT, Soderstrom SB, Uzzi B. 2010. Dynamics of dyads in social networks: assortative, relational, and proximity mechanisms. *Annual Review of Sociology* **36**(1): 91–115.
Rochet J, Tirole J. 2003. Platform competition in two-sided markets. *Journal of the European Economic Association* **1**(4): 990–1029.
Rogers EM. 2003. *The Diffusion of Innovations*, 5th ed. Free Press: New York.
Rysman M. 2009. The economics of two-sided markets. *Journal of Economic Perspectives* **23**(3): 125–143.
Schelling TC. 1980. *The Strategy of Conflict*, rev. ed. Harvard University Press: Cambridge, MA.
Seamans R. 2012. Fighting city hall: entry deterrence and new technology deployment in local cable TV markets. *Management Science* **58**(3): 461–475.
Shankar V, Bayus BL. 2003. Network effects and competition: an empirical analysis of the home video game industry. *Strategic Management Journal* **24**(4): 375–384.
Skahill P. 2015, March 27. Police departments open up 'safe lots' for Craigslist transactions. *NPR Morning Edition*.
Smith C. 2019. *Fyre: The Greatest Party That Never Happened*. Netflix.
Sterman J. 2000. *Business Dynamics : Systems Thinking and Modeling for a Complex World*. Irwin/McGraw-Hill: Boston.
Strang D, Soule SA. 1998. Diffusion in organizations and social movements: from hybrid corn to poison pills. *An* **24**: 265–290.
Tee R, Gawer A. 2009. Industry architecture as a determinant of succesful platform strategies: a case study of the i-mode mobile internet service. *European Management Review* **6**: 217–232.
Teece DJ. 2007. Explicating dynamic capabilities: the nature and microfoundations of (sustainable) enterprise performance. *Strategic Management Journal* **28**(13): 1319–1350.
Thomas LDW, Autio E, Gann DM. 2014. Architectural leverage: putting platforms in context. *Academy of Management Perspectives* **28**(2): 198–219.
Tiwana A. 2015. Evolutionary competition in platform ecosystems. *Information Systems Research* **26**(2): 266–281.
Tucker C, Zhang J. 2010. Growing two-sided networks by advertising the user base: a field experiment. *Marketing Science* **29**(5): 805–814.
Venkatraman N, Lee C-H. 2004. Preferential linkage and network evolution: a conceptual model and empirical test in the U.S. video game sector. *Academy of Management Journal* **47**(6): 876–892.
Wade J. 1995. Dynamics of organizational communities and technological bandwagons: an empirical investigation of community evolution in the microprocessor market. *Strategic Management Journal* **16**: 111–133.
West J, Wood D. 2013. Evolving an open ecosystem: the rise and fall of the Symbian platform. In *Collaboration and Competition in Business Ecosystems*, Adner R, Oxley JE, Silverman BS (eds). Emerald Publishing Limited: Bingley, U.K.: 27–67.
Wu L, Wang D, Evans JA. 2019. Large teams develop and small teams disrupt science and technology. *Nature* **566**: 378–382.
Zhu F, Iansiti M. 2012. Entry into platform-based markets. *Strategic Management Journal* **33**(1): 88–106.

TABLES

Table 1: Summary statistics. Observations are at the developer-platform-month level, representing 1,302 developers from January 2012 to October 2017 attending 167 platform-sponsored hackathons. The original data is derived from Devpost.com and Github.com. *Exp. subsidy* is reported in thousands of USD; this variable is log-transformed in regression models.

Dependent Variables	**Obs.**	**Mean**	**Std. Dev.**	**Min.**	**Max.**
Platform adoption	783,474	0.001	0.026	0	1
Platform development	1,183,171	0.179	0.678	0	33
Main Independent Variables					
Hackathon attendance	1,183,171	0.038	0.191	0	1
Network concentration	1,183,171	0.115	0.319	0	1
Exp. subsidy	1,183,171	0.005	0.121	0	15
Local adoption rate	1,183,171	0.081	0.185	0	1
Control Variable					
Project experience	1,183,171	9.559	13.141	0	100

Table 2: Correlation matrix for independent variables.

Variable		**1**	**2**	**3**	**4**	**5**
Hackathon attendance	**1**	1				
Network concentration	**2**	0.111	1			
Exp. subsidy	**3**	0.203	0.036	1		
Local adoption rate	**4**	0.191	0.757	0.048	1	
Project experience	**5**	0.071	0.170	0.024	0.217	1

Table 3: Developer–platform mechanisms from sponsorship. This table depicts the correspondence between theoretical mechanisms affecting developers' decisions whether to adopt a platform and the empirical variables that we use to test the mechanism (for theoretical variables) or else to control for it.

		Corresponding variables in our empirical analysis					
		Theoretical variables				***Control variables***	
Mechanisms affecting adoption		*Hackathon attendance*	*L Exp. subsidy*	*Local adopt. rate*	*Network concentration*	*L Developer experience*	*Platform-month FE*
Economic	Sponsor-provided subsidy	X	X				
Social	Social influence	X		X			
Social	Social influence on expected network size	X		X	X		
Other	Developer-specific cognition (e.g., IQ)					X	
Other	Expected indirect network size						X
Other	Stand-alone value						X

Table 4: Linear regression on *Platform adoption*. In Models (5.C) through (5.4), we run linear probability models with a dependent variable of *Platform adoption*. Across all models, the control variable *L Project experience* and fixed effects for platform-month are included. We do not include individual-level fixed effects in any of the models. Variables preceded by L are logged as ln(1+x). Robust standard errors clustered by developer shown in parentheses. *p*-values shown in brackets.

	Variable	**(4.C)**	**(4.B)**	**(4.1)**	**(4.2)**	**(4.3)**	**(4.4)**
				Platform adoption			
BH	Hackathon attendance		0.019	0.018	0.018	0.013	0.017
			(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
			[0.000]	[0.000]	[0.000]	[0.000]	[0.000]
H1	L Exp. subsidy			0.015			-0.001
				(0.005)			(0.004)
				[0.001]			[0.866]
H2	Local adoption rate				0.011	0.013	0.013
					(0.001)	(0.001)	(0.001)
					[0.000]	[0.000]	[0.000]
	Network concentration					-0.004	-0.002
						(0.000)	(0.000)
						[0.000]	[0.000]
H3	Network concentration					0.025	
	X Hackathon attendance					(0.003)	
						[0.000]	
H4	Network concentration						0.040
	X L Exp. subsidy						(0.011)
							[0.000]
	L Project experience	-3.8e-4	-3.8e-4	-3.9e-4	-4.4e-4	-4.1e-4	-4.4e-4
		(0.000)	(0.000)	(0.000)	(0.000)	(0.000)	(0.000)
		[0.000]	[0.000]	[0.000]	[0.000]	[0.000]	[0.000]
	Platform X Month FE	YES	YES	YES	YES	YES	YES
	Adjusted R^2	0.007	0.022	0.022	0.024	0.029	0.025
	Developers	1302	1302	1302	1302	1302	1302
	Observations	783474	783474	783474	783474	783474	783474

FIGURE

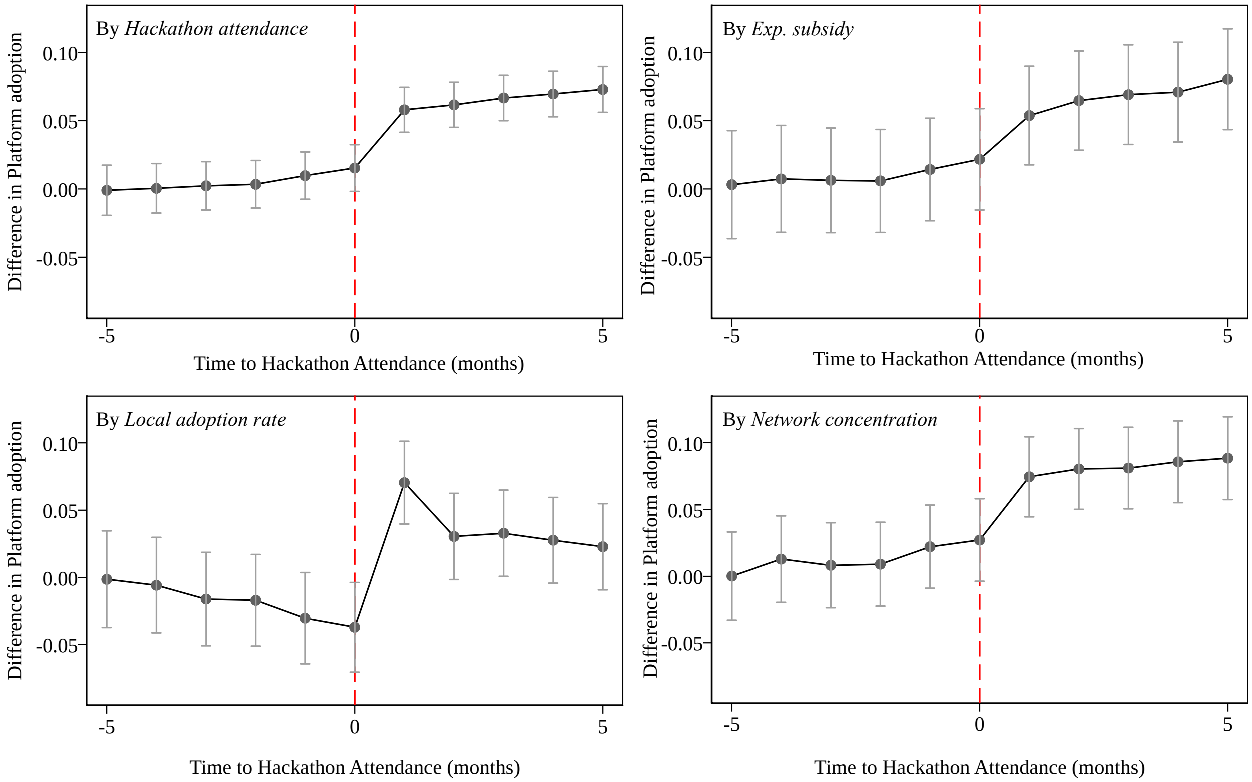

Figure 1. Pre- and post-hackathon trends. This figure depicts the differences in trends for Platform Adoption between treatment and control groups of developer–platform pairs. We compare: pairs that receive the *Hackathon attendance* treatment against those that do not (upper left); pairs that experience an above-mean *Exp. subsidy* against pairs that also attend a platform-backed hackathon but receive an expected average subsidy below or at the mean (upper right); pairs that attend a hackathon with above-mean *Local adoption rate* of the platform against other pairs that also attend a platform-backed hackathon (lower right); and finally pairs that attend a hackathon and experience an increase in expected *Network concentration* against pairs that attend a hackathon but do not experience an increase (lower left). The dashed vertical line represents the last period before treatment. The vertical axis on all panels is the Difference in *Platform adoption*, which indicates whether a developer has already adopted a particular platform for a particular month, between the relevant treatment and control groups. The horizontal axis for Time since Hackathon Attendance should be interpreted as the number of months preceding or following hackathon attendance. The reference group for this comparison is t = -5, the difference between the treatment and control groups five months before hackathon attendance.

APPENDIX

This appendix documents details on the data construction methodology and additional empirical tests to supplement the main manuscript. The order of the items in this Appendix follows the order that they are mentioned in the main paper. First, we detail the process by which we identified platforms in the software developer codebases through appearances of the platform's corresponding application programing interface (API). Second, we describe the hackathon sample construction criteria and provide examples of included hackathons. Third, as a test to address selection by developers towards specific hackathons, we report a balance test on pre-treatment observables, finding no evidence of selection at least on observable characteristics. Fourth, we test our hypotheses with an alternate dependent variable which measures platform usage, as opposed to platform adoption which was reported in the main paper. Fifth, we examine the moderating role of hackathon duration; we find that longer hackathons have a stronger effect on developer behavior, supporting out theory which rests on social interactions as an important factor relating hackathon attendance to platform adoption. Sixth, as another test to address potential selection by developers towards specific hackathons, we examine whether hackathon attendance and platform adoption are moderated by sponsor prominence, and find that they are not. Finally, we verify our main results with an instrumental variables analysis based on the instrument of developer-hackathon geographic distance.

Platform API identification process

We document the process by which we identified platforms through application programing interfaces (APIs) in the codebase of our sample of software developers. Our sample of platforms consists of the platforms that most frequently sponsored hackathons. We first compile a list of 238 platform identifying keywords by searching the sponsors and prizes for each hackathon for mentions of platforms and the associated APIs (e.g. “$1000 – Best use of the Venmo API”). Devpost lists the prizes offered for every hackathon, and we conduct textual analysis on prize data to identify the set of platforms. We then manually remove invalid keywords (such as “target”, “or”, “echo”, etc.). We systematically focus on the 29 platforms that sponsored the most hackathons, as there is a long tail of sponsors for hackathon events. Table A1 presents the list of APIs used in our study and the associated platform owner and category for each API. We limit our set of sponsoring platforms because we could not practically conduct a full search of associated platform adoption through APIs appearing in the codebase of our sample of software developers; API syntax is non-standardized across platforms. By focusing on this subset of platform providers, we are able to maintain accuracy and precision in our measurement of APIs appearing in the codebase.

---------- INSERT TABLE A1 ----------

After selecting the list of platforms and identifying API syntax associated with those platforms, we run a strict matching algorithm on Amazon Web Services to search through the Github code for each developer. Our direct output from this matching strategy is a dictionary that lists each hackathon project and the corresponding set of APIs that were used in each project. We optimize our matching strategy by adding cases for specific languages and entry points. For example, invoking an API in a java application (import ibmwatson) is different from invoking an API through a web service (curl -X POST https://gateway-a.watsonplatform.net). We considered these cases to reduce concerns about a “greedy” matching strategy that would inappropriately capture API usage. For example, this adjustment is especially important when considering social media platforms, such as Facebook or Twitter, where a developer can include a Twitter button on her application, but not leverage any true functionality through the Twitter API, such as receiving data for a stream of Twitter posts.

To validate our matching strategy, we leverage the fact that projects submitted to Devpost also include tags or keywords that detail what technology was used for a project. These are self-described tags, but because these tags are manually assigned and used by hackathon organizers to find projects that qualify for a prize offered by the hackathon organizer or the platform sponsor(s), we imagine them to have a reasonable degree of accuracy. We run our matching strategy to produce a list of API usage for each project, and then compare the list terms with the self-described tags. We are able to get an accuracy of 67.8% in our matching strategy, suggesting that our matching strategy is not producing many false negatives.

In addition, we extract the line of code for which we find the platform/API mentioned in the codebase of the software developer's projects. By looking at the context of the match, we can examine how the keyword is being used. Table A2 provides an example of two different contexts of how a keyword might appear in the software developer's code. Based on this analysis, we are able to confirm that our matching strategy does not produce many false positives. This analysis supports the validity of the novel measures of platform adoption and development introduced in this paper.

---------- INSERT TABLE A2 ----------

Hackathon sample construction

We provide illustrative detail on the sample of hackathons included in the study in Table A3. Our starting sample includes all hackathons submitted to Devpost.com that take place between January 2014 and May 2017. We then select hackathons for inclusion in our sample based on a number of criteria that match our theoretical framework and empirical strategy. First, we include hackathons that are based in a physical venue and have at least ten identifiable participants. In the next stage, we select hackathons that offer prizes from two or more sponsors. In the final step, we exclude hackathons that prominently featured a single platform sponsor in the event title. The remaining set of 167 hackathons serve as the quasi-experimental treatment events in our study.

---------- INSERT TABLE A3 ----------

Table A4 shows an example of selected hackathon data available through Devpost.com which we used in the filtering process. We filter the set of hackathons based on the name and physical venue location of each hackathon event. We exclude a hackathon if the hackathon name includes any mention of sponsoring platforms or if a physical location is not listed. In the next column, we measure the number of projects by scraping data on each submission made to the hackathon on Devpost. We exclude hackathons that receive more than 10 submissions. Using the date column, we restrict the set of hackathons to those between January 2014 and May 2017. Finally, each hackathon lists the set of prizes offered, which we use to determine the sponsoring platforms of the hackathon.

---------- INSERT TABLE A4 ----------

Balance test on pre-treatment observables

We check for balance on pre-treatment observables between the treatment and control groups across the various platforms considered and find no statistically significant differences between the two groups. To address the concern that developers are self-selecting into hackathons based on sponsor, we present a portion of the analysis in Table A5 for three platforms: Amazon Web Services, Google, and Microsoft. These platforms are popular and frequently sponsor hackathon events, and thus they are most susceptible to developer selection. For each platform, we find that the level of platform adoption and development is comparable between the treatment and control group in the pre-period. We find no statistically significant differences between the two groups across all the platforms considered in our study. In addition, we perform an analysis that examines pre-treatment observables across all platforms and developers. Developer–platform pairs in the two groups appear to be comparable because they exhibit similar levels of project experience and develop on a similar number of total platforms prior to the hackathon. We find no statistically significant differences between the two groups across all the platforms considered in our study.

------ INSERT TABLE A5 -------

Alternate dependent variable: *Platform development*

We verify the robustness of our results relative to an alternative dependent variable that accounts for a developer's intensity of use of a platform. Our main results relate to the adoption of a platform by a software developer who had not used it prior to hackathon attendance. To assess another dimension of the complementor–platform relationship, we also test empirical models that measure the effect of hackathons on the intensity of complementor usage of the platform. This dependent variable captures not just new adoptions but also the ongoing activity of existing platform users.

The alternate dependent variable, *Platform development*, measures the count of a developer's active GitHub projects that use a specific platform. To construct this variable, we count a project if a platform's API was used in the raw code of a developer's project and the developer actively made changes to the project during that month. Given the skewed distribution of the base measure (see Table 1 in the main manuscript), we apply a log-transformation to *Platform development* before using it in the regression analysis.

We run regressions with the dependent variable *L Platform development* using the full panel of data, including developer–platform pairs that have adopted the platform prior to attending any hackathon. In addition to the control variable *L Developer experience* and platform-month fixed effects, we include developer fixed effects in these models to control for time-invariant unobservable heterogeneity across developers. Table A6 reports the results of this analysis. We find that: attendance at a hackathon is associated with an increase in platform development (Baseline Hypothesis); a higher local adoption rate at the event is associated with an increase in monthly development of complements (Hypothesis 2); high network concentration at the event increases platform development activity (Hypothesis 3); and the positive association between network concentration and platform development is positively moderated by the level of platform subsidies (Hypothesis 4). The effect of the expected subsidy (Hypothesis 1) is positive but not as statistically significant as in the main analysis ($p = 0.22$). Considered alongside our main results, the finding that subsidies appear to have a stronger effect on platform adoption than on platform usage is highly consistent with a conceptual model in which subsidies help overcome the fixed cost of adopting a platform, but not necessarily in motivating on-going developer usage over time.

---------- INSERT TABLE A6 ----------

Social mechanism moderator: Hackathon duration

We examine the moderating role of hackathon duration and find that hackathons of longer duration have a stronger effect. We use event duration as a proxy for the level of in-person social interactions that developers might experience from attending a hackathon – a hackathon that occurs over a longer period of time will offer more opportunities for interaction, on average. If social mechanisms are an important factor in the association between hackathon attendance and platform adoption, as our theory suggests, then we would expect that hackathon duration serves as a positive moderator in this relationship.

To conduct this analysis, we collect additional data on the time duration of hackathons, as measured in hours. The hackathon duration come from self-reported information on Devpost. Devpost provides optional fields that allow hackathon organizers to input details such as the length of an event, the event schedule, or dates for the hackathon. Given that this data is not available for all hackathons, this analysis uses fewer observations than our main analysis: we document the duration of 148 hackathons out of the 167 used in our main analysis.

By distinguishing hackathons based on the duration of each event, we create two measures that proxy for the amount of exposure that developers have with one another at the event. We implement two alternative versions of our main independent variable on hackathon attendance to account for the duration of the event. We define *Hackathon attendance: short* as a dichotomous variable that takes the value of one for the month in which a developer attended a hackathon sponsored by the platform and all the months that follow, if the hackathon duration is 24 hours or less, and zero otherwise. We define *Hackathon attendance: long* as a dichotomous variable that takes the value of one for the month in which a developer attended a hackathon sponsored by the platform and all the months that follow if the hackathon occurs over a period of more than 24 hours, and zero otherwise.

We present the results of our robustness tests using the duration moderators in Table A7a. In model A7a.1, we run the same specification as model 4.3, replacing *Hackathon attendance* with our alternative independent variables of *Hackathon attendance: short* and *Hackathon attendance: long*. Consistent with Hypothesis 3, the coefficients on the *Network concentration X*

Hackathon attendance: short and *Network concentration X Hackathon attendance: long* interaction terms are positive.

---------- INSERT TABLE A7a ----------

The Wald tests in Table A7b demonstrate that the effect of the interaction between *Network concentration* and attendance is larger for those who attend a longer hackathon as opposed to a shorter one. In model A7b.2, we find similar effects using our alternate dependent variable *L Platform development*. This supports our reasoning that those who attend a hackathon for a longer period of time are more likely to receive an increase in their expected network concentration, due to more opportunities for social learning. These analyses add to our confidence that social interaction at the hackathon itself is one of the mechanisms generating the pattern of results we observe.

---------- INSERT TABLE A7b ----------

Developer selection towards major sponsor

We consider the possibility that hackathon participants are drawn to the hackathon by the most prominent sponsor and select into the hackathon with the intention of adopting the major sponsor's platform. In our main analysis, we filter the hackathons in our sample to only include those that offer prizes from more than one sponsoring platform and do not include the sponsor in the name of the event to mitigate this concern. However, we cannot observe other ways in which the hackathon organizers and platform sponsors may promote the hackathon outside of Devpost. The sponsors that are investing more in financial and in-kind support for a hackathon may be more likely to share information ex-ante. If there is a selection effect where developers are attending a hackathon based on ex-ante information about the sponsors and prize amounts, a disproportionate amount of information should be shared about the major sponsor. If developers are self-selecting into hackathons based on the major sponsor, we would expect to find a stronger association between hackathon attendance and adoption of the major sponsors' platforms than adoption of the minor sponsors' platforms. We argue that if there is significant selection in hackathon attendance, our hypotheses should be less significant for less prominent, or "minor", sponsoring platforms at each hackathon event.

In order to test this, we create a measure *Major sponsor* that indicates whether a platform was the major sponsor at a hackathon. We define the major sponsor for a hackathon as the sponsoring platform that is offering the highest expected subsidy to developers. We operationalize this by creating a time-invariant dummy variable for each developer-platform combination. This dummy is set to one if the platform was the largest platform sponsor, in terms of expected subsidy, at the hackathon attended by the developer.

In Table A8, model A8.1 includes our major sponsor moderator in a specification similar to model 4.1 in Table 4 of the main manuscript. This moderator allows us to measure heterogeneity in the treatment effect between platforms that are major sponsors (the most prominent sponsor) at the hackathon attended by a developer and the other sponsoring platforms. Our base term, *Hackathon attendance*, measures the baseline association between attending a platform-sponsored hackathon and the developer's hazard of adopting the sponsor platform, for all platforms. The coefficient for this term is, as expected, positive ($p \sim 0.000$) and consistent with the estimates from our main analysis. This provides evidence that our effect is present for all sponsoring platforms at the hackathon, rather than just the most prominent sponsor. These findings suggest that our results still hold even if there were a selection effect by developers to hackathons based on prior knowledge of a sponsor.

---------- INSERT TABLE A8 ----------

Instrumental variable analysis

To further address the possibility that developers might self-select into hackathons when they already intend to adopt the sponsor's platform, we use a two-stage least squares instrumental variables model. In the first stage, we use a developer's geographic proximity to a hackathon as an exogenous instrument that predicts their likelihood of attending a given hackathon. In the second stage, we model a developer's subsequent platform adoption behavior using the fitted values for the endogenous independent variable from the first-stage model.

Relevance condition and exclusion restriction

We argue that the geographic proximity of a developer to a hackathon serves as a valid instrumental variable because it meets both the relevance condition and the exclusion restriction. The relevance condition holds if the physical distance between the developer and a hackathon affects the developer's attendance at a hackathon. In addition, we argue that proximity also meets the exclusion restriction, which means that the instrument only affects the dependent variable through its effect on the independent variable, with no direct effect on the dependent variable. The relative distance between the hackathon and the developer is unlikely to be correlated with unobserved platform preferences of the developer. Instead, a developer's choice to adopt a platform might be related to their previous experience, their set of skills, or other unobservables. The developer-hackathon distance should not be directly related to these factors, and so we expect that being close to a hackathon will affect platform adoption only by increasing a developer's likelihood of attendance.

Sample construction

We use an alternative cross-sectional data structure for this analysis, because the instrumental variables analysis exploits time-invariant variation in the distance between the developer and the potential set of hackathons they might attend. In contrast, the main analysis exploits temporal variation in when a developer attends a hackathon.

We now describe the structure of this developer-platform-hackathon dataset, first intuitively and then formally. Intuitively, we create a placebo hackathon-platform risk set for each developer. For each developer, we consider them at risk to attend any hackathon earlier or in the same period as the hackathon they did actually attend in real life. Accordingly, they are at risk to be treated by sponsors of that set of hackathons. We only include observations relating a developer to a hackathon-platform if that platform did sponsor a hackathon in the developer's hackathon risk set. Formally, we consider a hackathon k and platform j pair in the risk set for a developer i, if the platform was a sponsor at the hackathon, and the hackathon event occurred prior to or in the same period as the actual hackathon that the developer attended. This construction results in a developer-platform-hackathon dataset of 88,125 observations.

Variable construction

The instrumental variable *Developer-hackathon distance* measures the distance between the developer and a hackathon. We define this using data collected from Github about developer city of residence, and data from Devpost on the hackathon venue address. Github lists location data for 530 of the 1302 developers in our main analysis. After geocoding the developer and hackathon locations at the city-level, we measure the geodetic distance between each developer-hackathon pair.[1] The *Developer-hackathon distance* variable is measured in kilometers, and we log transform the variable due to skew. Because a developer's propensity to attend a hackathon is inversely related to distance, we code this variable as the negative value of the *Developer-*

[1] To calculate this distance, we use the geodist package in Stata to create a distance matrix for each developer-hackathon combination. The geodist function calculates the length of the shortest curve between two points along the surface of a mathematical model of the earth. Reference: Picard, R. (2010). GEODIST: Stata module to compute geodetic distances. Statistical Software Components, Boston College Department of Economics.

hackathon distance; this allows us to interpret the first-stage estimates as the positive association between geographical proximity to the hackathon and a developer's attendance at the event.

Our dependent variable, *Post-event platform adoption*, is a dichotomous variable that indicates whether or not developer *i* ever adopted platform *j* in the post-period. This contrasts with our main dependent variable, *Platform adoption*, which measures whether the developer had adopted in a particular month period. To keep our analysis consistent with the main model, we only consider developers who have not adopted prior to attending a hackathon.

Results

We now present the results of the first and second stage of our instrumental variables analysis. In both stages, we control for *L Developer experience* and include hackathon fixed effects and platform fixed effects to control for time-invariant heterogeneity in hackathons and platforms. Because we have one instrument, we can only conduct analyses that include only a single endogenous independent variable, so we analyze the effect of *Hackathon attendance* and *Local adoption rate* in two separate models.

Table A9b shows the first-stage results for the endogenous independent variables of *Hackathon attendance* and *Local adoption rate*. We test the relevance condition of the instrument in the first stage; in Table A9a, the *F* statistic for the instrument is 1026.3 for *Hackathon attendance* and 534.8 for *Local adoption rate*. The coefficients for our estimates also suggest that a developer's attendance at a hackathon and the *Local adoption rate* at a hackathon are increasing as the *Developer-hackathon distance* shrinks.

---------- INSERT TABLE A9a ----------

Table A9b provides the IV estimates on *Post-event platform adoption* from the second stage of the 2SLS model. This second stage uses the fitted values from the first stage and regresses them against *Post-event platform adoption*. In model A9b.1, we examine the association between attending a platform-sponsored hackathon and the developer's likelihood of adopting the sponsor platform at any point after the hackathon. Upon attending a hackathon sponsored by a given platform, the likelihood the developer adopts the platform increases by 47.8 percentage points. In model A9b.2, we examine *Local adoption rate* and find that every ten percentage point increment in the *Local adoption rate* of a given platform at a hackathon is associated with a 17.7 percentage point increase in the likelihood of adoption.

---------- INSERT TABLE A9b ----------

These results are consistent with our main analysis. Our point estimates are different because this instrumental variable analysis uses a different dependent variable that measures likelihood to adopt at any point after the hackathon, rather than a monthly hazard rate.

APPENDIX TABLES

Table A1: API list. For our 29 platforms, we search for the relevant API keyword on Programmable Web, and identify the listed service functionalities of each API. We place APIs that match in at least one function into the same category and identify the primary service provided by each group. Finally, we list the platform owner of each API, which was determined by the API documentation. Our list is sorted in alphabetical order by platform owner and API keyword.

Platform Owner	API	Category
Amazon	alexa	Internet of Things
Amazon	amazon	eCommerce
Amazon	aws	Cloud
Clarify	clarify	Audio
Dropbox	dropbox	Storage
Ebay	ebay	eCommerce
Facebook	facebook	Social
Facebook	fbsearch	Search
Facebook	instagram	Photo
Foursquare	foursquare	Social
Google	firebase	Productivity
Google	google	Search
Google	nest	Internet of Things
IBM	ibm	Cloud
IBM	watson	Internet of Things
Mastercard	mastercard	Payment
Microsoft	azure	Cloud
Microsoft	microsoft	Productivity
Microsoft	outlook	Productivity
MongoDB	mongodb	Productivity
Paypal	paypal	Payment
Spotify	spotify	Audio
Staples	staples	eCommerce
Twilio	twilio	Productivity
Twitter	twitter	Social
Uber	uber	Transportation
Visa	visa	Payment
Yahoo	yahoo	Search
Yelp	yelp	Social

Table A2: False positive verification example.

Sample Code Context	Matching
"import maps.google.com"	Valid
"we searched google for the best validation technique"	False Positive

Table A3: Sample size at each stage. As detailed in the hackathon sample subsection of our paper, we filter our set of hackathons in order to minimize the risk of endogeneity from self-selection by developers into hackathon participation based upon which platform was sponsoring the event. At each stage of our sample selection process, we remove hackathons that do not match our criteria. Our final set contains 167 hackathons with 1,302 attending developers. Developer Count indicates the number of developers for which we could identify their Github account.

Sample Stage	Hackathons	Developers
All Hackathons	1,587	12,439
Physical Hackathons with More Than Ten Submissions	438	2,948
Hackathons with at Least One Platform sponsor	198	1,430
Final Set of Hackathons	167	1,302

Table A4: Sample hackathons. We present two sample hackathons included in our final sample with selected data available through Devpost. We selected hackathons if the hackathon name does not include any mention of sponsoring platforms. Selected hackathons were based at physical venues and received more than 10 submissions. Each hackathon offered a set of prizes, which we use to determine the sponsoring platforms of the hackathon. Finally, we restrict the set of hackathons to those between January 2014 and May 2017. Projects refers to the number of submitted Github projects associated with the hackathon.

Name & Location	Projects	Date	Prizes
Hack the North 2016 *University of Waterloo Waterloo, ON N2L 3G5, Canada*	104	Sep 16, 2016 – Sep 18, 2016	*Top 12 Winners* Hack the North will award each member on the team with a choice of: XBOX One, Playstation 4, iPad *Microsoft Azure* Microsoft will award a Surface Pro 4 and Surface Arc Mouse to each member of the team that best utilises Microsoft Azure. *Yelp API* Yelp will award a Leap Motion to each member of the team that best utilises the Yelp API.
HackHarvard 2015 *Harvard University Agassiz Theatre Cambridge, MA, USA*	25	Nov 14, 2015 – Nov 15, 2015	*Best use of Microsoft API* Best use of Microsoft Azure API receives prize. *Best use of Facebook API* Samsung Gear VR Innovator Edition for every team member. *Best use of AWS (MLH)* 1TB Hard Drive for every team member, sponsored by MLH.

Table A5: Pre-treatment balance test. In this table, we subset our balance tests on only developer–platform pairs for three selected platforms: Amazon Web Services, Google, and Microsoft. Observations occur one period before developers attended a hackathon (the final pre-treatment period). *Platform development* and *Platform adoption* are specific to the Developer–Platform relationship.

	Not Sponsor		**Sponsor**			
Variable	**Obs.**	**Mean**	**Obs.**	**Mean**	**Difference**	***p*-value**
AWS						
Platform adoption	746	0.331	556	0.369	-0.037	0.411
Platform development	746	0.390	556	0.426	-0.036	0.227
Google						
Platform adoption	1072	0.532	230	0.522	0.011	0.766
Platform development	1072	0.629	230	0.800	-0.171	0.099
Microsoft						
Platform adoption	932	0.406	370	0.403	-0.003	0.915
Platform development	932	0.346	370	0.359	-0.014	0.816

Table A6: Linear regression on *L Platform development*. All models are OLS regressions with a dependent variable of *L Platform Development*. The control variable *L Project experience* and fixed effects for platform-month and developer are included across all models. Variables preceded by L are logged as ln(1+x). Model (A6.C) serves as a baseline model with only control variables, and we use different independent variables through Models (A6.B) to (A6.4) to test our hypotheses. Robust standard errors clustered by developer shown in parentheses. p-values shown in brackets.

		(A6.C)	**(A6.B)**	**(A6.1)**	**(A6.2)**	**(A6.3)**	**(A6.4)**
	Variable	L Platform development					
BH	Hackathon attendance		0.021	0.019	0.011	0.001	0.010
			(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
			[0.000]	[0.001]	[0.020]	[0.770]	[0.047]
H1	L Exp. subsidy			0.036			-0.020
				(0.029)			(0.024)
				[0.221]			[0.402]
H2	Local adoption rate				0.180	0.179	0.179
					(0.020)	(0.024)	(0.024)
					[0.000]	[0.000]	[0.000]
	Network concentration					-0.003	-0.000
						(0.007)	(0.007)
						[0.539]	[0.968]
H3	Network concentration X Hackathon attendance					0.039	
						(0.011)	
						[0.001]	
H4	Network concentration X L Exp. subsidy						0.088
							(0.050)
							[0.080]
	L Project experience	0.024	0.024	0.024	0.017	0.017	0.017
		(0.003)	(0.003)	(0.003)	(0.003)	(0.003)	(0.003)
		[0.000]	[0.000]	[0.000]	[0.000]	[0.000]	[0.000]
	Platform X Month FE	YES	YES	YES	YES	YES	YES
	Developer FE	YES	YES	YES	YES	YES	YES
	Adjusted R^2	0.323	0.323	0.323	0.330	0.330	0.330
	Developers	1302	1302	1302	1302	1302	1302
	Observations	1183171	1183171	1183171	1183171	1183171	1183171

Table A7a: Hackathon duration moderators.

	Variable	(A7a.1) Platform adoption	(A7a.2) L Platform development
BH	Attendance: short	0.012	-0.010
		(0.001)	(0.002)
		[0.000]	[0.000]
BH	Attendance: long	0.014	0.015
		(0.001)	(0.003)
		[0.000]	[0.000]
H2	Local adoption rate	0.011	0.047
		(0.001)	(0.006)
		[0.000]	[0.000]
	Network concentration	-0.003	-0.010
		(0.000)	(0.002)
		[0.000]	[0.000]
H3	Network concentration	0.015	0.022
	X Attendance: short	(0.004)	(0.005)
		[0.000]	[0.000]
H3	Network concentration	0.026	0.047
	X Attendance: long	(0.004)	(0.006)
		[0.000]	[0.000]
	L Project experience	-4.2e-4	-3.6e-4
		(0.000)	(0.000)
		[0.000]	[0.000]
	Platform X Month FE	YES	YES
	Adjusted R^2	0.027	0.186
	Developers	1071	1071
	Observations	651430	1070419

Table A7b: Wald test comparing short and long hackathon duration estimates. This table provides Wald tests on the difference in estimated coefficients from Table A7a. In (A7b.1), we test the difference between groups 1 and 2 using the estimates from (A7a.1). In (A7b.2), we test the difference between groups 1 and 2 using the estimates from (A7a.2). Robust standard errors clustered by developer shown in parentheses. p-values shown in brackets.

		(A7b.1)	**(A7b.2)**
		Difference	
Group 1	Group 2	Platform adoption	L Platform development
Attendance: short	Attendance: long	0.002	0.025
		(0.002)	(0.003)
		[.221]	[.000]
Network concentration X Attendance: short	Network concentration X Attendance: long	0.011	0.025
		(0.006)	(0.007)
		[.070]	[.001]

Table A8: Major sponsor moderator.

	Variable	**(A8.1)** Platform adoption
BH	Hackathon attendance	0.018
		(0.001)
		[0.000]
	Major sponsor	-0.001
		(0.000)
		[0.000]
	Hackathon attendance	0.001
	X Major sponsor	(0.001)
		[0.376]
	L Project experience	-3.6e-4
		(0.000)
		[0.000]
	Platform X Month FE	YES
	Adjusted R^2	0.007
	Developers	1302
	Observations	783474

Table A9a: First stage coefficients.

	(A9a.1) Hackathon attendance	**(A9a.2)** Local adoption rate
L Developer-hackathon distance	.023	.006
	(0.001)	(0.000)
	[0.000]	[0.000]
L Developer experience	.001	.001
	(0.001)	(0.000)
	[0.008]	[0.000]
Platform FE	YES	YES
Hackathon FE	YES	YES
F Statistic	1026.28	534.75
Observations	88125	88125

Table A9b: Reduced form coefficients on *Platform adoption*.

	(A9b.1)	**(A9b.2)**
	Platform adoption	
Hackathon attendance	0.478	
	(0.047)	
	[0.000]	
Local adoption rate		1.771
		(0.179)
		[0.000]
L Developer experience	0.160	0.159
	(0.002)	(0.002)
	[0.000]	[0.000]
Platform FE	YES	YES
Hackathon FE	YES	YES
Adjusted R^2	0.056	0.050
Observations	88125	88125